Reliable Recognition of Handwritten Digits Using A Cascade Ensemble Classifier System and Hybrid Features

Ping Zhang

A Thesis
in
The Department of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montreal, Quebec, Canada

April 2006

Chapter One

Introduction

1.1 OCR: the Motivation

Optical Character Recognition (OCR) is a branch of pattern recognition, and also a branch of computer vision. OCR has been extensively researched for more than four decades. With the advent of digital computers, many researchers and engineers have been engaged in this interesting topic. It is not only a newly developing topic due to many potential applications, such as bank check processing, postal mail sorting, automatic reading of tax forms and various handwritten and printed materials, but it is also a benchmark for testing and verifying new pattern recognition theories and algorithms.

In recent years, many new classifiers and feature extraction algorithms have been proposed and tested on various OCR databases and these techniques have been used in wide applications.

Numerous scientific papers and inventions in OCR have been reported in the literature. It can be said that OCR is one of the most important and active research fields in pattern recognition. Today, OCR research is addressing a diversified number of sophisticated problems. Important research in OCR includes degraded (heavy noise) omnifont text recognition, and analysis/recognition of complex documents (including texts, images, charts, tables and video documents). Handwritten numeral recognition, (as there are varieties of handwriting styles depending on an applicant's age, gender, education, ethnic

background, etc., as well as the writer's mood while writing), is a relatively difficult research field in OCR.

1.2 Focus of This Thesis

The focus of this thesis is the recognition and verification of unconstrained handwritten numerals, which is a challenging research project as these numerals are written without any constraints, (e.g., they are not all written in separate boxes, nor all written neatly, nor all using a specific type of pen). In addition, as mentioned before, unconstrained handwritten numerals have varieties of writing styles due to different backgrounds of the writers.

Technically speaking, OCR systems pursue a high recognition rate while seeking the highest reliability, which makes it practical for recognizing unconstrained handwritten numerals. Here are some criteria for measuring the recognition performance.

The recognition rate (***RR***) is defined as:

$$RR = \frac{Number\ \ of\ \ correctly\ \ recognized\ \ characters}{Total\ \ number\ \ of\ \ testing\ \ characters} \qquad \ldots\ldots (1.1)$$

The reliability (***RE***) can be denoted as:

$$RE = \frac{Total\ \ number\ \ of\ \ testing\ \ characters - Number\ \ of\ \ misrecognized\ \ characters}{Total\ \ number\ \ of\ \ testing\ \ characters} \qquad \ldots\ldots (1.2)$$

Our work started with research on hybrid feature extraction. It is important to make the feature extraction algorithms insensitive to the character's size, rotation, shifting, and to the variation of writing styles.

One research aspect was to design effective feature extraction methods. A novel Medial Axial Transformation (MAT) based feature extraction method was developed, which has produced an excellent recognition performance. We proposed a two dimensional real wavelet transformation and a two dimensional complex wavelet transformation for the hybrid feature extractions. In total, seven sets of features were proposed.

Our next focus was to develop a verification model for similar character pairs. Theoretical research on multi-modal nonparametric discriminant analysis was proposed in order to reduce feature dimensionality, thereby enhancing the classifiers' computation efficiency.

In the last part of our research work, we proposed a cascade ensemble classifier recognition system for the recognition of handwritten numerals with a rejection strategy in order to obtain the highest recognition rate with a minimal error rate, or the best reliability performance, based on the following reasons:

It is common sense that the misrecognition rate can be a sensitive issue in some applications such as bank check reading and mailing letter sorting. Recognition with a proper rejection option provides a means to reduce the error rate through a rejection mechanism, i.e., the rejection option may withhold a decision if the confidence value is not high enough and it may direct a rejected pattern to an exceptional handling for manual inspection. With a rejection option, the system reliability is increased.

As we know, it is difficult for a single classifier to get a very high reliability rate for handwritten digital recognition due to the variability of handwriting styles. There are a few possible solutions to help reduce the number of errors. One solution is to employ a verification module. Another solution is to use a combination of multiple classifiers. The different features extracted by different means, which are inputted to different classifiers for classification, have different merits for recognition because some of the features are complementary. It is reasonable to combine several classifiers to produce the highest reliability and at the same time to seek the lowest misrecognition rate.

1.3 Research Goals

Research goals of this thesis are twofold: theory and application. The theoretical aspect is focused on the following: research on hybrid feature extraction, multi-modal nonparametric analysis for feature dimensionality reduction for the verification of handwritten numerals, and the cascade ensemble classifier recognition system with rejection strategy in order to pursue the highest reliability. The applications are based on the proposed theories, to implement the OCR system.

1.3.1 Theory

This thesis will mainly address the following issues:

- Proposing hybrid feature extraction methods.
- Researching a multi-modal nonparametric analysis method for the curse of dimensionality of the features.

- Analyzing the tradeoff of the error, rejection and recognition rates of the cascade ensemble classifier system.
- Designing a novel ensemble classifier scheme, which consists of Artificial Neural Networks (ANNs) and Gating Networks (GNs).

1.3.2 Applications

We can apply the theoretical research to the following aspects:

- To implement hybrid feature extraction methods (including 2-D real and complex wavelet features; Medial Axial Transformation-based gradient features, etc.)
- To implement a series of pair-wise verifiers based on multi-modal nonparametric analysis for feature dimension reduction.
- To implement a cascade ensemble classifier system for the recognition of handwritten numerals with rejection strategies in order to pursue the highest recognition rate with a minimal error rate.

1.4 Outline of the Thesis

This thesis is organized as follows:

In Chapter One, Motivation of handwritten numeral recognition is described. The focus of our work, the research goal and implementation are discussed.

In Chapter Two, a comprehensive survey of feature extraction and selection, classification, recognition, and verification methods for handwritten numerals is given.

In Chapter Three, preliminary knowledge about the theory and the structure of the Artificial Neural Network (ANN) with back-propagation learning algorithms, used as a classifier in this thesis, is discussed. The basic theory and concept of wavelets are presented in order to give a theoretical background for wavelet-based feature extraction.

In Chapter Four, seven sets of hybrid features are extracted by using different approaches. Based on the multi-class divergence analysis, a multi-class feature ranking and feature random selection scheme is proposed in order to produce three sets of new randomly selected features. A general recognizer is implemented and some handwritten numeral recognition results based on the extracted features are also given.

In Chapter Five, research on multi-modal nonparametric analysis for feature compression and selection is conducted. The multi-modal nonparametric analysis for feature dimensionality reduction leads to the design of a verification model for better distinguishing the similar numeral pairs of the recognition system.

In Chapter Six, theoretical analysis of the tradeoff of the error, rejection, and correct recognition rates in an ANN classifier, an ensemble classifier including an ensemble logical "and" scheme and an ensemble average scheme, and a cascade ensemble classifier system are investigated. The theoretical research proves that it is possible to design and implement a cascade ensemble recognition system for the recognition of handwritten digits with a very high recognition rate and a minimal error rate.

In Chapter Seven, a series of cascade ensemble classifier schemes are proposed. One of the novel ensemble classifier schemes includes three ANNs and three gating networks with a rejection strategy in order to reduce the misrecognition rate while pursuing the

highest recognition rate. Comprehensive experiments are conducted using different strategies and the recognition results on the MNIST are given.

In Chapter Eight, conclusions are drawn about the contributions of this thesis. The analysis of the proposed methods and suggestions for further research are presented.

Chapter Two

Literature Review

This chapter includes a handwritten digit recognition literature review. Topics to be reviewed are: handwritten recognition systems, handwritten verification methods, handwritten digit feature extraction, feature dimensionality reduction and selection, and recognition with rejection strategies for improving the recognition system's reliability.

2.1 Handwritten Digit Recognition System

OCR has been researched for many years and OCR systems have been continuously improving as will be explained in this section:

In Brown et al. [7], a recognition system for the unconstrained handprinted numerals was proposed, which used topological, geometrical and local measurements to identify the character or to reject the character as unrecognizable. The recognition system yielded a recognition rate of 97% with a substitution error rate of 0.3% and a rejection rate of 2.7%.

In Stringa [106], a pattern recognition system was applied to the unconstrained alphanumeric character recognition. The recognition system was designed to allow hierarchical re-description of the input images and the phrase-structure grammars were

developed. The experiments conducted on handwritten digits indicated that the recognition rates were comparable to the best OCR system at that time, but with a considerable reduction in computing time.

In Suen et al. [108], four experts for the recognition of handwritten digits were proposed. In expert one, the skeleton of a character pattern was decomposed into branches. The pattern was then classified according to the features extracted from these branches. In expert two, a fast algorithm based on decision trees was used to process the more easily recognizable samples, and a relaxation process was applied to those samples that could not be uniquely classified in the first phase. In expert three, statistical data on the frequency of occurrence of features during training were stored in a database. This database was used to deduce the identification of an unknown sample. In expert four, structural features were extracted from the contours of the digits. A tree classifier was used for classification. The resulting multiple-expert system proved that the consensus of these methods tended to compensate for individual weakness, while preserving individual strengths. The high recognition rates were reported and compared favorably with the best performance in the field.

Mitchell and Gillies [77] used the tools of mathematical morphology to extract cavity features as the starting input for their specialized digit recognizers. A classification system was implemented by a symbolic model matching process.

Le Cun et al. [19] achieved excellent results with the convolutional neural networks, which were specifically designed to deal with the variability of two dimensional (2-D) shapes. For the recognition of handwritten numerals, the recognition rate with this method could be as high as 99.18% on the MNIST database.

Recently, many improvements have been reported, especially in pursuing a higher recognition rate. In Simard et al [105], authors expanded the training set of the MNIST dataset by adding a new form of distorted data, and the convolutional neural networks were better suited for classification purposes. The recognition rate was achieved at 99.60%.

Shi et al. [102] proposed a handwritten digit recognition system using the gradient and curvature of the gray character image in order to improve the accuracy of handwritten numeral recognition. The experiments were conducted on IPTP CDROM1, NIST SD3, and SD7 databases. The recognition rates could reach from 98.25% to 99.49%.

Teow and Loe [110] proposed a handwritten digit recognition system based on a biological vision model. The features were empirically extracted by the model, which could linearly separate over a large training set (MNIST). The high recognition rate was reported, where the error rate was 0.59%.

Decoste and Scholkopf [21] proposed a handwritten digit recognition system where the prior knowledge about invariance of a classification problem was incorporated into the training procedure. Support Vector Machines (SVMs) were used as classifiers. The system achieved a low error rate of 0.56% when using this procedure with the MNIST dataset.

Recently, many handwritten digit recognition systems with very high recognition rates have been emerged. These recognition systems were conducted on the well known MNIST database. Here are some examples:

- 99.58% of SVCs on gradient features (Liu et al., 2002) [69],
- 99.41% of LIRA_grayscale (Kussul and Baidyk, 2004) [61],

- 99.46% of Trainable Feature Extractor and Support Vector Machine (TFE-SVM) with affine transformations for increasing the training set (Lauer et al., 2005) [65],
- 99.56% of Image Recognition Systems with Permutative Coding (Kussul et al., 2005) [62].
- 99.63% of Support Vector Machine VSVM^{b} (Dong et al. 2005) [22,23]

A comprehensive survey on handwritten numeral recognition by using different feature extraction methods, and different classifiers on CENPARMI, CEDAR, MNIST databases has been reported in [70]. The classifiers included one k-nearest classifier, three neural classifiers, a learning vector quantization classifier, a discriminative learning quadratic discriminative function classifier and two support vector classifiers. On the MNIST test dataset, 80 recognition results were given by combining eight classifiers with ten feature vectors. The error recognition rates were between 1.50 and 0.61.

2.2 Verification Methods for Handwritten Digits

A verification model was proposed in the 1990's in order to increase the OCR system's reliability. For the verification of handwritten numerals, Zhou et al. [125] investigated some verification schemes. Verification-enhanced systems were proposed with extensive experiments conducted on both isolated and touching numerals. There were two layers of verification modules: class-specific verifiers and pair-wise verifiers. A class-specific verifier was designed to distinguish one class from other classes, e. g., (Is it a "1"?). A pair-wise verifier was used to verify the recognized characters into two categories, e. g., (Is it a "4" or "9"?).

Oliveira et al. [85] discussed two types of level verifications on handwritten numeral strings: high-level and low-level. The high-level verifier dealt with a subset of the classes in order to confirm or deny the hypotheses produced by the general-purpose recognizer. The low-level verifier dealt with meta-classes of the system (characters and parts of characters). The purpose of the low-level verifier was to determine whether a hypothesis generated by the general-purpose recognizer was valid or not.

Teredesai et al. [111] proposed Genetic programming (GP) to evolve secondary classifiers for disambiguating between pairs of handwritten digit images. A two-step classification strategy was presented in the paper. The first step of the classification used a full feature set. If the confidence was high, the recognition result would lead to an end of the recognition process. Otherwise, a secondary classifier was designed in the second step by a subset of the original feature set and the information available from the earlier classification step would help classify the input further. The combination of first- and second-stage classifiers was able to achieve a 99.0% acceptance rate and a 0.3% error rate.

2.3 Feature Extraction

The purpose of feature extraction is to get the most relevant and the least amount of data representation of the character images in order to minimize the within-class pattern variability while enhancing the between-class pattern variability. There are two categories of features: statistic features and structural features.

In the statistic feature domain, Hu [46] introduced the use of moment invariants as features for pattern recognition. Hu's absolute orthogonal moment invariants (invariant to translation, scale and rotation) have been extensively used in the recognition systems.

In Krzyzak et al. [55], features were firstly extracted from the contours of numerals: 15 complex Fourier descriptors were extracted from the outer contours and simple topological features were extracted from the inner contours. These features were directly presented as the input of a three-layer ANN for recognition.

In recent years, wavelet transform has been an emerging tool for feature extraction. In Chen, Bui and Krzyzak's paper [13], a multiwavelet orthonormal shell expansion was used on the contour of the character to get several resolution levels and their averages. Finally, the shell coefficients were used as the features input into a feed-forward neural network to recognize handwritten numerals.

Tao et al. [109] investigated the utility of several emerging techniques to extract features. The central projection transformation was applied to describe the shape of the characters; then the wavelet transformation was used to aid in the boundary identification, and the fractal features were employed to enhance image discrimination for the recognition of printed Chinese characters and English letters of varying fonts.

In Lee's paper [67], Kirsch masks were adopted for extracting four directional local feature sets and one global feature set. A three-layer cluster neural network with five independent subnetworks was developed for classifying similar numerals.

In the structural feature extraction domain, in Suen et al. paper [108], the comprehensive structural features were systemically implemented, such as the combined branch features, giving information on the following: shape, length, angular change, degree of curvature,

vertical and horizontal general directions, nature of the starting and ending points (J points and E points), their coordinates, the distance and the primitive features such as line segments, (open) convex polygons, and loops, etc.

Liu et al. [70] summarized state-of-the-art feature extraction techniques, which included the extraction of chaincode features, gradient features, profile structure features and peripheral direction contributivity. The recognition performance comparisons among different types of features were given in the paper.

In Gader et al. [33], a linear correlation feature extractor for handwritten digit recognition was described. Two different evaluation measures: orthogonality and information, were used to guide the search for features. ANNs with Back Propagation (BP) algorithms were used as classifiers in the recognition experiments of handwritten digits. The classification rates compared favorably with results published in the literature.

Weideman et al. [114] extracted 36 normalized moment features, 18 topological features, 24 2-D FFT features, and 16 shadow features that were found by projecting the character onto the nearest bars in the horizontal, vertical, and diagonal directions. The length of the shadow on each bar was used as a feature. The comparisons of a neural network and a nearest-neighbor classifier for the recognition of numeric handprint characters were reported.

Oh et al. [82] proposed two feature sets based on distance transformation. In the first feature set, the distance from each white pixel to the nearest black pixel in the character image without the thinning operation was considered as a distance transformation feature. The second feature was called Directional Distance Distribution (DDD), which contained rich information encoding both black/white and directional distance distributions. A new

method of map tiling was also introduced and applied to the DDD feature to improve its discriminative power. The experiments were conducted on three sets of characters (numerals, English letters, and Hangul initial sounds). The results confirmed the superiority of both the DDD feature and the map tiling.

In Yang et al. [117], high-order B-splines were used to calculate the curvature of the contours of handwritten numerals. The concept of a distribution center was introduced so that a one-dimensional periodic signal could be normalized as a shift invariant. The curvature of the contour of a character became rotation invariant. ANNs and SVMs classifiers were employed to train the features. High verification rates on similar numeral pairs were reported.

Oliveira et al. [87] proposed a specific concavity, contour-based feature sets for the recognition and verification of handwritten numeral strings. The OCR system could process either isolated digits or handwritten numeral strings.

Gao and Ding [34] proposed two new feature extraction strategies: the modified multiple discriminant analysis and the difference principal component analysis. The proposed algorithms were useful in automatic feature extraction from different patterns. Experiments have shown that the two new methods provided more effective feature metrics for pattern discrimination in the recognition of Chinese character fonts and handwritten digits.

Trier et al. [113] presented a comprehensive overview of feature extraction methods for off-line recognition of isolated characters. The feature extraction methods included: (1) template matching; (2) deformable templates; (3) unitary image transforms; (4) graph descriptions; (5) projection histograms; (6) contour profiles; (7) zoning; (8) geometric

moment invariants; (9) Zernike moments; (10) spline curve approximations; and (11) Fourier descriptors. The mentioned methods could be applied to one or more of the following character forms: (1) gray-level character images; (2) binary character images; (3) character contours; and (4) character skeletons or character graphs.

2.4 Feature Dimensionality Reduction and Selection

There are two types of feature dimensionality reduction. One is called feature selection, which uses some criteria to select fewer features from the original feature set. The second type uses an optimal or sub-optimal transformation to conduct feature dimensionality reduction. The latter is an information congregation operation rather than the operation of deleting less useful features.

Feature selection is an important step in OCR. In a large feature set (where, normally, the number of features is greater than 100), the correlation of features is complicated. Retaining informative features and eliminating redundant ones are a recurring research topic in pattern recognition. Generally speaking, feature extraction and feature dimensionality reduction serve two purposes: (1) to improve the training and testing efficiency, and (2) to improve the reliability of a recognition system.

Divergence distance measurement is one feature selection criterion. Intuitively, if the features show significant differences from one class to another, the classifier can be designed more efficiently with a better performance [83]. Some well-known feature selection methods include: Sequential Forward Selection/Sequential Backward Selection (SFS/SBS), Sequential Forward Floating Selection/Sequential Backward Floating

Selection (SFFS/SBFS) [91], and the Branch and Bound algorithm (BAB) [80], as well as its varieties of the improved algorithms [38, 103]. These algorithms have shown good feature selection performance in small- and medium-scale feature sets. The time complexities for both SFFS/SBFS and BAB are $O(2^n)$, where n is the dimension of the feature set. The BAB algorithm requires the criterion function to be monotonic [56].

Genetic algorithms (GAs) offer a particularly attractive approach to feature selection since they can generally perform quite an effective search of a large, non-linear space [104]. In the handwritten character recognition area, some researchers [51, 86] have developed OCR-oriented criteria or fitness functions, which can alleviate the computation complexity for a given feature number m, ($m \leq n$, n is the number of features initially extracted). However, the main drawback of the GA method lies in the difficulty of exploring different possibilities of trade-off between having classification accuracy and having different subsets of selected features [86].

Some feature selection is based on the unsupervised scheme using feature similarity. For example, Mitra et al. [76] proposed a method to measure similarity between features whereby redundancy therein is removed.

Oh et al. [84] proposed a novel hybrid algorithm for feature selection. Local search operations were devised and embedded in hybrid GAs to fine-tune the search. The hybridization technique produced two desirable effects: a significant improvement in the final performance and an acquisition of subset-size control.

In Bressan et al.'s paper [4], a feature selection method was proposed based on the Independent Component Analysis (ICA) under the assumption of class-conditional independence.

Nunes et al. [81] presented an optimized Hill-Climbing algorithm to select a subset of features for handwritten character recognition. The search was conducted taking into account a random mutation strategy and the initial relevance of each feature in the recognition process. The experiments showed a reduction in the original number of features used in an Multiple Layer Perceptron (MLP)-based character recognizer from 132 to 77 features (a reduction of 42%) without a significant loss in recognition rates.

Another research direction of feature dimensionality reduction is based on information theory. The basic idea is to retain as much information as possible while conducting feature transformation. When feature extraction or dimensionality reduction is carried out, a criterion function should be given for minimizing the information loss and for increasing classification separability. If the mapping is linear, it means that the mapping function is well defined and our task is simply to find the coefficients of the linear function so as to maximize or minimize a given criterion. Unfortunately, in many applications of pattern recognition, those important features, which are not simply linear functions of the original measurements, are highly nonlinear functions. Therefore, the goal is to find an appropriate nonlinear mapping function for a given dataset as follows:

Karhunen-Loeve (K-L) transformation or Principal Component Analysis (PCA), taken as a whole, is an optimal signal representation method in the sense that it provides the smallest mean square error for the compression of a given set of data. Fukunaga and Koontz [30] successfully applied K-L expansion to feature selection and ordering. Fukunaga-Koontz's method works well in problems where the covariance differences are dominant and where there is little or no mean difference between two pattern classes. In

order to find the best vectors for discriminating between two classes, it is necessary to select an optimal criterion.

Foley and Sammon [26] used the Fisher ratio shown in equation (2.1) to get discriminant vectors and the corresponding discriminant values iteratively. The discriminant vectors can be ordered according to their corresponding discriminant values.

$$R(d) = \frac{(d^t \Delta)^2}{d^t A d} \qquad \ldots\ldots(2.1)$$

where

d: n-dimensional column vector on which the data are projected;

Δ: the difference vector in the estimated means of two classes;

$\boldsymbol{W_i}$: within-class scatter for class *i*;

A=c$\boldsymbol{W_1}$+(1-c)$\boldsymbol{W_2}$ and $0 \le c \le i$.

In practical applications, Foley and Sammon's method fails to find the correct feature vector if there is little or no difference in the means between two classes.

Feature extraction using nonparametric discriminant analysis is a useful and efficient way of using discriminant analysis in statistics. Within-class, between-class, and mixture scatter matrices are used to formulate the criteria of class separability. We review the definitions of the above-mentioned scatter matrices below.

A within-class scatter matrix (S_w) shows the scatter of data around their respective class expected vectors, and is expressed by:

$$S_w = \sum_{i=1}^{L} P_i E\{(X - M_i)(X - M_i)^T \mid \omega_i\} = \sum_{i=1}^{L} P_i \sum_i \qquad \ldots\ldots (2.2)$$

A between–class scatter matrix (S_b) is the scatter of the expected vectors around the mixture mean as follows:

$$S_b = \sum_{i=1}^{L} P_i (M_i - M_0)(M_i - M_0)^T \quad \ldots\ldots (2.3)$$

where M_0 represents the expected vector of the mixture distribution and is given by

$$M_0 = E\{X\} = \sum_{i=1}^{L} P_i M_i \quad \ldots\ldots (2.4)$$

The mixture scatter matrix (S_m) is the covariance matrix of all the data regardless of their class assignments, and is defined by the following:

$$S_m = E\{(X - M_0)(X - M_0)^T\} = S_w + S_b \quad \ldots\ldots (2.5)$$

If we set the optimal discriminating power on a classifier, we can use a criterion J, which has the following form:

$$J = trS_2^{-1} S_1 \quad \ldots\ldots (2.6)$$

where $tr(.)$ is the trace operation; S_1 represents the between-class matrix S_b, while S_2 represents the within-class scatter matrix S_w.

Fukuanaga and Mantock [31] gave a mono-modal based nonparametric discriminant analysis method for feature extraction of a two-class classification problem. The authors processed all training samples to deduce the optimal criterion ***J***. In addition, authors applied the de-emphasing method [32] to those data that were far from the classification boundary, in order to preserve the classification structure. In order to calculate a nonparametic S_b of the two-class classification problem, for each training sample, Fukuanaga and Mantock's algorithm needed to calculate the mean of *K-Nearest Neighbor* (*K-NN*) data in the other class, then form a between-class scatter matrix. The computation complexity for S_b is $O(N^2 \log N)$. In that method, the decision boundary information is not taken into consideration in designing the feature extraction algorithm.

Lee and Langdgrebe [66] introduced the concepts of discriminantly redundant features and discriminantly informative features for classification. If a feature vector was parallel to the decision boundary, then the feature vector was considered to be discriminantly redundant. These features did not make any contribution to recognition performance. The effective decision boundary feature matrix was constructed by finding some training sample pairs in two classes near the decision boundary, by forming the unit normal vector to the decision boundary, and by calculating an estimate of the effective decision boundary feature matrix. However, the training sample pairs might have been selected in such a way that they could reflect all the distributions of the two classes along the decision boundary, especially for multi-modal distribution.

Hastie et al. [39] proposed that in many situations, a single prototype was not sufficient to represent inhomogeneous classes, and that mixture models were more appropriate. The authors used the mixture discriminant analysis (MDA) method to model each class by using a mixture of two or more Gaussians with different centroids. A weighted optimal scoring scheme was presented in order to produce a blurred response matrix, and then both of the flexible discriminant analysis and the penalized discriminant analysis adapted naturally to the MDA.

In reference [40], Hastie and Tibshirani fitted Gaussian mixtures to each class to facilitate effective classification in non-normal settings, especially when the classes were clustered subclasses. A subclass shrinkage method was introduced in order to deduce the between-subclass variability relative to the between-class variability. Furthermore, the authors [41] proposed a locally adaptive form of nearest neighbor classification to suppress feature dimensionality. An iterative scheme was applied to estimate an effective

metric for computing neighborhoods, and then to shrink the neighborhoods in directions orthogonal to these local decision boundaries, and to elongate them parallel to the boundaries.

Torkkola [112] used the nonparametric estimation of mutual information between class labels and transformed features. A quadratic divergence measure was employed to make an effective non-parametric implementation.

In the paper by Fukumizu et al. [29], the authors treated the problem of dimensionality reduction as that of finding a low-dimensional effective subspace, and they derived a contrast function to estimate the effective subspace for dimensionality reduction.

In reference [5], Bressan and Vitria explored the connection between Nonparametric Discriminant Analysis (NDA) and the Nearest Neighbors (NN) classifier and proposed a modified Mono-modal Nonparametric Discriminant Analysis method (MNDA) for feature dimensionality reduction. The between-class scatter matrix S_b was calculated in the same way as Fukunaga's method. However, for computing the within-class scatter matrix S_w, the authors adapted a modified method as follows: Firstly, find the mean ($Mean(X_i)$) of the *k-NN* for each training sample X_i ($i=1,2,...,D$; where D is the number of training samples); then, calculate the difference between the training sample and its *k-NN* mean as follows: $\Delta_i = X_i - Mean(X_i)$; finally, congregate the covariance matrix of all the training samples: $S_w = \frac{1}{D}\sum_{i=1}^{D} \Delta_i \Delta_i^T$ to form a new within-class scatter matrix. The computation complexity of S_b was the same as that of NDA. This method also needed to calculate an additional *k-NN* from the same class for each training sample

in order to compute S_w. So the overall computational complexity of this method was greater than that of the NDA method.

Feature selection or feature dimensionality reduction based on nonparametric discriminant analysis is a promising method. In Chapter 5, we will further develop a new algorithm for feature dimensionality reduction, where the nonparametric discriminant analysis and decision boundary information are combined.

2.5 Classifier

Designing a classifier is not a research topic in this thesis; however, classification is a vital step in the OCR system. A brief survey on classifiers is investigated below.

Nearest Neighbor classifier (NN) was well described in a book written by Duda et al. [25]. It uses a predefined distance to measure the similarity between a feature vector of the testing sample and a feature vector set for the class. The distance function can be Euclidean or Hamming distance. The problem with this method is that it has a high computation cost and is inflexible.

The polynomial discriminant classifier [100] assigns a pattern to a class with the maximum discriminant value, which is computed by a polynomial in computing a feature vector. The class models are implicitly represented by the coefficients in the polynomial.

The Bayesian classifier assigns a pattern to a class with the maximum a posterior probability. The class prototypes are used in a training stage to estimate the class-conditional probability density function for a feature vector [25, 79].

Tree classifiers are used to reduce the complexity in prototype matching. There are many well-known tree classifiers, such as CART [3] and C4.5 [92]. Ho [45] used the C.5 decision tree and reported good results on recognition problems.

Hidden Markov Model (HMM) [93] consists of a set of states and the transition probabilities between consecutive states. When using HMM for a classification problem, an individual HMM is constructed for each pattern class. For each observation sequence, i.e., for each sequence of feature vectors, the likelihood for the sequence is calculated. The class in which the HMM achieved the highest probability, is considered to be the class that produced the actual sequence of observations. For example, Gunter and Bunke [37] used HMM for handwritten text recognition. Britto et al. [6] proposed the recognition of handwritten numeral strings using a two-stage HMM-based method, and Park et al. [88] used a 2-D HMM for character recognition.

The utilization of the Support Vector Machine (SVM) classifier has gained immense popularity in the past years [8, 54, 89]. SVM is a discriminative classifier based on Vapnik's structural risk minimization principle. It can be implemented on flexible decision boundaries in high dimensional feature spaces. Generally, an SVM solves a binary (two-class) classification problem, and multi-class classification is accomplished by combining multiple binary SVMs. Good results on handwritten numeral recognition by using SVMs can be found in Dong, et al.'s paper [23].

Artificial Neural Networks (ANN), due to its useful properties such as: highly parallel mechanism, excellent fault tolerance, adaptation, and self-learning, has become increasingly developed and successfully used in character recognition [2, 9, 14, 15, 35, 118]. The key power provided by such networks is that they admit fairly simple

algorithms where the form of nonlinearity that can be learned from the training data. The models are thus extremely powerful, have nice theoretical properties, and apply well to a vast array of real-world applications.

One of the most popular methods for training such multilayer networks is based on gradient descent method, namely, the backpropagation algorithm or generalized delta rule. The method is powerful, useful, and relatively easy to understand and implement. As ANN is mainly used as classifier in my thesis, a three-layer ANN with BP algorithms will be discussed in detail later in Chapter 3.

2.6 Combination of Classifiers

The classifier Combination, or the ensemble classifier has attracted theoretical and practical attention. There are two purposes, one is to increase recognition reliability; another is to increase recognition accuracy [53, 94].

Xu et al. [116] proposed four combining classifier approaches according to the levels of information available from the various classifiers. The experimental results showed that the performance of individual classifiers could be improved significantly.

Huang and Suen [47, 48] proposed the Behavior-Knowledge Space method in order to combine multiple classifiers for providing abstract level information for the recognition of handwritten numerals.

Ho [44] proposed the theory of a decision combination scheme based on the ranking of classes by each classifier. The ranks were implemented by different types of classifiers.

Woods et al. [115] presented a method using local accuracy estimates. The combination method used the estimates of each individual classifier's local accuracy in small regions of a feature space. Synthesized data sets were used for the experiments.

Cho and Kim [16] conducted experiments with network fusion using a fuzzy integral. The fuzzy integral was a nonlinear function that was defined with respect to a fuzzy measure.

Lam and Suen [64] studied the performance of combination methods that were variations of the majority vote. A Bayesian formulation and a weighted majority vote (with weights obtained through a genetic algorithm) were implemented, and the combined performances of seven classifiers on a large set of handwritten numerals were analyzed.

In recent years, some new theories and solutions on combinations of classifiers have been proposed in the boarder areas, but were not limited to the OCR field.

Kuncheva and Jain [57] proposed two ways to use a genetic algorithm (GA) to design a multiple-classifier system. The first GA version selected disjoint feature subsets to be used by individual classifiers, whereas the second version selected overlapping feature subsets and the types of the individual classifiers. GA design could be made less prone to overtraining by including penalty terms in the fitness function, which accounts for the number of features used.

In reference [58], Kuncheva investigated six fusion methods of classifiers by estimating the posterior probability for each class. It was assumed that the estimates were independent and identically distributed (normal or uniform) and that the formulas were given for the classification error for each of the following fusion methods: average, minimum, maximum, median, majority vote and oracle.

In references [59, 60], Kuncheva et al. proposed a combination of classifier selection and fusion methods by using statistical inference to switch between the two. Selection was applied in those regions of the feature space where one classifier strongly dominated the others from the pool and fusion was applied in the remaining regions. The Decision Templates (DT) method was adopted for the classifier part. The papers included a discussion on when to combine classifiers and on how classifier selection can be misled.

Alkoot and Kittle [1] also gave an experimental evaluation of expert fusion strategies and validated the classifiers experimentally. The experimental results on different experts were given.

Liu et al. [71] investigated a number of confidence transformation methods for the measurement-level combination of classifiers. Each confidence transformation method was a combination of a scaling function and an activation function. The confidence transformation methods were used in handwritten digit recognition. The results showed that confidence transformation was efficient enough to improve the combination performance.

The research on multi-layer ensemble classifiers is a new and challenging topic, which will be discussed in Chapters 6 and 7.

2.7 Classification with Rejection Strategy

There are a few hierarchical or multi-classifier strategy recognition systems published in the literature. For example, some researchers pursue the higher recognition accuracy, or

less computation time; others introduce a hierarchical rejection policy, as summarized below:

Giusti et al. [36] proposed a two-stage system based on first (global) classifier with rejection followed by a (local) nearest-neighbor classifier. In that system, the patterns rejected by the first classifier were classified by the second classifier (the nearest-neighbor classifier), which looked for the top-h classes. The experiments were conducted on NIST-3 handwritten digits. Recognition rates ranged from 93.80% to 94.09% with different networks; however, the tradeoff of the error rate and the rejection rate (reliability) was not given explicitly. Rodriguez et al. [98] dealt with handwritten digit recognition using a three-level classifier with rejection techniques. *K-NN* and *K-NCN* were used as classifiers. The rejection strategy at each level was different. The recognition rates were reported from 97.34% to 99.54% conducted on NIST database. As an example with rejection option, the reliability of 99.99%, recognition rate of 86.68%, and rejection rate of 13.23% was given. Mayraz and Hinton [75] proposed the product of experts learning procedure for the recognition of handwritten digits. The experiments were conducted on the MNIST database and the recognition rate was over 98%. No rejection rate (reliability) was given. Cecotti and Belaid [12] proposed a rejection strategy for the convolutional neural network models. A self-organizing map was used to change the links between the neural network layers. Instead of learning all the possible deformation of the patterns, ambiguous patterns were rejected and the network's topology was modified. The experiments were conducted on the MNIST database. As an example, the recognition rate of 92.12% and the corresponding rejection rate of 7.63% were reported. Nunes et al. [81] introduced a cascade classifier based on feature subsets of

different sizes. The proposed method demonstrated a significant reduction in terms of computational complexity. Rahman et al. [95] proposed a multiple expert framework (3 classifiers, 2 parallel and 1 serial) and Clifford algebra to form ANN's weights. The experiments were conducted on NIST handwritten digits and the recognition rate of 90.34% was reported. Schettini et al. [99] presented a hierarchical classification scheme for classifying images into photographs, graphics, texts and compound documents. The classifier used was the *CART* tree. Heisele et al. [43] used a serial classifier structure to recognize foreground images and to reject background images for face detection. SVMs were used as classifiers. In reference [27], Frelicot and Mascarilla dealt with a combination of pattern classifiers with two rejection options. A decision rule was proposed for classifying or rejecting patterns either for distance or ambiguity. The experimental results were conducted on the Waveforms, Satimages and Iris of the UCI Repository of Machine Learning Database. In reference [126], Zimmermann et al. investigated three different rejection strategies for offline handwritten sentence recognition. The rejection strategies were implemented as a postprocessing step of a text recognition system based on Hidden Markov Model.

Chapter Three

Preliminaries

In this chapter, we will review two preliminaries: Artificial Neural Network (ANN) and Wavelet Transform (WT). ANN has been widely used in pattern recognition as a classifier due to its useful properties such as: highly parallel mechanism, excellent fault tolerance, adaptation, and self-learning. One of the most successful applications is its application to OCR. WT has been a hot topic for the past twenty years due to its merits, such as localization, multiresolution analysis and a fast algorithm, which are very useful for fine (detail) feature extraction in the pattern recognition field. A detail analysis will be presented in the following section.

3.1 Three-Layer ANN Classifier

An ANN is an interconnected group of artificial neurons. ANN refers to electrical, mechanical or computational simulations or models of biological neural networks. One of the most popular methods for training a multilayer network is based on the gradient descent principle using the back-propagation algorithm or generalized delta rule. The principle is a natural extension of the Least Mean Squares (LMS) algorithm because it is powerful, useful, and relatively easy to understand and implement.

An ANN classifier consists of input units, hidden units, and output units. In terms of classifying ten numerals, we will have ten output units, one for each of the ten numerals,

and the signal from each output unit is the discriminant function $g_k(x)$. The discriminant function can be expressed as:

$$g_k(x) \equiv z_k = f\left(\sum_{j=1}^{r} w_{kj} f\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j0}\right) + w_{k0}\right) \quad \ldots\ldots (3.1)$$

where x_i is a feature component; w_{ji} is a weight between the input layer and the hidden layer; w_{kj} is a weight between the hidden layer and the output layer; *i=1,...,d,* and *d* is the number of nodes in the input node; *j=1,..., r,* and *r* is the number of nodes in the hidden layer*;* *k*=0,1,2,…, 9, which represents the number of nodes in the outputs layer. For example, 10 nodes of outputs represent ten digits.

Thus, the discriminant function can be implemented by a three-layer neural network. The configuration of the three-layer neural network for the recognition of ten handwritten numerals is drawn in Fig. 3.1. A more intuitive proof of the universal expressive power of three-layer nets is inspired by Fourier's theorem. The theorem states that any continuous function $g_k(x)$ can be approximated arbitrarily by a possible infinite sum of the harmonic function, given a sufficient number of hidden units n_H, proper nonlinearities, and weights [25].

We now turn to the crucial problem of setting the weights based on training patterns and the desired output.

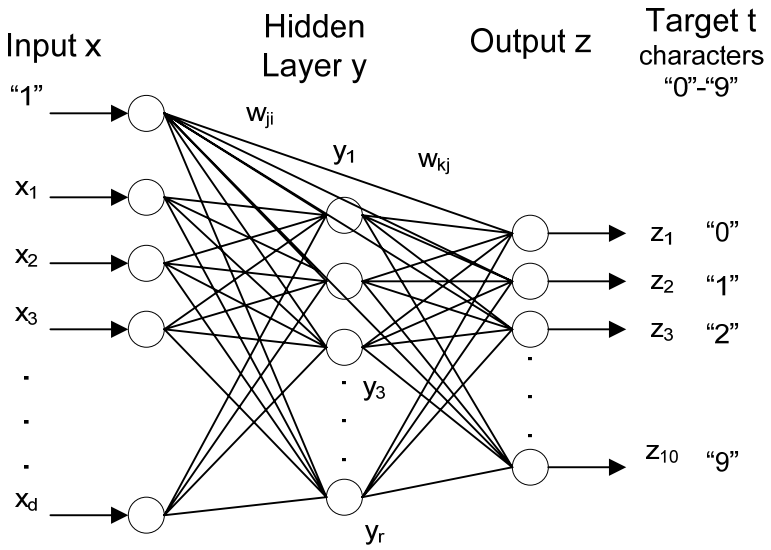


Fig. 3.1 Configuration of three-layer neural networks

3.2 Backpropagation Algorithm

Backpropagation is one of the simplest and most general methods for the supervised training of multilayer neural networks. The training error on a pattern is considered to be the sum of the output units from the squared differences between the desired output t_k given by a teacher and the ANN' output z_k:

$$J(w) \equiv \tfrac{1}{2}\sum_{k=1}^{c}(tp_k - z_k)^2 = \tfrac{1}{2}\|tp - z\|^2 \qquad \ldots\ldots (3.2)$$

where *tp* and *z* are the target and the network output vectors of length c, and w represents the weights in the network.

The backpropagation learning rule is based on gradient descent. The weights are initialized with random values, and then they are changed in a direction that leads to a reduction in the squared error in equation (3.2):

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad \ldots\ldots (3.3)$$

where η is a learning rate. The iterative algorithm updates the weights as follows:

$$w(m+1) = w(m) + \Delta w(m) \quad \ldots\ldots (3.4)$$

where *m* indexes the particular pattern presentation.

For a three-layer neural network, consider first the hidden to output weights: w_{kj}, if we do differentiation:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}} \quad \ldots\ldots (3.5)$$

Apply equation (3.5) to equation (3.2), then δ_k can be simply represented as:

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k} = (tp - z_k) f'(net_k) \quad \ldots\ldots (3.6)$$

Each output similarly computes its net activation based on the hidden unit signal y_j as

$$net_k = \sum_{j=1}^{r} y_j w_{kj} + w_{k0} = \sum_{j=0}^{r} y_j w_{kj} \quad \ldots\ldots (3.7)$$

where y_0=1, and the following derivative exists:

$$\frac{\partial net_k}{\partial w_{kj}} = y_j \quad \ldots\ldots (3.8)$$

So the weight update or learning rule for the hidden-to-output weights is:

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (tp_k - z_k) f'(net_k) y_j \quad \ldots\ldots (3.9)$$

In analogy with equation (3.6), the sensitivity of the hidden unit is defined as [25]:

$$\delta_j \equiv f'(net_j) \sum_{k=1}^{c} w_{kj} \delta_k \quad \ldots\ldots (3.10)$$

The learning rule for the input-to-hidden weight is:

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \sum_{k=1}^{c} w_{kj} \delta_k f'(net_j) x_i \quad \ldots\ldots \quad (3.11)$$

A simple threshold or sign function can be defined as:

$$f(net) = \mathrm{sgn}(net) \begin{cases} 1 & if \quad net \geq 0 \\ -1 & if \quad net < 0 \end{cases} \quad \ldots\ldots \quad (3.12)$$

or

$$f(net) = 1/(1 + e^{-net}) \quad \ldots\ldots \quad (3.13)$$

3.3 Practical Considerations for Improving ANN Training Procedure

We use equations (3.4, 3.9, 3.11) to update the weights in the three-layer ANNs in order to minimize the squared errors in equation (3.2).Practical considerations will be discussed in this section.

Scaling Input: In order to avoid the variations of feature values, the input pattern should be shifted so that the feature vector's values are scaled into domain [0,N].

Firstly, original features are normalized as follows:

$$x'_{i,j} = \frac{x_{i,j} - \min_{k=1,\ldots,l}(x_{k,l})}{\max_{k=1,\ldots,l}(x_{k,j}) - \min_{k=1,\ldots,l}(x_{k,j})} * N \quad \ldots\ldots \quad (3.14)$$

where $x_{i,j}$ is the jth feature of the ith training samples($j=1,\ldots n;\ i=1,\ldots,l$). $x'_{i,j}$ is the normalized feature, l is the total number of training samples and n is the number of features.

Target Values: The target value (the desired output) of the output category is chosen as +1, while others are set equal to 0.0.

Training with Outer Layer: In order to increase the neural network's discriminant ability, some outer layer images are produced and put into the training set. For handwritten numeral recognition and verification, the outer layer images may consist of the part of the characters, touching pairs of the characters, etc.

Number of Nodes in the ANN: According to a convenient rule of thumb, the total number of weights in the net is roughly chosen as *n/10~n/4*. Here *n* is the number of training samples.

Initializing Weights: Random data are generated for all weights in the range of -1.0< all weights <+1.0.

Learning Rates: In general, the learning rate is small enough to ensure convergence. A learning rate of (0.1-0.4) is often adequate as a first choice.

Training Different Patterns: We used the following strategies to train the classifiers: our training procedure concentrates on the "difficult" patterns. Firstly, an ANN classifier is trained on all training samples, then the same set of training samples are fed into the ANN for testing. Those "difficult" patterns, which are not correctly recognized, are copied several times and randomly put into the training set for training again. As more "difficult" patterns are in the training set, the ANN can adaptively learn how to correctly recognize those "difficult" patterns without losing its generality.

Training Imbalanced Data: In Section 5.4, when we design absolute verifiers, for example, verifying numeral 4 (one class) from other nine numerals (another class), even the number of the training samples in each numeral category is approximately equal, the training data can be highly imbalanced for training a two-class classifier. ANN learning from imbalanced training data can result in ignoring the minority class. The back

propagation algorithm can be biased toward the majority class and the convergence of ANN can be very low. In order to overcome above mentioned problems, we used a prior duplication strategy to boost up the minority training data. The training samples in the minority class are copied several times and randomly put into the training set to keep the training data balance.

3.4 What Is a Wavelet?

The simple answer is a "short" wave. For creating a wavelet, two conditions have to be satisfied: First, the wavelet must be oscillatory (wave). Second, its amplitudes must be nonzero only during a short interval. Another definition of a wavelet transformation is that it can be considered as a mathematical tool for waveform representations and segmentations, time-frequency analysis, and fast algorithms for easy implementation [18].

There are two famous signal analysis methods: Fourier analysis and wavelet analysis.

Fourier analysis is a well-known technique of spectral analysis, which uses trigonometric functions. Any finite power signal *f(x)* can be represented as a series of components in the frequency domain. Fourier analysis has been a traditional and efficient tool in many fields of science and engineering.

However, Fourier analysis has its own deficiencies:

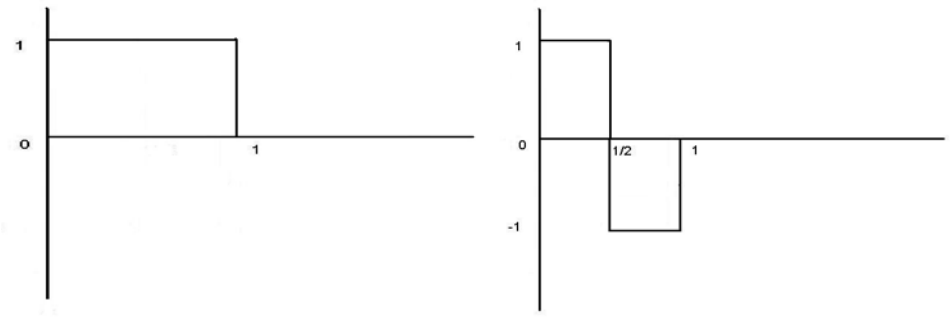
- Fourier analysis cannot show the signals locally in the time domain.
- Fourier expansion can approximate the stationary signals well, but cannot do so for the non-stationary signals.

This makes Fourier transform a less than optimal representation for analyzing signals, images and patterns which contain transient or localized components.

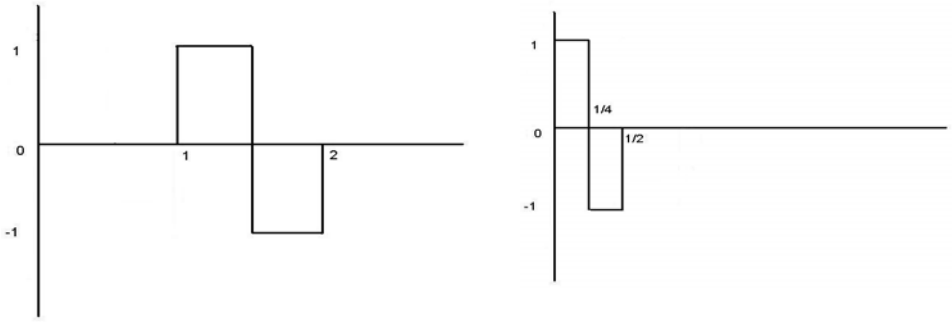
Normally, in pattern recognition analysis, many important features are highly localized in spatial positions.

In order to clearly explain the wavelet, a typical wavelet function (Haar wavelet) is shown in Fig. 3.2.

Haar wavelet is the simplest wavelet. There is a relationship between the scaling function and wavelet function.



(a) Haar scaling function waveform (b) Haar wavelet function waveform



(c) Haar translation waveform (d) Haar dilation waveform

Fig. 3.2 Haar wavelet function

If we find a function $\psi(x)$, which has the dilated and translated formula:

$$\psi(2^j x - k) \mid j,k \in Z$$

$$\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k) \qquad (j,k) \in Z \qquad \ldots\ldots (3.15)$$

This formula may constitute an orthogonal basis of the finite energy signal space $L^2(R)$ *:*

Thus, in the wavelet domain, any finite energy signal *f(x)* can be represented by:

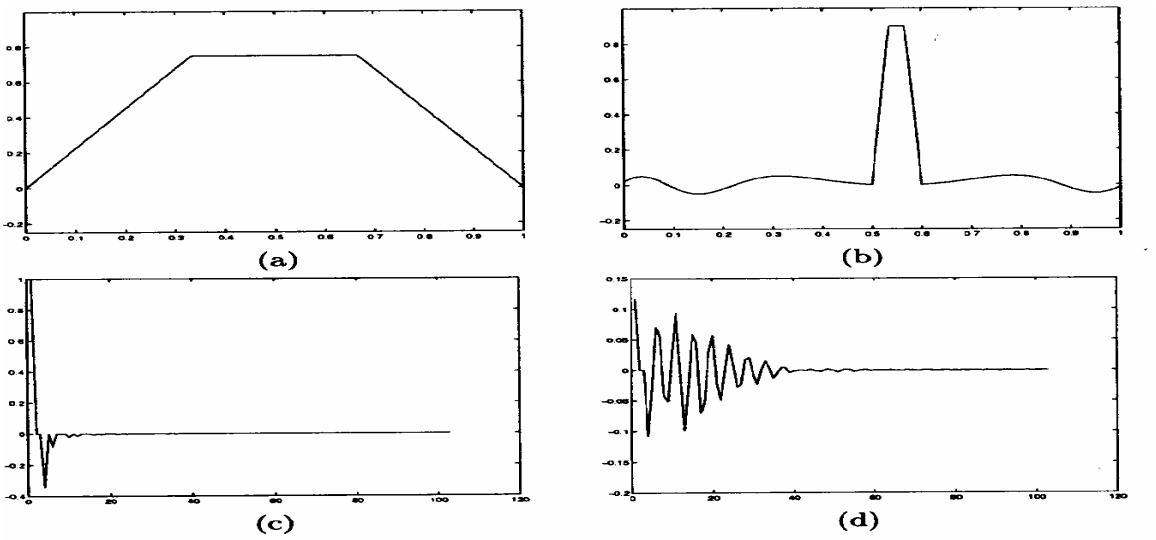
$$f(x) = \sum_{j \in Z} \sum_{k \in Z} w_{j,k} \psi_{j,k}(x)$$

$$w_{j,k} = \int_R f(x) \psi_{j,k}(x)\, dx \qquad \ldots\ldots (3.16)$$

For example, in order to compare the difference between Fourier transform and wavelet transform, two original signals $f_1(x)$ and $f_2(x)$ are shown in Fig. 3.3 (a) and (b). The Fourier Transform results of the two waves are shown in Fig. 3.3 (c) and (d), respectively.

We can observe that $f_2(x)$ possesses a transient component with a very short interval. Its corresponding Fourier expansion contains many terms which produce a long vibration with a long duration.

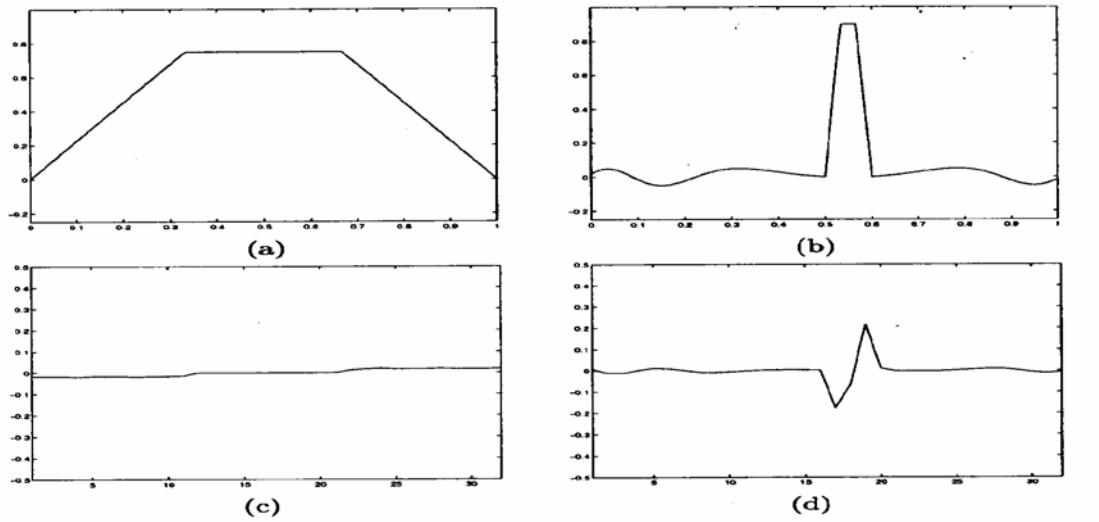
As a comparison to Fourier transform, the two original signals $f_1(x)$ and $f_2(x)$ are drawn in Fig. 3.4 (a-b) again; the Haar wavelet transform results of the two waves are shown in Figs. 3.4 (c-d).



(a) Original signal $f_1(x)$ (b) Original signal $f_2(x)$

(c) Fourier coefficients of $f_1(x)$ (d) Fourier coefficients of $f_2(x)$

Fig. 3.3 Two original signals and their Fourier coefficients



(a) Original signal $f_1(x)$ (b) Original signal $f_2(x)$

(c) Haar coefficients of $f_1(x)$ (d) Haar coefficients of $f_2(x)$

Fig. 3.4 Two original signals and their wavelet coefficients

Note: x-axis represents the x value of $f_1(x)$ or $f_2(x)$; y axis represents the amplitude for $f_1(x)$ or $f_2(x)$ in different functions.

The above coefficients can successfully localize the signals. For signal $f_1(x)$ in Fig. 3.4 (a), nearly all Haar coefficients are close to zero (shown in Fig. 3.4 (c)), which implies that no high frequency is included in the original signal.

By contrast, two peaks of the vibration in Fig. 3.4 (d) correspond to the two positions of the transient components in signal $f_2(x)$ *locally* (shown in Fig. 3.4 (b)).

Each coefficient in the wavelet domain is determined by the local action of the signal.

Compared to Fig.3.3 of Fourier approximation, Haar wavelet has a better approximation for transient signals. However, as Haar wavelet has a saw-tooth waveform, it is not an optimal wavelet kernel. In recent years, many new wavelets have been proposed. Some examples include: Daubechies wavelet, Shannon wavelet, Meyer wavelet, Coiflet wavelet, Symmlet wavelet, etc. [18].

Generally speaking, compared to Fourier transform, wavelet transform has the following features:

1) Better coefficients to approximate

2) Orthogonality (scaling functions (v_j) is orthogonal to wavelet function (w_j))

3) Symmetry (to deal with boundaries)

4) Short compact support: support-FIR filter

The wavelet transform also has some advantages over Fourier transform:

• A complicated signal *f(x)* can be constructed by the linear combination of wavelets, which are produced by the dilations and translations of the basic function.

• The expansion coefficients of a signal using the basic wavelet function can reflect the locations of the transient in the time domain.

• The new basic function $\psi(x)$ and its families can fit transient signal $f(x)$ much better than the Fourier kernel. In other words, they can minimize the error between the approximation of $f(x)$ and signal $f(x)$ itself.

In the next section, we will discuss multiresolution analysis.

3.5 Multiresolution Analysis

Fast Fourier Transform (FFT) is a revolutionary achievement in Fourier transform for signal processing. In the same way, Multiresolution Analysis (MRA) is also a great breakthrough in wavelet analysis in terms of algorithms and calculations. MRA has now become a very important tool in wavelet analysis and has been used in signal processing, pattern recognition, and other related fields.

A function $f(x)$ is projected at each step j on the subset V_j:$(\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots)$.

The scalar project $c_{j,k}$ is defined by the dot product of $f(x)$ with the scaling function $\phi(x)$, which is dilated and translated as follows:

$$c_{j,k} =< f(x), \phi_{j,k}(x) >$$
$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k) \quad \ldots\ldots (3.17)$$

where the difference between c_{j+1} and c_j is contained in the detail component belonging to the space W_j, which is orthogonal to V_j:

$$W_j \oplus V_j = V_{j+1}$$
$$V_j \cap W_j = \{0\}, \quad j \in Z \quad \ldots\ldots (3.18)$$

Suppose $\psi(x)$ is the wavelet function. The wavelet coefficients are obtained by:

$$w_{j,k} =< f(x), 2^{j/2}\psi(2^j x - k) > \quad \ldots\ldots (3.19)$$

Some relationships exist between $\phi(x)$ and $\psi(x)$ as follows:

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_n h(n)\phi(x - n)$$

$$\frac{1}{2}\psi\left(\frac{x}{2}\right) = \sum_n g(n)\phi(x - n) \quad \ldots\ldots (3.20)$$

where $h(n)$ and $g(n)$ represent low-pass and high-pass filters; n is the number of the filtering coefficients; the value of n depends on which wavelets are chosen. The filtering coefficients can be obtained from wavelet books [18, 20, 72, 73, 90].

In other words, the low frequency components and the high frequency components can be obtained directly by computation from $c_{j,n}$ using Equation (3.21):

$$c_{j-1,k} = \sum_n h(n - 2k)c_{j,n}$$

$$w_{j-1,k} = \sum_n g(n - 2k)c_{j,n} \quad \ldots\ldots (3.21)$$

The reconstruction algorithm is shown in Equation (3.22):

$$c_{j,k} = \sum_n h(k - 2n)c_{j-1,n} + g(k - 2n)w_{j-1,n} \quad \ldots\ldots (3.22)$$

A decomposition diagram of one-dimensional signal is given in Fig. 3.5. This diagram shows a scheme of decomposing a signal c_j into low frequency component c_{j-1} and high frequency component w_{j-1} in the next layer, and so on.

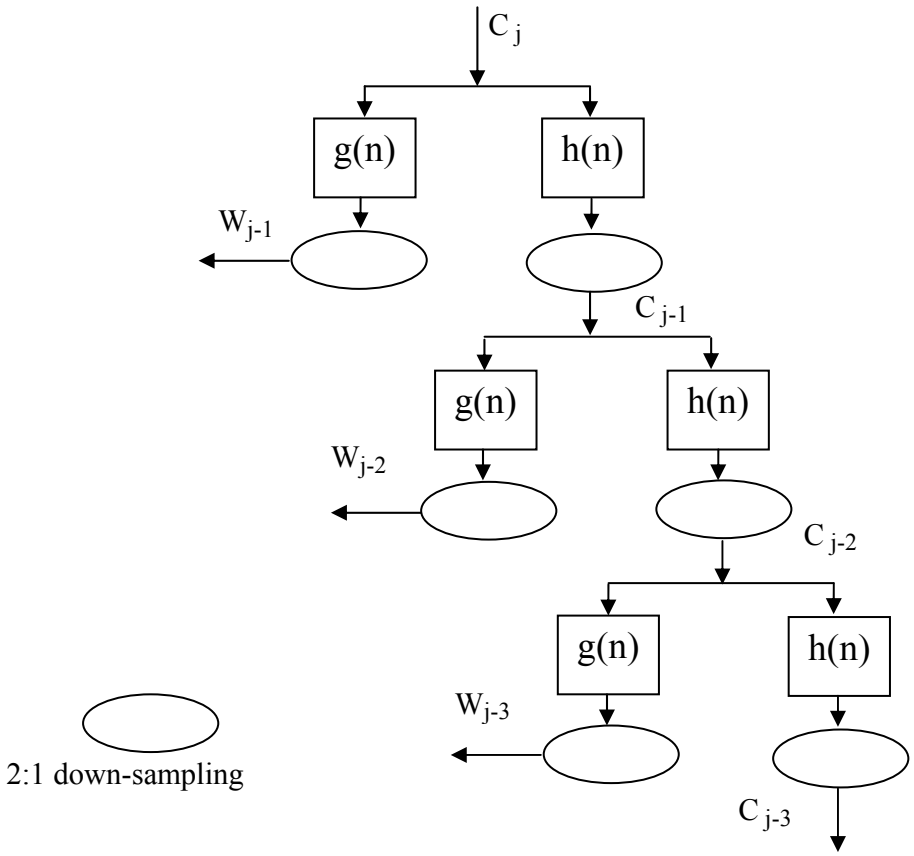


Fig. 3.5 One dimensional signal decomposition

For the filter coefficients, Daubechies D4 wavelet coefficients are listed below [18]:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}};\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}};\ h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}};\ h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}};$$

$$g_0 = h_3;\ g_1 = -h_2;\ g_2 = h_1;\ g_3 = -h_0;$$

Daubechies D4 wavelet will be used in our proposed feature extraction method in the next chapter.

3.6 Two Dimensional Wavelet Structure

In the similar way to 1-D signal analysis, a 2-D image can be decomposed into four components: low-pass rows with low-pass columns (LL); high-pass rows with low-pass columns (HL); low-pass rows with high-pass columns (LH); and high-pass rows with high-pass columns (HH). Mathematically, we can express the recursive algorithm as follows:

$$
\begin{aligned}
A^{(n-1)}_{LL,k_1,k_2} &= \sum_{l_1,l_2} h_{l_1-2k_1} h_{l_2-2k_2} A^{(n)}_{LL,l_1,l_2} \\
D^{(n-1)}_{LH,k_1,k_2} &= \sum_{l_1,l_2} h_{l_1-2k_1} g_{l_2-2k_2} A^{(n)}_{LL,l_1,l_2} \\
D^{(n-1)}_{HL,k_1,k_2} &= \sum_{l_1,l_2} g_{l_1-2k_1} h_{l_2-2k_2} A^{(n)}_{LL,l_1,l_2} \\
D^{(n-1)}_{HH,k1,k2} &= \sum_{l_1,l_2} g_{l_1-2k_1} g_{l_2-2k_2} A^{(n)}_{LL,l_1,l_2}
\end{aligned}
\qquad \ldots\ldots (3.23)
$$

where $\{h_k\}$, $\{g_k\}$ are filter decomposition coefficients relating to scale function $\phi(x)$ and wavelet function $\psi(x)$, which result in various wavelet transformations such as Daubechies, Coiflet, etc. A wavelet decomposition scheme of a 2D image is shown in Fig. 3.6.

In Fig. 3.6, a 2-D image signal can be decomposed into the next layer by convoluting h_k or g_k, and down-sampling (2:1) on the rows, respectively, then convoluting h_k and g_k and down-sampling (2:1) on the columns.

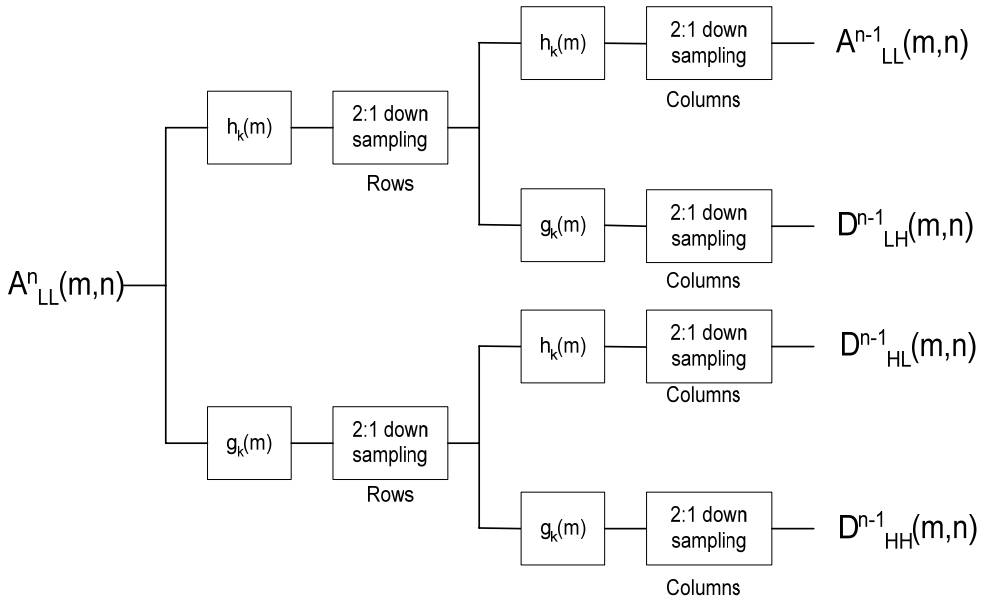


Fig. 3.6 2D image wavelet decomposition scheme

3.7 Fast Algorithm of Wavelet Transform

Wavelet transform can be implemented by designing a pair of Finite Impulse Response (FIR) filters, then by down-sampling (2:1). This decomposition is repeated as a pyramid structure. The decomposed signal can be reconstructed by using the inverse pyramid structure [73]. It is well known that for FFT (Fast Fourier Transform), the computation complicity is $O(n\log_2 n)$; however, a fast wavelet transform takes $O(n)$ operations. Wavelet transform is a versatile tool with a very rich mathematical content and many applications.

Localization and multiresolution analysis are two main properties of wavelet transform, which can be used to extract local and detail features in pattern recognition. The fast algorithm of wavelet transform makes its real-time applications possible. However,

wavelet transform has a 2:1 down-sampling operation, which results in its main drawback: it is sensitive to the shift of the signal. In recent years, researchers proposed a complex wavelet transform to overcome its deficiency. We will discuss complex wavelet in the next section and apply the complex wavelet transform to handwritten numeral feature extraction in the next chapter.

3.8 Complex Wavelet Transform

According to the wavelet theory, a conventional two-dimensional wavelet discrete transform (2D-DWT) can be regarded as equivalent to filter the input image with a bank of filters, whose impulse responses are all approximately given by scaled versions of a mother wavelet. The output of each level consists of four sub-images: LL, LH, HL, HH with 2:1 down-sampling. If the wavelet filters are real and we use Mallat's dyadic wavelet decomposition tree [73], which has a fast algorithm, the coefficients of decomposition will suffer from the lack of shift invariance and poor directional selectivity (only two direction decompositions).

Two dimensional complex wavelet transform (2D-CWT) does not only keep wavelet transform's properties of multi-resolution decomposition analysis and perfect reconstruction, etc., but it also adds its new merits: its magnitudes being insensitive to the small image shifts and multiple directional selectivity, which recently has been used successfully in signal and image processing. 2D-CWT can be implemented using a dual-tree structure [52].

In two-dimensional complex wavelet transform (2D-CWT), we can set the basic functions to closely approximate complex Gabor-like functions, which exhibit strong characteristics of spatial locality and orientation selection, and are optimally localized in the space and frequency domains. Therefore, 2D-CWT functions have the following form:

$$h(x,y) = a(x,y)e^{j(w_x x + w_y y)} \quad \ldots\ldots (3.24)$$

where $a(x,y)$ is a slowly varying Gaussian-like real window function centered at (0,0), and (w_x, w_y) is the center frequency of the corresponding subband. So the complex coefficients of the *i*th subband of the *l*th level can be written as:

$$c_i^l = u_i^l + jv_i^l \quad \ldots\ldots (3.25)$$

The magnitude of each component of each subband is calculated as:

$$C_i^l = \sqrt{(u_i^l)^2 + (v_i^l)^2} \quad \ldots\ldots (3.26)$$

Since $a(x,y)$ is a slowly varying function, the magnitude is insensitive to small image shift.

The directional properties of the 2D-CWT arise from the fact that *h(x,y)* has a constant phase along the lines such that $w_x x + w_y y$ is constant. Complex filters in two dimensions provide true directional selectivity. There are six subband images of complex coefficients at each level, which are strongly oriented at angles of ±15°, ±45°, and ±75°. These two properties are useful for pattern recognition.

2D-CWT can be implemented using a dual-tree structure. For each tree, its structure is similar to 2D-DWT, which has two decomposition operations at each level, namely row

decomposition and column decomposition. However, the different filters of 2D-CWT are applied for perfect reconstruction and the outputs of subband images are congregated into complex wavelet coefficients. Interested readers can refer to reference [52] for further details.

Chapter Four

Feature Extraction and Feature Selection Based on Multi-Class Divergence Analysis

4.1 Feature Extraction

Feature Extraction is a vital step in pattern recognition. In this chapter, seven sets of features are extracted. We use MNIST handwritten digit database, which includes 60,000 training samples and 10,000 testing samples. All the digit images in the MNIST database are grayscale images with 28x28 sizes. In the preprocessing, each 28x28 grayscale digit image is binarized and normalized to a size of 32x32. These feature sets and the methods of extracting them are summarized below:

4.1.1 Feature Set I: Directional-Based Wavelet Features

We use Kirsch nonlinear edge enhancement algorithm to extract statistical features from the characters and apply wavelet transform on these statistical features to form original features.

The directional-based feature extraction is implemented as follows: firstly, the Kirsch nonlinear edge enhancement algorithm is applied to an NxN character image to extract horizontal, vertical, right-diagonal and left-diagonal directional features and global features; then 2-D wavelet transform is used to filter out the high frequency components of each directional feature image and character image, respectively, and to convert the feature matrix into a 4x4 matrix.

Suppose that we define the eight neighbors of pixel (i,j) as follows:

A0 A1 A2

A7 (i,j) A3

A6 A5 A4

Fig. 4.1 Definition of eight neighbors of pixel (i,j)

Kirsch defined a nonlinear edge enhancement algorithm as follows:

$$G(i,j) = \max\ \{1, \max\nolimits_{k=0}^{7}[|\ 5S_k - 3T_k\ |\} \quad \ldots\ldots (4.1)$$

where

$$S_k = A_k + A_{k+1} + A_{k+2}$$

and

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7} \quad \ldots\ldots (4.2)$$

In order to extract four directional features from horizontal (H), Vertical (V), Right-diagonal (R) and Left-diagonal (L) directions, we can use the following templates:

$$\begin{aligned} G(i,j)_H &= \max(|\ \ 5S_0 - 3T_0\ |, |\ 5S_4 - 3T_4\ |), \\ G(i,j)_V &= \max(|\ \ 5S_2 - 3T_2\ |, |\ 5S_6 - 3T_6\ |), \\ G(i,j)_R &= \max(|\ \ 5S_1 - 3T_1\ |, |\ 5S_5 - 3T_5\ |), \\ G(i,j)_L &= \max(|\ \ 5S_3 - 3T_3\ |, |\ 5S_7 - 3T_7\ |). \end{aligned} \quad \ldots\ldots. (4.3)$$

Apply Daubechies-4 wavelets to four directional feature matrices and the character image, and only keep 4x4 low frequency components of each as features. The schematic diagram of the directional-based wavelet feature extraction is shown in Fig. 4.2. In total, 16x5=80 features can be extracted from each character.

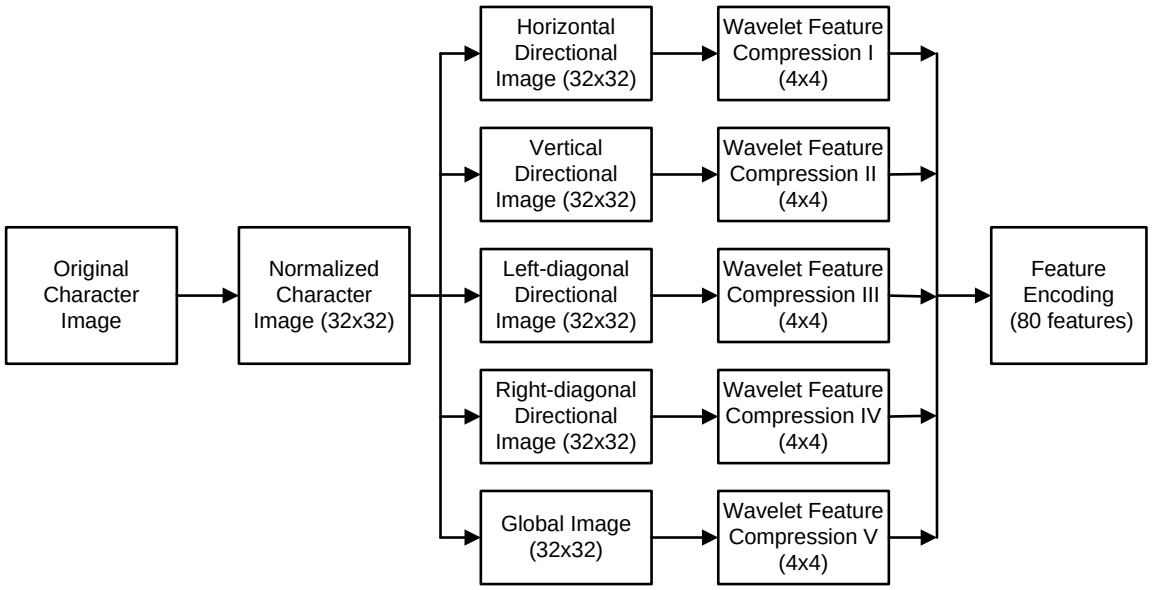


Fig. 4.2 Schematic diagram of gradient based wavelet feature extraction

4.1.2 Feature Set II: MAT based Gradient Features

As we know, a grayscale image has richer information than a binary image for discrimination. A grayscale image can be created by two methods: it can be scanned by a real grayscale character image, or it can be created through a pseudo-grayscale image, which can be produced by using Medial Axial Transformation Algorithm (MAT). In this thesis, we use the second method.

MAT is a method of finding a binary image centre skeleton and at the same time, it changes a binary character image into a grayscale character image with maximum values on the central skeleton of the character. This method has the following advantages:

(1) The algorithm can change a binary image into a grayscale image, which has richer information for image processing and pattern recognition;

(2) The transformed character image highlights the centre skeleton of the character strokes with maximum grayscale values and keeps stroke information and local information.

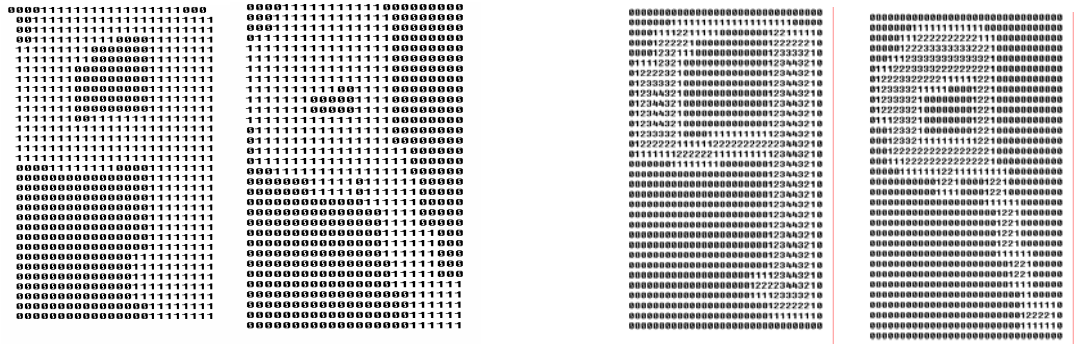
Our iterative MAT algorithm is implemented as follows:

1(a) Design the structure matrix of erosion as a 3x3 matrix *E* with all elements being set to 1 and set the initial iteration number as 1;

1(b) Erode an *NxN* character image ***Im*** by the morphological erosion operator *E*, and the value of the eroded pixel in the character image is set equal to the current iteration number;

1(c) Increase the iteration number by 1, then repeat step 2 until no more new eroded pixels are created.

Figs. 4.3 (a-b) show two binary images of Character "9" selected from MNIST. Figs. 4.3 (c-d) is the MAT transformed character images of Figs. 4.3 (a-b), respectively.



(a) (b) (c) (d)

Fig. 4.3 Two binary character images (a,b) and their pseudo-grayscale character images after MAT (c,d)

After getting the MAT transformed images, we can use the following steps to extract MAT gradient-based features:

1(b) Normalize the MAT image with its pixel values from 0.0~1.0;

2(b) Convolute the normalized character image ***Iz*** with Sobel operators to generate the amplitudes and phases of the gradient image.

The templates of the Sobel operators ***Sx*** and **Sy** are listed in Tables 4.1 and 4.2.

Table 4.1 Template of Sobel operator ***Sx***

-1	0	1
-2	0	2
-1	0	1

Table 4.2 Template of Sobel operator ***Sy***

1	2	1
0	0	0
-1	-2	-1

The X-gradient character image can be calculated by:

$$\boldsymbol{Ix=Iz*Sx} \qquad \ldots\ldots \quad (4.4)$$

and the Y-gradient character image is calculated by:

$$\boldsymbol{Iy=Iz*\ Sy} \qquad \ldots\ldots \quad (4.5)$$

The gradient magnitude and phase are then obtained by:

$$r(i,j) = \sqrt{Ix^2(i,j) + Iy^2(i,j)}$$

$$\theta(i,j) = \tan^{-1}\frac{Iy^2(i,j)}{Ix^2(i,j)} \qquad \ldots\ldots \quad (4.6)$$

2(c) Count the gradient direction of each pixel of the convoluted image with nonzero gradient magnitude values as a direction feature.

In order to generate a fixed number of features, each gradient direction is quantized into one of eight directions at $\pi/4$ intervals. Each normalized gradient image is divided into

16 sub-images. The number in each direction of each sub-image is counted as a feature. In total, the number of features is 4x4x8=128.

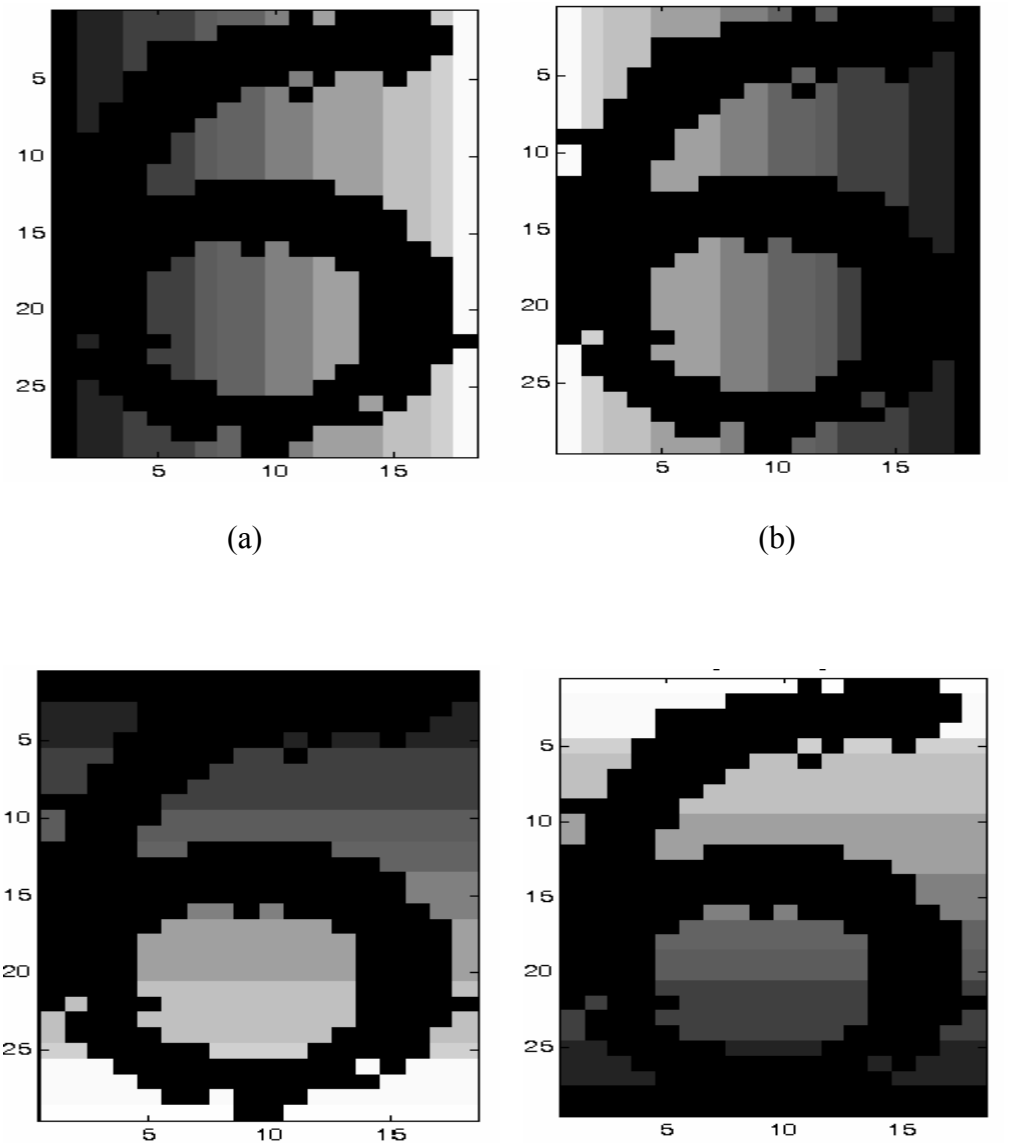
A scanned image can be affected by shadows, non-uniform illumination, and low contrast. Fig. 4.4 shows four simulated grayscale images affected by non-uniform illumination from four directions ((a) from right to left; (b) from left to right; (c) from bottom to top; (d) from top to bottom).

If we do feature extraction on the four grayscale images of Fig. 4.4 (a-d), the different feature vectors will be obtained from those grayscale images. Namely, the features extracted from the four non-uniform illumination images are totally different. However, if we carry out the following operations, the feature extraction will be insensitive to the non-illumination. The procedure is listed as follows:

1) Change a non-uniform illumination grayscale image into a binary image;
2) Apply MAT transform to the binarized image to form a pseudo-grayscale image;
3) Extract gradient-based features

For example, the four non-uniform illumination images shown in Fig. 4.4 can produce the same MAT grayscale image, which is shown in Fig. 4.5.

The four non-uniform illuminated grayscale images can produce the same MAT pseudo-grayscale image, therefore, the MAT transformed image is insensitive to the variation of the illumination.



(a) (b)

(c) (d)

Fig. 4.4 Grayscale character images affected by non-uniform illumination

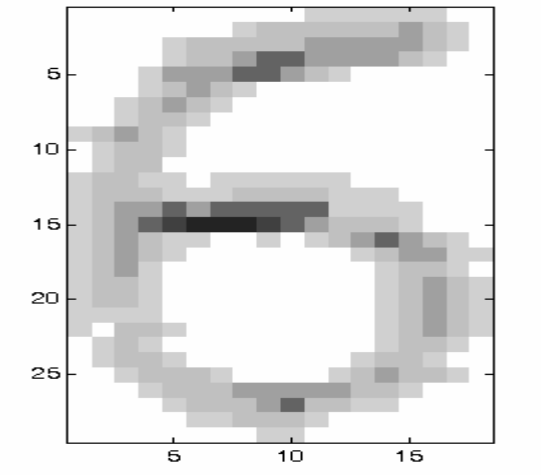


Fig. 4.5 MAT transformed image from non-uniform illumination grayscale images of Fig. 4.4 (a-d).

4.1.3 Feature Set III: Complex Wavelet Features

As we know a real 2D wavelet transform suffers from the following problems: lack of shift invariance and poor directional selectivity [52]. (2D-CWT) overcomes these deficiencies. Our experiments have demonstrated that the 2D-CWT features for handwritten numeral verification can make an ANN classifier more reliable and it can converge more easily.

Fig. 4.6 shows our proposed 2D-CWT feature extraction scheme for the recognition and verification of handwritten numerals. The dual-tree complex wavelet decomposition consists of two trees: Tree A and Tree B which have the same structure. In order to realize a perfect reconstruction from the decomposed subimages, a lowpass filter and a highpass filter at the first level need to be specially designed and denoted as Lop1 and Hip1 for tree A; Lop2 and Hip2 for tree B. Those special filters are called pre-filters. The

other complex filters in the higher levels are set to Lo1 and Hi1 for Tree A, Lo2 and Hi2 for tree B.

A character image of size *NxN* is decomposed into four subband images: LL, LH, HL, HH at the first level of each tree and each of the subband images has a size of $\frac{N}{2} x \frac{N}{2}$. At each higher level, the decompositions are based on the LL subband image at the previous level. For example, if a 32x32 character is decomposed into the third level, the final size of each subband image is 4x4. Then we can extract the complex wavelet coefficients as features.

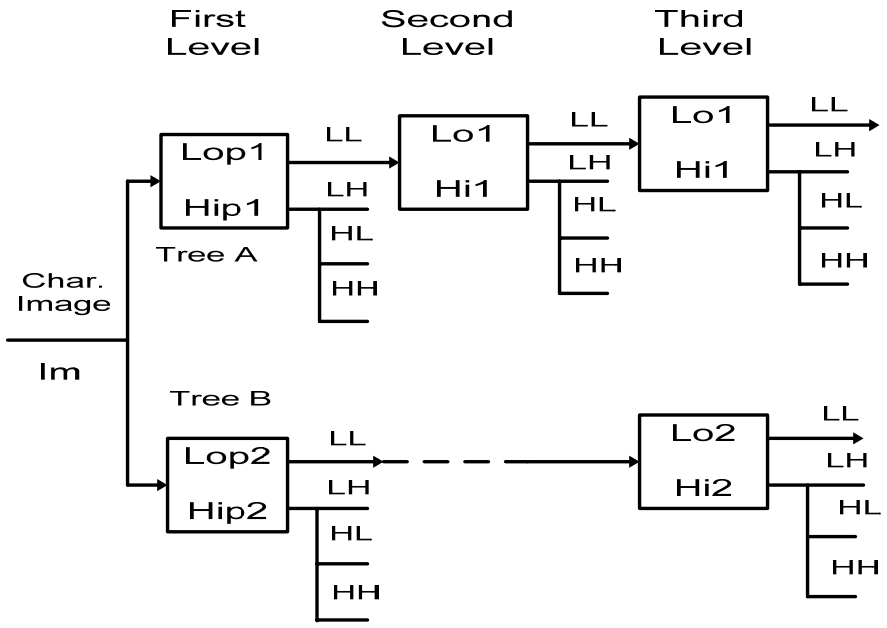


Fig. 4.6 The schematic diagram of 2D-CWT for character feature extraction

The feature extraction is conducted at the third level. We only keep amplitude coefficients for the three high frequency components and both amplitude and phase information for the low frequency component. The number of features = 4x4 (for each

subband image) *3 (high frequency subband images for each tree) *2 (trees) +4x4 (for each subband image) *2 (trees)* 2(parts: real and imaginary) =160. As the real and imaginary coefficients of each LL subband image are extracted as features, the phase information is preserved with a good directional selectivity.

4.1.4 Feature Set IV: Binary Image Gradient Features

This feature extraction method is the same as that of MAT-based Directional Features except that no MAT transform is needed. The gradient features are extracted directly from the binary character image. A feature vector of 128 is extracted for each handwritten character image.

4.1.5 Feature Set V: Median Filter Gradient Features

Three steps are required to extract the median filter gradient features:

1) To convolute a character image ***Im*** by a 2D median filter;

The template of the 2D median filter is listed in Table 4.3

Table 4.3 Template of 2D median filter

1	2	1
2	4	2
1	2	1

2) To use Robert operators on the median-filtered image to generate the amplitudes and phases;

The templates of the Robert operators are listed in Tables 4.4 and 4.5.

Table 4.4 Template of Robert operator ***Rx***

0	0	0
0	1	0
0	0	-1

Table 4.5 Template of Robert operator ***Ry***

0	0	0
0	0	1
0	-1	0

3) To count the gradient direction of each pixel with nonzero gradient magnitude values as a direction feature. So the total number of features is 128.

4.1.6 Feature Set VI: Image Thinning Distance Features

In a feature set, the distance features in both horizontal and vertical directions are extracted as follows: firstly, an ***NxN*** character image is thinned and the thinned image is scaled into an 8x8 array. The thinned image is scanned both horizontally and vertically, respectively. In the horizontal scanning, for each pixel in the 8x8 thinned image, if the value of the pixel is 0 (black), then the distance is 0; otherwise, the distance is set to the distance from that pixel to the nearest black pixel in both horizontal directions on the scanning line. For any pixel, if there are no nearest black pixels in both directions, the distance of the pixel is set to the distance from the pixel to one of two edges, whichever has longer distance to the edge.

In the vertical scanning, the same algorithm is applied. The distance features are normalized to [0.0, 1.0]. In total, there are 128 features.

4.1.7 Feature Set VII: Geometrical Features

In order to explore a character's geometric features, we use concave features on the character's four profiles; middle line features; horizontal segment features on the left and right profiles; character width features in the top five rows; middle ten rows; and bottom five rows; as well as endpoint and crossing point features. These geometrical features are encoded as 20 features.

4.1.8 General Purpose Recognizer and Its Recognition Performance Using Hybrid Features

In handwritten character recognition, it is common sense that the recognition model consists of one General Purpose Recognizer (GPR) and various verifiers in order to boost the recognition rate. Fig. 4.7 shows such a recognition and verification system.

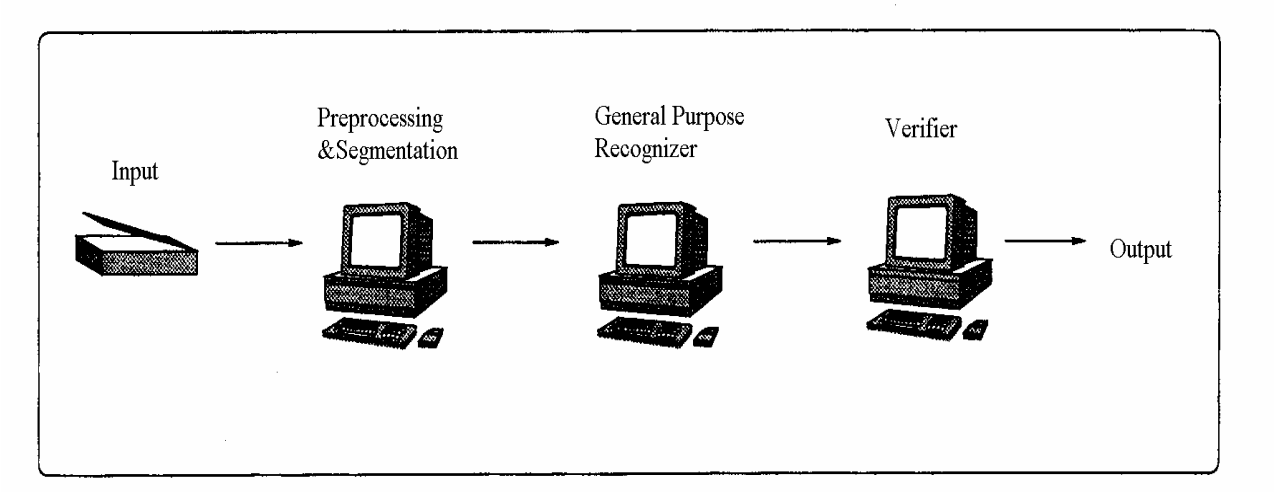


Fig. 4.7 A recognition and verification system

The verification of confusing handwritten character pairs is a challenging task. It is also one of our research goals in this thesis. In Chapter 5, we will discuss this issue in detail. In this section, the recognition performance of a GPR using our proposed hybrid feature sets will be discussed. Conceptually speaking, the first six feature sets are statistical features. The last one is a structural feature set. In order to complementarily congregate these feature sets for the recognition of handwritten numerals, we conducted a series of recognition experiments based on the combinations of feature sets.

A three-layer ANNs with Back Propagation (BP) algorithm was used as a classifier with the following configurations:

No. of nodes in the input layer: No. of features

No. of nodes in the hidden layer: 150

No. of nodes in the output layer: 10 (representing 10 digits)

No. of training samples: 60000 (MNIST database)

No. of testing samples: 10000 (MNIST database)

Table 4.6 lists the recognition rates using different feature sets

Table 4.6 List of Feature sets and their recognition accuracies

Name of Feature Set	No. of Features	Recognition Accuracy (%)
Feature Set A (I+VII)	100	98.55
Feature Set B (II+VII)	148	98.58
Feature Set C (III+VII)	180	98.30
Feature Set D (IV+VII)	148	98.47
Feature Set E (V+VII)	148	98.40
Feature Set F (VI+VII)	148	97.55
Feature Set G (I+(1/2)VI)	164	98.49

Notes: The recognition rates range from (98.58%~97.55%).

In order to use verification procedure (which will be discussed in the Chapter 5) to increase recognition performance, we use Feature Set B as an example of error analysis:

In the Feature Set B testing result, the recognition rate is 98.58%. It means that out of 10,000 testing samples, there are 142 errors, of which, 40 testing samples are not voted into the top two ranks. This means that the general recognizer does not correctly vote for the recognized character in the top two ranks with the largest or second largest confidence values. The verification model cannot correct these errors any more.

The remaining 102 errors occur in such conditions that: the general recognizer votes for another digit as the first output candidate, so the recognized character is misrecognized; however, the general recognizer votes for the recognized character as the second candidate with the second largest confidence values. Therefore, some of the errors in this category can be corrected in the verification model.

4.2 Feature Selection with Divergence Criterion

Divergence distance measurement is one of the feature selection criteria. Intuitively, if the features show significant differences from one class to the other classes, the classifier can be designed more efficiently with a better performance [83]. For our hybrid features, there are 772 features. We need to develop some methods to keep powerful discriminant features and at the same time to delete the less useful features in order to easily use random forest feature selection.

4.2.1 Divergence Criterion

In order to get the Gaussian-like distribution for each feature y, it is transformed to a new feature x by the following formula:

$$x = y^{0.5}$$

A commonly used distance measure density, and for its connection with information theory, is the Kullback-Leibler distance:

$$KL(C_i, C_j) = \int_R p(x/C_i) \log \frac{p(x/C_i)}{p(x/C_j)} dx \quad \ldots\ldots (4.7)$$

where $1 \le i,j \le K$, and K is the number of classification categories. In order to simplify its computation, a new version of distance divergence criterion based on Kullback-Leibler's symmetric measurement of divergence is introduced in [4]:

$$D_{i,j} = KL(C_i, C_j) + KL(C_j, C_i) \quad \ldots\ldots (4.8)$$

The divergence is another criterion of class separability. Thus, the divergence is defined as:

$$D_{i,j} = \int_\Omega p(x/C_i) \log \frac{p(x/C_i)}{p(x/C_j)} dx + \int_\Omega p(x/C_j) \log \frac{p(x/C_j)}{p(x/C_i)} dx \quad \ldots\ldots (4.9)$$

For a multi-variable normal distribution, if we assume that the conditional probability of C_i class is a normal distribution: $N_x(u_i, \Sigma_i)$ and that of C_j is $N_x(u_j, \Sigma_j)$. For a feature vector x, the following formula will hold:

$$\int_\Omega p(x/C_i) \log \frac{p(x/C_i)}{p(x/C_j)} dx = \frac{1}{2} \log \frac{|\Sigma_j|}{|\Sigma_i|} + \frac{1}{2} tr\Sigma_i(\Sigma_j^{-1} - \Sigma_i^{-1}) + \frac{1}{2} tr\Sigma_j^{-1}(u_i - u_j)(u_i - u_j)^T$$

..... (4.10)

For simplicity, due to the difference in means between two classes, the divergence $K(C_i,C_j)$ can be written as:

$$K(C_i, C_j) = \frac{|u_i - u_j|^2}{2\delta_j^2} \quad \ldots\ldots (4.11)$$

Therefore, the overall divergence can be defined as:

$$D_{i,j} = \frac{|u_i - u_j|^2}{2\delta_j^2} + \frac{|u_j - u_i|^2}{2\delta_i^2} \quad \ldots\ldots (4.12)$$

4.2.2 Divergence Criterion for Feature Selection in the Multi-class Classification Problem

Although the divergence criterion provides a way to measure the distance between two classes, we can extend it to the multi-class case. In the multi-class pattern recognition, there are N classes, each represented by w_i. The domain of the multi-class case can be denoted as $W=\{w_1,w_2,w_3,\ldots, w_N\}$. Each class w_i has M_i samples in the training database. A set of samples from the class w_i is denoted by $S_i = \{s_i^1, s_i^2, s_i^3, \ldots, s_i^{M_i}\}$. A feature vector consists of d features: $Y = (y_1, y_2, y_3, \ldots y_d)$.

For each feature y_j, we can extract M_j feature values for each of the N classes, which are denoted by $X_j^n = (x_{j,1}^n, x_{j,2}^n, \ldots x_{j,k}^n, \ldots, x_{j,M_j}^n)$. Here, $j=1,2,3,\ldots, d,$ where d is number of features; $n=1,2,3,\ldots,N,$ where N is the number of classes; $k=1,2,3,\ldots, M_j$, and M_j is number of training samples in the jth training class.

The mathematical expectation value and variance of each sub-feature vector for each class is denoted as follows:

$$u_{j,n} \equiv \varepsilon[X_j^n] = \frac{1}{Mj}\sum_{k=1}^{M_j} x_{j,k}^n$$

$$\delta_{j,n}^2 \equiv \varepsilon[(X_j^n - u_{j,n})^2] = \frac{1}{M_j}\sum_{k=1}^{M_j}(x_{j,k}^n - u_{j,n})^2 \quad \ldots\ldots (4.13)$$

In the above two equations, $u_{j,n}$ and $\delta_{j,n}^2$ represent the expected value and variance of the *j*th feature in the *n*th class obtained from the training set.

According to our analysis in the previous section, we can calculate the divergence coefficient for each feature *m* based on *N* classes.

$$D(m) = \sum_{l=1}^{N} \sum_{r=1}^{N} D_{l,r}(m) * (1 - p_{l,r}(err)) \quad \ldots\ldots \quad (4.14)$$

$$D_{l,r}(m) = \frac{|u_{m,l} - u_{m,r}|^2}{2\delta_{r,m}^2} + \frac{|u_{m,r} - u_{m,l}|^2}{2\delta_{l,m}^2} \quad \ldots\ldots \quad (4.15)$$

where $p_{l,r}(err)$ is the misrecognition rate of the samples in the *lth* class, being recognized as the *rth* class. We can obtain it by training classifiers using the training samples, and testing classifiers on the test set without any feature selection beforehand. In Equations (4.14 & 4.15), less weight is put on those class pairs, which are inclined to cause more errors. As a result, those with more easily misrecognized classes will have less power to dominate the divergence coefficients for feature selection, thereby decreasing the recognition errors and improving the recognition performance.

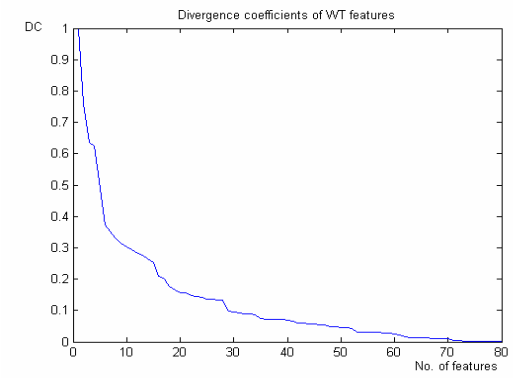
Fig. 4.8 (a, b, c) shows the distribution of the divergence coefficients for three feature sets:

Feature Set I: Directional-Based Wavelet Features

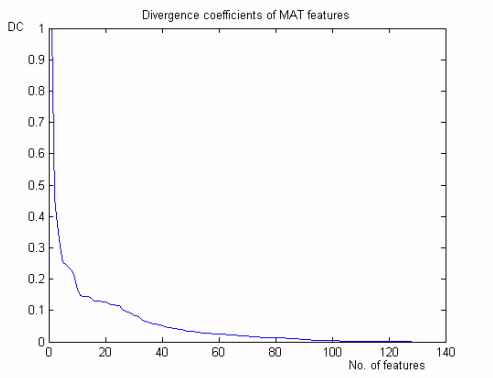
Feature Set II: MAT Gradient-Based Features

Feature Set III: Complex Wavelet Features

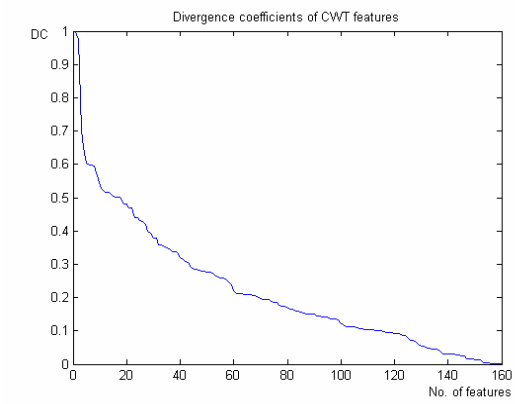
The features are ranked by the divergence coefficients from largest to smallest, not by the order of the features. We can keep those features with larger divergence coefficients and delete the features with less divergence coefficients.



(a) Distribution of feature set I



(b) Distribution of feature set II



(c) Distribution of feature set III

Fig. 4.8 Distribution of feature sets I, II, and III

Note: (DC: divergence coefficients)

The extracted feature number can be chosen based on the following rule: we may add divergence coefficients in decreasing order until the accumulation exceeds a certain percent of the total sum; then we may set that number of the divergence coefficient order as a feature number to retain. Alternatively, we may retain the divergence coefficient that is greater than the given proportion of the largest divergence coefficient and we may set the retained feature number.

From Fig. 4.8 (a, b, c), it can be observed that Feature Set II: MAT gradient-based feature set has the best convergence property. It indicates that for Feature Set II, more discriminate information is accumulated in a few feature components compared to Feature Sets I and III. Our experiments have demonstrated that Feature Set II has the highest recognition rate for an ANN classifier.

In order to reduce the dimensions of features by deleting some less useful or no information features, we retain the different number of features for each original feature set according to its divergence distribution. In total, 450 features are kept for random feature selection.

4.3 Random Feature Selection

Given a large number of features, especially consisting of different sets of features, most likely, from the information theoretical point of view, the features from different sets are complementary because they are extracted in different ways. In order to increase recognition rate and reliability, we design several classifiers. Each individual classifier can be trained by only a subset of all features. The subset of features is randomly chosen from the entire set of features.

After deleting less useful features, we use the Random Feature Selection method (RFS) to construct three new feature sets, by randomly choosing feature components from the seven newly ranked feature sets. The three new random feature sets are called Random Feature Set I (200), Random Feature Set II (218), and Random Feature Set III (240). The number in the bracket is the number of dimensions. Unlike other selection schemes,

which only select the features from the ranked feature sets, the random scheme can produce more than one set of features. Some feature components can be overlapped.

The systematic diagram is shown in Fig. 4.9

We conducted recognition experiments on the general purpose recognizers for the recognition of handwritten numerals by using the three new randomly selected hybrid feature sets. The ANN classifier was used for classification. As expected, the recognition results were better than any of the seven original feature sets reported in section 4.1.8.

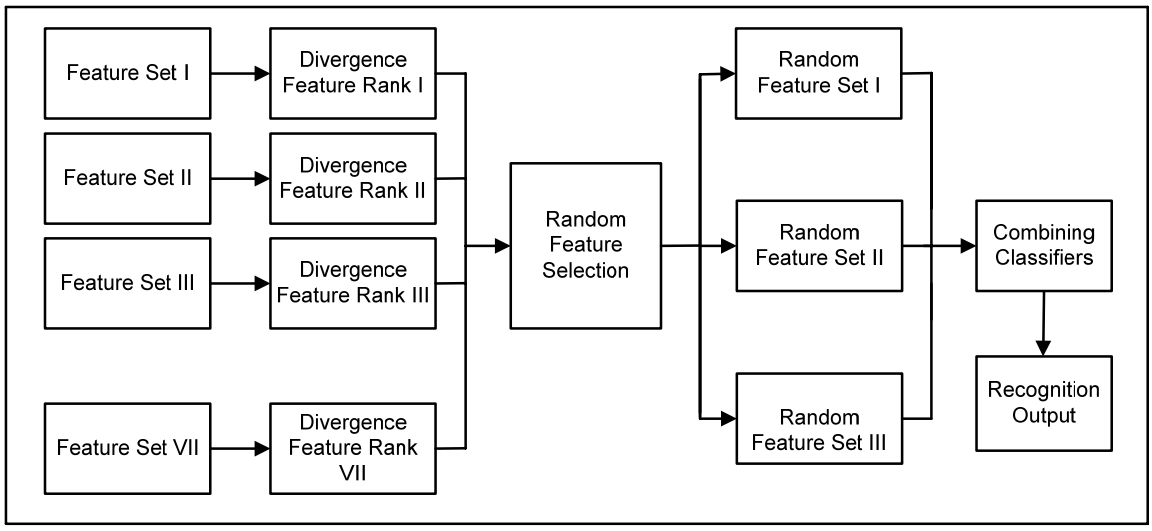


Fig. 4.9 The schematic diagram of hybrid feature extraction and random feature selection

Further experiments on the combination of three ANN classifiers using three random feature sets were conducted. The majority vote was used in the experiments. The overall recognition rate was 99.16%.

The recognition results using three sets of randomly selected hybrid features are listed below:

Feature Set	Recognition Rate
Classifier I (Random Feature Set I: 200)	99.11%
Classifier II (Random Feature Set II: 218)	98.90%
Classifier III (Random Feature Set III: 240)	99.12%
Combination of Classifiers I, II, III (Majority Vote)	99.16%

Note: The number in the bracket after Feature Set I, or Feature Set II, or Feature Set III is the number of the feature dimension.

From these experiments, three randomly selected hybrid features showed the higher recognition performances compared to the seven original feature sets. As the dimensions of the feature sets are still high (>100), we will propose a feature dimensionality reduction method for handwritten numeral verification in the next chapter.

Chapter Five

Feature Dimensionality Reductions for Handwritten Digit Verifications[1]

In this chapter, we propose a novel multi-modal analysis to reduce feature dimensionality and we successfully use our proposed method to compress large-scale features for the verification of confusing handwritten character pairs without losing classification discrimination ability. Firstly, the k-means algorithm is applied to each class, which divides each class data into several clusters. Then, both the within-class scatter matrix and the between-class scatter matrix of the multi-modal data are calculated based on cluster information and decision boundary information. Finally, feature vectors are formed based on the optimal discriminant criterion.

5.1 Discriminant Analysis Criterion for Feature Dimensionality Reduction

In a large feature set, the correlation of features is complicated. Retaining informative features and eliminating redundant ones is a recurring research topic in pattern recognition. Generally speaking, feature extraction and dimensionality reduction serve two purposes: (1) to improve the training and testing efficiency, (2) to improve the reliability of a recognition system.

[1] This work was published in the Pattern Analysis and Applications, Vol. 7, No. 3, Dec. 2004, pp. 296-307.

There are two methods of feature dimensionality reduction. One is feature selection. Another method is to use an optimal or sub-optimal transformation to achieve feature dimensionality reduction. The latter is an information congregation operation, and this study deals with it. As mentioned in chapter one, the nonparametric discriminant analysis is a useful and efficient way in statistical pattern recognition. The within-class $\boldsymbol{S_w}$ matrix, between-class $\boldsymbol{S_b}$ matrix, and mixture scatter $\boldsymbol{S_m}$ matrix have all been used for optimal discriminant analysis, which were defined in Equations 2.2-2.5 in chapter one.

In order to implement a linear mapping from an N-dimensional feature vector to an M-dimensional feature vector ($M \leq N$), one optimal discriminant analysis criterion, which is derived from Fisher discriminant analysis [26], has been defined in Equation 2.6. Here, we rewrite it in Equation 5.1 as follows:

$$J = tr(S_w^{-1} S_b) \qquad \ldots\ldots \quad (5.1)$$

This criterion aims at minimizing within-class separability and at the same time, maximizing between-class separability in order to achieve the best discriminant ability for pattern recognition.

5.2 Multi-Modal Discriminant Analysis for Dimensionality Reduction

As handwritten digits vary a lot in writing styles, each class of numerals can be represented by multi-modals. Therefore, in this section, we will conduct multi-modal discriminant analysis for dimensionality reduction.

For a two-class problem, let X be an N-dimensional feature set obtained from training samples, under the hypothesis: Hi: $X \in w_i$, i=1,2. The recognition decision can be made according to Bayes' decision rule:

If $P(w_1)P(X \mid w_1) > P(w_2)P(X \mid w_2)$,

then $X \in w_1$,

otherwise $X \in w_2$

where $P(X \mid w_i)$ is a conditional density function; P_i is the probability of class i, i=1,2.

Let $h(X) = \frac{P(X|w_1)}{P(X|w_2)}$ and $t = \frac{P(w_2)}{P(w_1)}$,

then $X \in w_1$ if $h(X)>t$, else $X \in w_2$.

We will find a subspace Φ, with the minimum dimension M ($M \leq N$) and the spanning vector $\{\phi_i\}$ of the subspace such that for any observation X

$$(h(X)-t)(h(Y)-t) > 0 \qquad \ldots\ldots (5.2)$$

where Y is an approximation of X in the basis of the subspace. The meaning of expression (5.2) is that the classification result of Y is the same as the classification result of X. In practical applications, features can be selected in such a way as to maximize the number of observations for expression (5.2) while keeping the dimensionality of Y as small as possible.

The decision boundary is defined as $\{X \mid h(X) = t\}$. A decision boundary can be a line, a plane, a curved surface or a curved hyper-surface. Although a decision boundary can be extended to infinity, in most cases, the effective decision boundary is the region where most of the data are located and can be classified [66]. In that sense, the classification criterion should be chosen by the data on and nearthe decision boundary. Those training

data, which are located far from the effective decision boundary, will play little or no role in classification.

Hybrid features are drawn from a wide mixture of different kinds of features, such as geometrical features, wavelet features, etc. The hybrid features are extracted by different methods, some of which can be complementary. In addition, handwritten characters have a variety of writing styles. For instance, different handwritten writings of “4” and “6” are shown in Fig. 5.1. Therefore, the distributions of hybrid features are likely multi-modals.

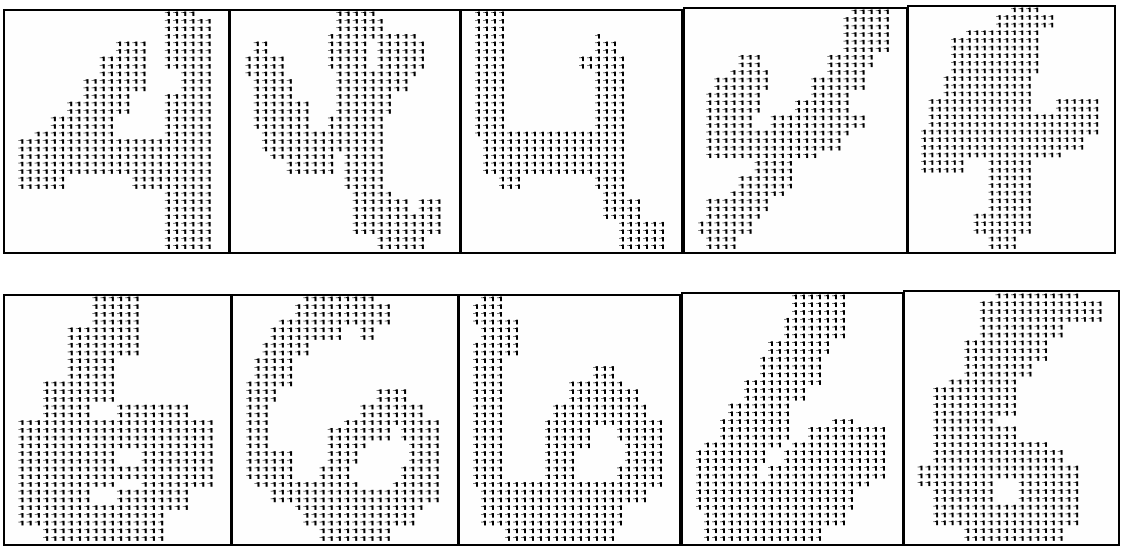


Fig. 5.1 Different writing styles of handwritten “4” and “6”

Fig. 5.2 shows a simple illustrative example of multi-modal data distribution.

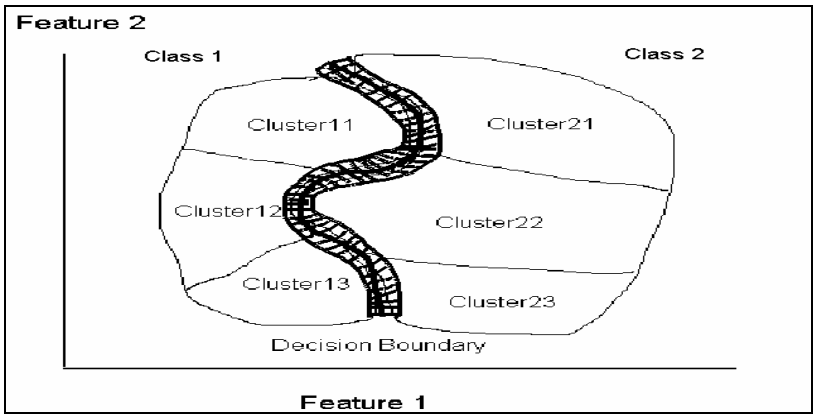


Fig. 5.2 Multi-modal data distribution

The *effective decision boundary* is highlighted by the masked area in the middle of Fig. 5.2, which is located at the intersection of classes w_1 and w_2. As the distribution of each class along the decision boundary is a complex curvature, we divide the training data of each class into several categories, such as clusters 11, 12, 13 for class 1 and clusters 21, 22, 23 for class 2 by using k-means or other clustering algorithms for a better estimation of the within-class scatter matrix and the between-class scatter matrix. The number of clusters in each class is selected from the writing styles of the training handwritten numerals. We can use an unsupervised clustering method [32] to get the number of clusters for each handwritten numeral beforehand.

It is true that there are cases such that using one modal of features fails to represent the best feature vectors or even good features. Hence, we propose a multi-modal discriminant analysis method for feature dimensionality reduction.

For a two-class classification problem: class w_1 and class w_2, each class is divided into several clusters, denoted as $\{C_{1i} \mid i=1,2,\ldots,r_1\}$ for w_1; and $\{C_{2i} \mid i=1,2,\ldots,r_2\}$ for w_2. We calculate the within-cluster scatter matrix of each cluster and congregate it to form the within-class scatter matrix S_w as follows:

$$S_w = \sum_{i=1}^{2} \sum_{j=1}^{r_i} \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} (X_{ijk} - E(X_{ij}))(X_{ijk} - E(X_{ij}))^T \quad \ldots\ldots (5.3)$$

where $X_{ijk} \in C_{ij}; i = 1,2; j = 1,2,..., r_i$; $N_{ij:}$ the number of training samples in C_{ij} ; and $E(X_{ij})$ is the mean of all training samples in C_{ij}.

As each class has been separated into r_i clusters, the training data in each cluster are more centralized. As a result, S_w will be less scattered than the mono-modal case.

In order to find the between-class scatter matrix S_b, first, for each cluster *j* of class one, we find *k-NN* data in cluster *l* of class two, with *k-NN* distance corresponding to the mean of cluster *j* in class one. The *k-NN* data can be denoted as:

$$\{X_l^{knn}(C_{1j}) \mid j = 1,2,..., r_1; l = 1,2,..., r_2\}$$

Similarly, we calculate *k-NN* data for each of r_2 clusters of class two. The *k-NN* data in class one corresponding to each cluster in class two is denoted as:

$$\{X_l^{knn}(C_{2j}) \mid j = 1,2,..., r_2; l = 1,2,..., r_1\}$$

Here $X_l^{knn}(C_{ij})$ represents the *k-NN* data of the *lth* cluster in the class other than *i* class, which is related to the *jth* cluster of class *i,* where *i=1,2.* For the verification experiments of handwritten numeral pairs, we do not have any theory on choosing the number of *k-NN's*. We set the number of *k-NN's* equal to min (N_{ij}/10, 50), according to the experimental results.

The above definition guarantees that the extracted *k-NN* data are located on and near the effective decision boundary and we will therefore use these data to compute the between-class scatter matrix. The following is a definition of an adjacent function.

Definition 5.1 Let function Adj(m,n)=1, if and only if the *mth* cluster in class *i* is adjacent to the *nth* cluster in the class other than class *i*, where *i*=1,2.

The between-class scatter matrix is calculated as follows:

$\forall_{1\le m\le r_1}\forall_{1\le n\le r_2}((Adj(m,n)=1)\rightarrow$

$S_{b1}^{m,n}=E([X_m^{knn}(C_{2n})-E(X_n^{knn}(C_{1m}))][X_m^{knn}(C_{2n})-E(X_n^{knn}(C_{1m}))]^T)$,

and $\forall_{1\le n\le r_2}\forall_{1\le m\le r_1}((Adj(n,m)=1)\rightarrow$

$S_{b2}^{n,m}=E([X_n^{knn}(C_{1m})-E(X_m^{knn}(C_{2n}))][X_n^{knn}(C_{1m})-E(X_m^{knn}(C_{2n}))]^T)$,

$$S_b=\sum_{m=1}^{r_1}\sum_{n=1}^{r_2}(S_{b1}^{m,n}+S_{b2}^{n,m}) \quad \ldots\ldots (5.4)$$

where $E(X_n^{knn}(C_{1m}))$ denotes the mean of the *k-NN* in cluster *n* of class two, corresponding to the mean of the cluster *m* in class one; and $E(X_m^{knn}(C_{2n}))$ represents the mean of the *k-NN* in cluster *m* of class one, corresponding to the mean of the cluster *n* in class two.

The determination of the *mth* cluster in one class being adjacent to the *nth* cluster in another class is based on Euclidean distance between the means of two clusters. If the distance between two clusters is less than or equal to a given threshold *T*, then the two clusters are said to be adjacent to each other. In our experiment, we assume each cluster is with Gaussian distribution and the threshold *T* is set equal to

$$T=c\,|M_1-M_2| \quad \ldots\ldots (5.5)$$

where M_1 is the mean of class one; M_2 is the mean of class two; and *c* is a weight coefficient. We have conducted experiments with different ranges of *c* on the verification of different handwritten pairs to estimate the optimal value for this parameter. In many of our experiments, we set the weight coefficient c to $0.85 < c < 1.20$.

The algorithm complexity for S_b is analyzed as follows: we assume that there are r_1 clusters in class one and r_2 clusters in class two. For each cluster in class one, we need to calculate the *k-NN* of each cluster in another class by using the quick sort algorithm. The algorithm complexity of *k-NN* computation is approximately $O(\frac{N}{r_2}\log\frac{N}{r_2})$ for each cluster in class one, and $O(\frac{N}{r_1}\log\frac{N}{r_1})$ for each cluster in class two. The computational complexity of Equation (5.4) is approximately equal to $O(n^2 r_1^{\,2} r_2^{\,2})$, where n is the dimension of a feature vector. The overall computation complexity of S_b is $O(r_1\frac{N}{r_2}\log(\frac{N}{r_2}) + r_2\frac{N}{r_1}\log(\frac{N}{r_1}) + n^2 r_1^{\,2} r_2^{\,2})$. All of n, r_1, r_2 are constants, where $r_1 << N$, $r_2 << N$, $n << N$, and $r_1 \approx r_2 < 10$, n=100 in the experiments, so the computation complexity of S_b is $O(N\log N)$, which is much smaller than that of NDA, which is $O(N^2\log N)$.

We now present our algorithm as follows:

1) Use the k-means algorithm to cluster the training data of each class into r_i clusters, denoted by $\{C_{ij} \mid i=1,2;\ j=1,2,\ldots, r_i\}$, and count the number of each cluster as N_{ij}.

2) Calculate the within-class scatter matrix S_w in Equation (5.3).

3) Linearly transform the original features X by the following equation (5.6), so that S_w is the identity matrix in the transformed space.

This procedure can be implemented using an *NxN* matrix

$$T = \Lambda^{(-1/2)}\Phi^T,$$

$$Y = T \bullet X. \qquad \ldots\ldots (5.6)$$

where Λ is an *NxN* diagonal matrix whose diagonal elements are the eigenvalues of S_w, and Φ is an *NxN* matrix, where the *ith* row is the *ith* eigenvector of S_w, corresponding to the largest *ith* eigenvalue of S_w.

4) Compute the between-class scatter matrix S_b in (5.4) based on the transformed feature matrix *Y*.

5) Find the *M* largest eigenvalues and the corresponding *M* eigenvectors and form the selected feature vectors:

$$\Psi = [\psi_1, \psi_2, ..., \psi_M]$$

Thus, the transformation from the original *N*-dimensional data into the *M*-dimensional features can be formed by $\Psi^T \Lambda^{(-1/2)} \Phi^T$.

The extracted feature number *M* can be determined based on the following rule: we may add eigenvalues of S_b in decreasing order until the accumulation exceeds a certain percentage of the total sum, then we set that number of the eigenvalues as a feature number to retain. Alternatively, we may retain the eigenvalues that are greater than a given proportion of the largest eigenvalue and set the retained feature number.

Our proposed multi-modal discriminant analysis has the following properties:

(1) The within-class scatter matrix S_w is more centralized because it is computed from multi-modal clusters;

(2) The *k-NN* data of all clusters in two classes are located on and near the effective decision boundary and these data can represent more truthfully the whole complex distribution along the effective decision boundary;

(3) The computation of the between-class scatter matrix S_b is based on the *k-NN* training samples on and nearthe effective decision boundary rather than on the whole training set, which therefore has much less computation than NDA;

(4) The optimal discriminant criterion $J = tr(S_w^{-1} S_b)$ is used to generate the transformation matrix, which has the functionality of minimizing the within-class distance and maximizing the between-class distance. As a result, our proposed method for calculating S_b and S_w leads to a criterion being minimized for within-class separability and being maximized for between-class separability. Hence, the discriminanting ability is enhanced for pattern recognition.

In conclusions, multi-modal analysis can be used in the complex distribution situations. No prior assumptions need to be made about class and cluster densities. However, for a two-class classification problem, if the training samples in two classes overlap heavily, the computation of the between-class scatter matrix will be affected. In such a case, more clusters are needed to reduce the degree of overlap.

5.3 Handwritten Numeral Verification

In order to increase the recognition rate of GPR, we need a model to verify the recognized digits. In this chapter, we will only focus on pair-wise and cluster verifications. For improving the efficiency and stability of the classifier for verification, we use our proposed method to conduct feature dimensionality reduction experiments.

The verification of confusing handwritten numeral pairs is a challenging task because the confusing character pairs are quite similar in terms of the features used in GPR (General Purpose Recognizer) or in terms of their shapes. It is necessary to develop a new

verification engine to explore detailed hybrid features for distinguishing between these similar and easily confusing character pairs. Theoretically, for 10 numerals, there are 45 confusing digit pairs (10x9/2=45). For cluster verification, the clusters with three characters are also discussed in the next sections.

There are four types of verifiers according to the number of classes. Let Ω denote the working space of a verifier, and let | Ω | denote the dimension of the space. The four verifiers are:

| Ω | =n: General verifier, working on all classes in the problem.

0<| Ω |<n: Cluster verifier, verification of clustered categories e.g. (Is it a "4", "6", or "9"?).

| Ω |=2: Pair-wise verifier, verification between two categories e.g. (Is it a "4" or "9"?).

| Ω |=1: Class-specific verifier, working on one candidate class e.g. (Is it a "1"?).

Generally speaking, hybrid features are extracted by various means in such a way that they are more likely to be complementary to each other, which is helpful for the verification of similar characters. We use the directional-based wavelet features and geometrical features in the experiments. The extraction of the two feature sets were presented in Chapter Four. Here, the dimensionality of the two feature sets is listed below:

Dimension of directional-based wavelet features = 80 (Feature Set I in Section 4.1.1)

Dimension of geometric features =20 (Feature Set VII in Section 4.1.7)

So the total number of original features used in our experiments becomes 80+20=100.

For the classifier, we choose a three-layer ANN with Back Propagation (BP) training algorithm, using the following parameters: the number of nodes in the input layer is the

same as the number of features; the number of nodes in the hidden-layer is set at 20 and the number of nodes in the output layer is 2 to represent two classes.

Our experiments focus on a pair-wise verifier, which is a one-to-one verification between two categories. From our observations, some of the most confusing numeral pairs [85] are {4,6}, {0,8}, {2,3}, {2,1}, {4,0}, {7,3}, {9,7}, {4,9}, {5,3}, {0,6}, {8,3}, {7,1}, {9,5}, {7,2}, {9,8}, {8,2}, {6,5}, {8,5}, {9,0}, {8,4}, etc. depending on the output of GPR and the features used in the GPR. In order to test our proposed method for feature dimensionality reduction, as an example, the verification results of experiments conducted on handwritten numeral pairs "4" and "6" are analyzed and the verification results of other pairs are summarized.

- **Database**

We extracted three sets of data from the MNIST database. For each character in the pair, the first 3,000 samples are used as training samples; another set of 1,000 samples for verification while training, and the last 1,000 samples for testing.

For example, for the verification of handwritten numeral pair "4" and "6", we construct the training, verification and testing datasets as follows:

Character	Training dataset	Verification dataset	Testing dataset
4	3,000	1,000	1,000
6	3,000	1,000	1,000

- **Verification Experiments Conducted on Characters "4" and "6"**

As an example of the verification on the pair-wise numerals "4" and "6", 100 original features of each of the 3000 training samples and each of the 1000 verifying and the 1000

testing samples of character pair “4” and “6” were used as original data. Each class was divided into six clusters by k-means. The number of clusters in each class was determined empirically by the writing style of testing samples of the numeral pairs. To evaluate our proposed algorithm for dimensionality reduction, we conducted a series of experiments on different numbers of features extracted by our proposed method. For example, Figs 5.3-5.6 show the recognition rates obtained from the training samples, the ANN training errors, and the verification rates on characters “4” and “6” with the feature dimensionality varying from 100, 50, 10 until to 1.

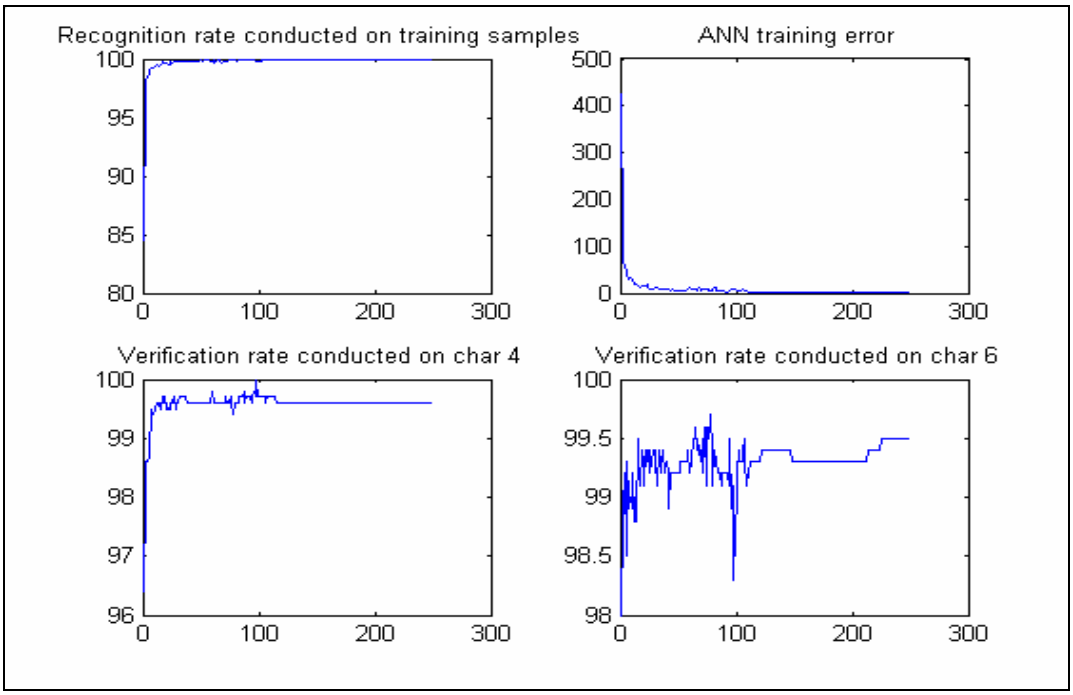


Fig. 5.3 ANN training and testing on 100 original features

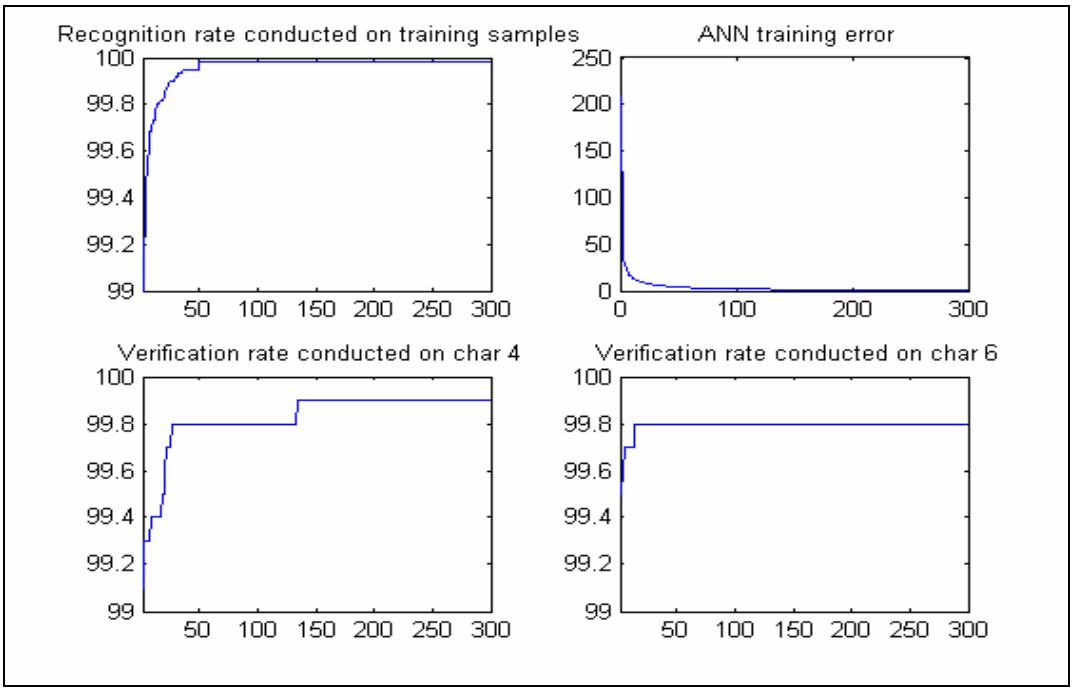


Fig. 5.4 ANN training and testing on 50 features extracted by our proposed method

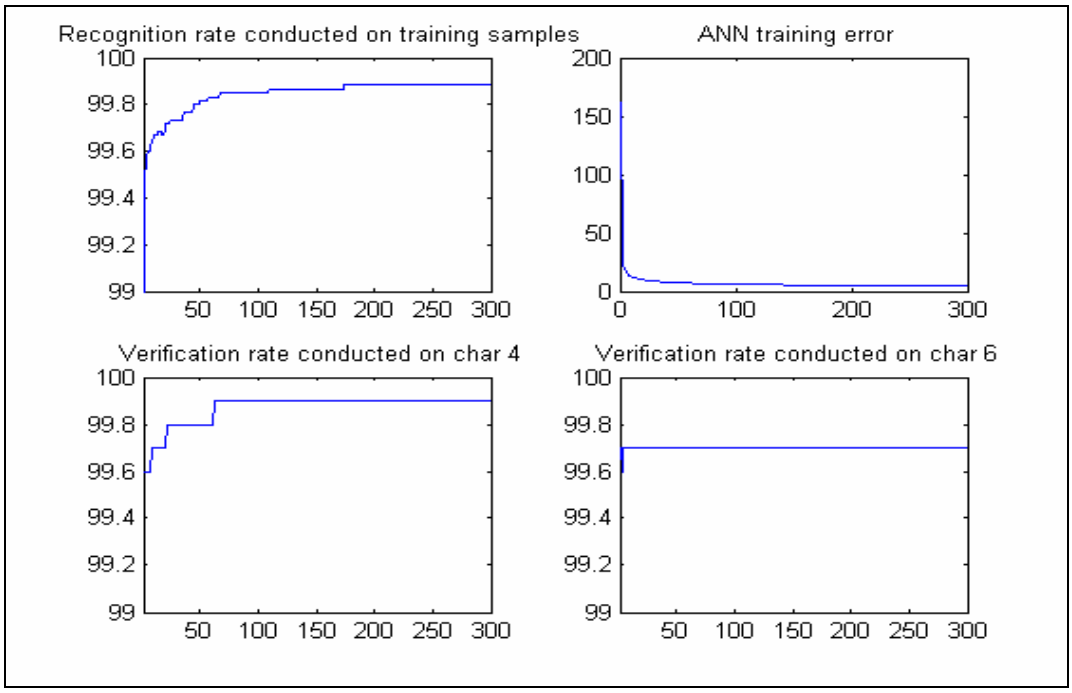


Fig. 5.5 ANN training and testing on 10 features extracted by our proposed method

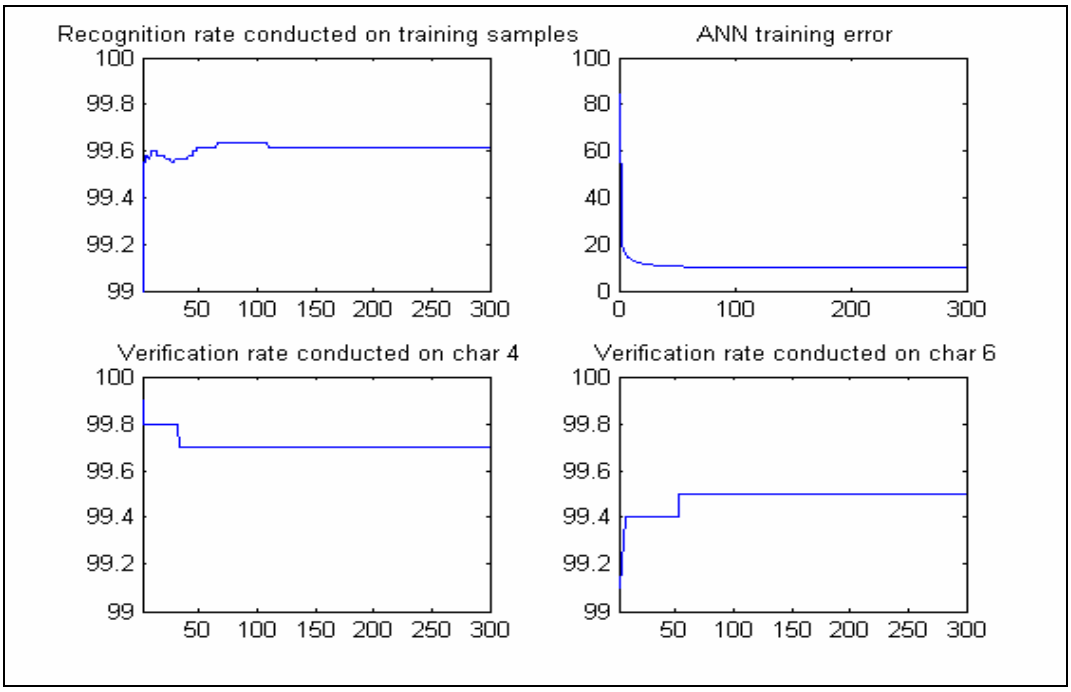


Fig. 5.6 ANN training and testing on 1 feature extracted by our proposed method

From Figs. 5.3-5.6, it can be observed that the recognition rates on the training samples and the verification rates on the verification samples do not deteriorate much, or not at all, when the feature dimensionality is reduced by our proposed algorithm. It means that our proposed method can be applied to large-scale feature compression without losing much discriminating information.

- **Verification Experiments Conducted on Other Character Pairs**

We conducted feature dimensionality reduction experiments on 20 of the most confusing pairs listed in reference [85]. All the verification experiments were conducted on 2000 testing samples (the ANN classifier was trained by 6000 training samples). In order to save space, we only list the verification results of the first 10 character pairs with experiments, conducted on different numbers of features, as shown in Table 5.1.

Table 5.1 Verification rates at different dimensionalities

CP / FD	4 & 6	0 & 8	2 & 3	2 & 1	4 & 0	7 & 3	9 & 7	4 & 9	5 & 3	0 & 6
100	99.8	99.7	99.3	99.6	99.5	99.5	99.6	99.2	99.4	99.5
70	99.8	99.7	99.3	99.6	99.5	99.5	99.6	99.2	99.4	99.5
50	99.8	99.7	99.3	99.6	99.5	99.5	99.6	99.2	99.4	99.5
20	99.8	99.6	99.3	99.6	99.4	99.5	99.6	99.2	99.4	99.5
10	99.8	99.6	99.2	99.6	99.4	99.5	99.5	99.1	99.3	99.5
5	99.6	99.5	99.2	99.5	99.4	99.5	99.5	99.1	99.2	99.4
1	99.5	99.4	99.1	99.4	99.3	99.4	99.4	99.1	99.1	99.3

(CP: character pairs, FD: feature dimensionality)

The features retained in each dimensionality from 1 to 100 are chosen by our proposed multi-modal discriminant method described in Section 5.2.

Table 5.2 presents a comparison of ANN training times of different numbers of features conducted on 6000 training samples using our proposed method.

Table 5.2 ANN training times for different numbers of features

No. of Features	Training Time (s)
100	3900
70	2700
50	2010
20	850
10	500
5	310
1	177

The experimental results show that our proposed method can keep a very high feature compression rate without losing much information on classification ability. As a result, this new method can reduce ANN's training complexity and can accelerate the ANN training procedure with a higher reliability.

- **Verification Model to Increase the General Recognizer's Performance**

In chapter 4, when Feature Set B was used as input to the general recognizer, the recognition rate was 98.58%. In the 10,000 testing samples of the MNIST database, 40 testing samples were not voted on the top two ranks, which could not be corrected by the verification model. The other 102 errors occurred when the general recognizer votes for the recognized character as the second candidate. Some of the errors in the category could be corrected.

A verification experiment was conducted on the 9960 testing samples (9,858 correctly recognized digits in the general recognizer + 102 misrecognized digits with second largest confidence values in the general recognizer). The verification modal used 20 compressed features of Feature Set A.

The overall recognition rate (General recognizer + Verifier) has been increased from 98.58% to 99.10%.

5.4 Handwritten Numeral Recognition using a Verification Model

Fig. 5.7 shows a character recognition system. We designed ten absolute verifiers for classification (e. g. one absolute verifier for distinguishing one numeral from the other

nine numerals). The output of the General Purpose Recognizer (GPR) is the congregation of the ten verifiers.

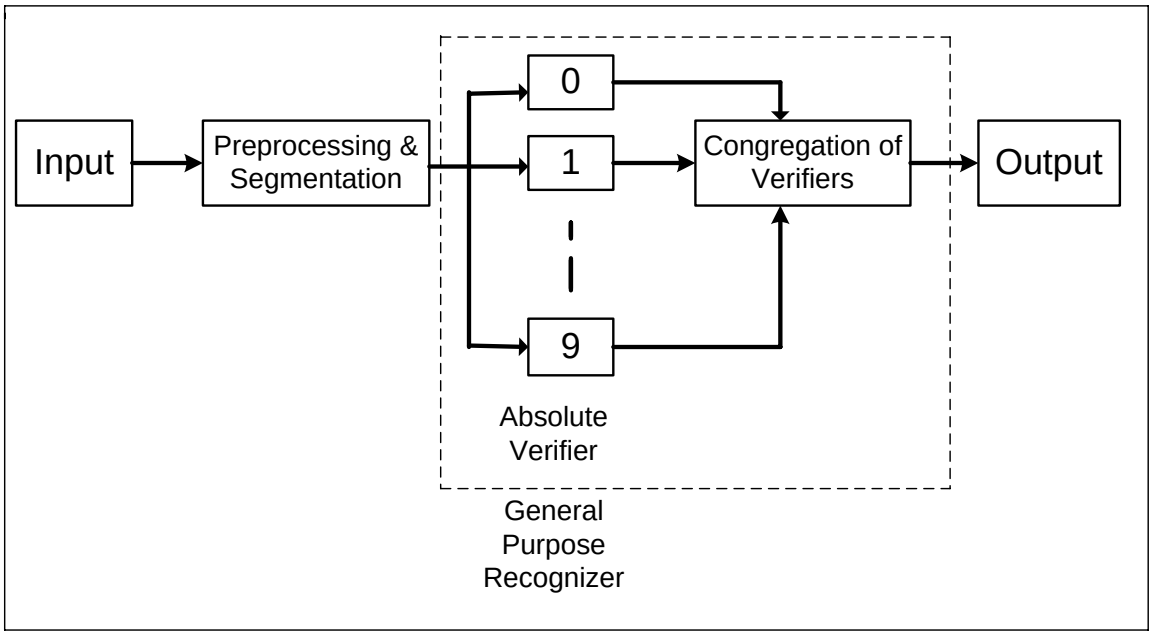


Fig. 5.7 A character recognition system using ten absolute verifiers

Our experiments focused on the absolute verification between two categories (e.g., classifying one numeral from the mixed class with the other nine categories). For 10 numerals, we needed to build ten classifiers. Therefore, the congregation of the ten classifiers, led to a higher recognition result.

The training and testing procedures and ANN classifier configurations are the same as those used in Section 5.3.

- **Verification Experiment Distinguishing Numeral 4 from Other Numerals**

Class of number 4 is divided into six clusters and the class of the mixed numbers (including characters 0, 1, 2, 3, 5, 6, 7, 8, 9) is divided into seventeen clusters,

empirically based on the similarity of features extracted from each class. Fig. 5.8 shows the recognition rate obtained from the training samples, the ANN training error, and the verification rates, conducted on two categories (character "4" and mixed characters), using 100 features with ANN iterations. As the training data in one class "4" and another class with the mixed numbers are highly imbalanced, we used the prior duplication method to keep the training data balance, which was introduced in Section 3.3.

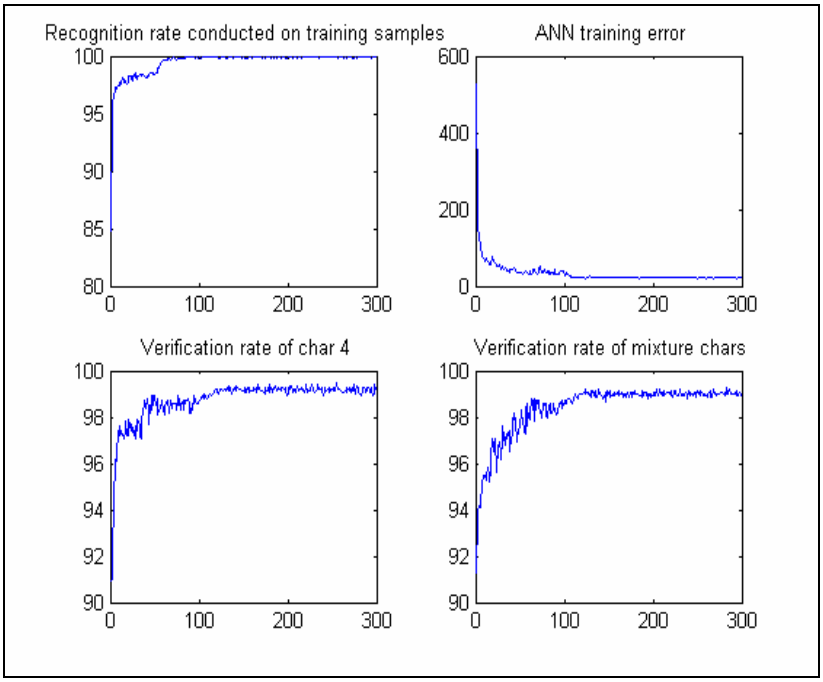


Fig. 5.8 ANN training and testing results conducted on 100 features of "4" and the mixed characters

- **Congregation of Verification Results for Handwritten Numeral Recognition**

We conducted nine other absolute verification experiments in the same way as described before. Table 5.3 lists the recognition rates conducted on ten handwritten numerals with

different features extracted by our proposed method. These results compare favorably with those obtained by other researchers in the field.

Table 5.3 Recognition rates conducted on ten numerals

NF / CH	100	70	50	20	10
Char. 0	99.1	99.0	99.0	98.6	98.4
Char. 1	99.2	99.1	99.0	98.7	98.3
Char. 2	99.0	98.7	98.6	98.4	97.8
Char. 3	98.9	98.7	98.5	98.3	98.0
Char. 4	99.0	98.3	98.2	98.1	97.3
Char. 5	99.0	98.4	98.2	98.1	98.0
Char. 6	99.1	98.9	98.6	98.4	98.4
Char. 7	99.3	99.1	98.8	98.5	98.0
Char. 8	99.1	99.0	98.7	98.4	98.1
Char. 9	98.7	98.4	98.1	98.0	97.5

Note: NF: No. of Features; CH: Characters

Experiments demonstrated that if fewer features are used during the ANN training, then fewer training iterations are needed for ANN convergence. In other words, the feature dimensionality reduction could greatly speed up the ANN training procedure and could make the ANN training converge more easily.

5.5 Verification Experiments Conducted on Clusters

It is possible that the GPR can output three or more candidates with similar confident values. In order to investigate how our proposed complex wavelet features (dimension of feature vector =160) and geometrical features (dimension of feature vector =20) can be

used for cluster verification, we conducted verification experiments on some confusing clusters with three characters in each cluster, such as {2, 3, 5}, {1, 7, 9}, {4, 6, 9,},{0, 6, 9}, etc. As an example, the recognition rate conducted on the training set, the verification rates conducted on the verifying set and the testing set of cluster {2, 3, 5} are shown in Table 5.4.

Table 5.4 The training, verifying, and testing results conducted on cluster {2, 3, 5}

Character pairs	2 - {3,5}	3 - {2,5}	5 - {2,3}
Recognition rate of training set (%)	99. 65	99.55	99.60
Verification rate of verifying set (%)	99.10	99.10	99.15
Verification rate of testing set (%)	99. 05	99.10	99.10

Similar verification experiments were conducted on three other clusters :{1, 7, 9}, {4, 6, 9} and {0, 6, 9}. Table 5.5 lists the overall verification results on the testing sets for the three clusters.

Table 5.5 Verification rates conducted on testing sets of three clusters

Clusters	Overall verification rate (%)
{1,7,9}	99.20
{4,6,9}	99.10
{0,6,9}	99.15

It can be concluded that the verification model can be used on clusters and good verification performances have been achieved.

5.6 Comparison with Other Similar Methods

Fukunaga et al. [31] developed a mono-modal nonparametric discriminant analysis method based on optimal criterion $J = tr(S_w^{-1} S_b)$ for classification problems. Hastie et al. [39-41] proposed a mixture discriminant analysis method to cluster each class into subclasses and to model each class by a mixture of two or more Gaussians with different centroids, then both of the flexible discriminant analysis and the penalized discriminant analysis adapt naturally to MDA, which is the prototype of our multi-modal method. Furthermore, the authors used a local linear discriminant analysis to estimate an effective metric for iteratively computing neighborhoods, and then to shrink neighborhoods in directions orthogonal to these local decision boundaries, and to elongate them parallel to the boundaries. Therefore, their global dimension reduction combines local dimension information. Bressan et al. [5] modified the computation of S_w of NDA by calculating the *k-NN* of each training sample in the same class and computing the within-class scatter matrix based on the difference between the training sample and its *k-NN* mean.

Our approach is based on Fukunaga et al.'s mono-modal nonparametric analysis. The concept of multi-modal discriminant analysis is inspired by Hastie et al.'s mixture discriminant analysis. However, our approach is different from their methods. The number of clusters in each class can be obtained by the unsupervised clustering method. Then S_w is calculated from the clusters, making it more centralized. We only consider the *k-NN* of each cluster, corresponding to each cluster coming from a different class, which guarantees that those *k-NN* training samples are located on the effective decision boundary when computing S_b. The optimal Fisher criterion based on our proposed S_b and S_w is used to generate a transformation matrix for maximizing the between class distance

and at the same time, minimizing the within class distance to increase the discriminant ability of the classifier. In addition, the overall computational complexity of the transformation matrix is less than those of other similar approaches. No prior assumptions about class and cluster densities are needed.

Fukunaga and Mantock's NDA [31] was a pioneer work on nonparametric discriminant analysis for dimensionality reduction. Bressan and Vitria's MNDA [5] is an example of a recent development for improving the recognition performance, which is similar to our approach. In order to compare our proposed method with PCA, NDA as well as MNDA, Fig. 5.9 shows four verification results derived from the average testing results of 20 of the most confusing numeral pairs.

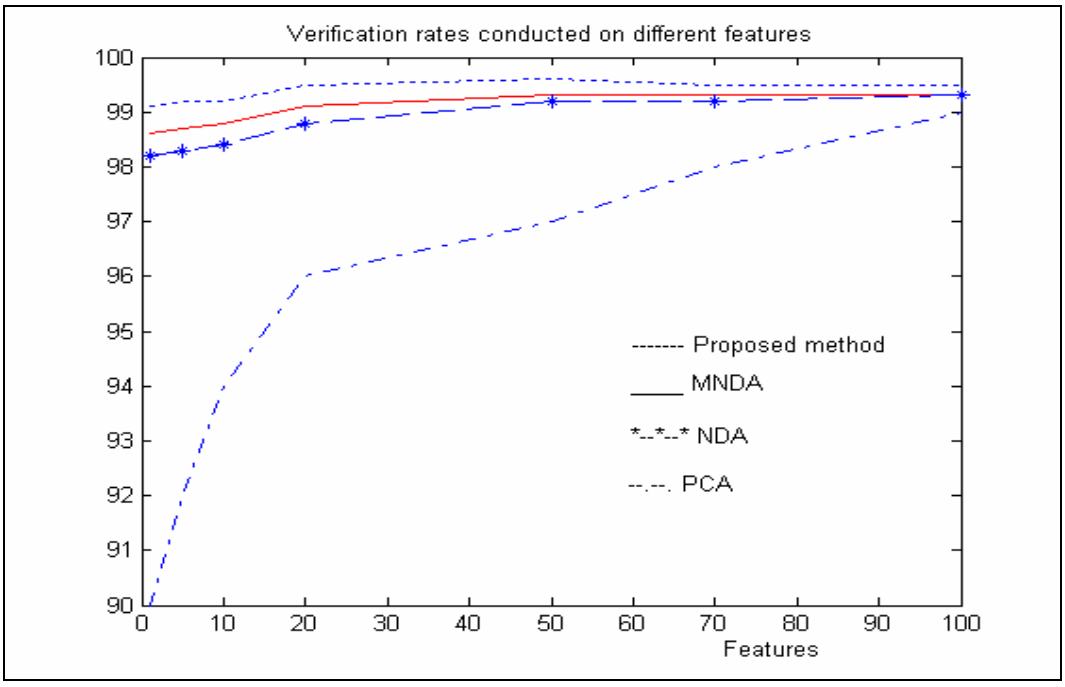


Fig. 5.9 Comparison of the verification performance of four methods

Among the four methods, our proposed method shows a better verification rate than NDA and MNDA on different features. The PCA algorithm had the worst performance.

The reason why our proposed method leads to a better classification performance than other similar approaches is that our method uses more relevant information on and near the decision boundary.

As we have analyzed in the previous section, to get the optimal criterion $\boldsymbol{J}$, the computational complexity of the S_b for both NDA and MNDA is approximately $O(N^2 \log N)$; whereas in our proposed method, it is $O(N \log N)$. Table 5.6 is the CPU time comparison of three similar methods to generate S_b (Experiments were conducted on an IBM PC computer with CPU speed of 2.0 GHz).

Table 5.6 Comparison of CPU time needed to generate the between-class scatter matrix

	Training Samples	CPU Time (second)
MNDA	6000	1440
NDA	6000	1440
Our proposed method	6000	12

Chapter Six

Analysis of Error, Rejection, and Recognition Rates in a Cascade Ensemble Classifier System

In this section, we will analyze the tradeoffs in error, rejection, and recognition rates in a cascade ensemble classifier system which consists of several levels of ensemble classifiers. An ensemble classifier is the combination of individual classifiers. The tradeoff analysis is conducted on an ANN classifier, an ensemble classifier and a cascade ensemble classifier, respectively. The solutions for improving recognition performance will be given.

6.1 Analysis of Error, Rejection, and Recognition Rates in an ANN Classifier

In our proposed classification system, ANNs are the dominant classifiers. We will analyze the relationships among the error, rejection and recognition rates of an ANN classifier using Bayesian probability theory. According to the rule of thumb [97]: 1) for a multi-class problem, a multilayer perceptron neural network trained with back-propagation has good estimates of Bayesian probabilities; 2) interpretation of network outputs as Bayesian probabilities makes it possible to compensate for differences in pattern class probabilities between test and training data, the error analysis of an ANN classifier is based on Bayesian estimation.

In order to pursue the highest reliability and the lowest error rate with rejection strategy, a recognition rule is optimum if for a given recognition rate, it minimizes the error rate (error probability) and puts the uncertain testing candidates into the rejection category. According to reference [17], suppose there is the n-class problem and X is a feature vector, if the decision rule has a rejection strategy, we need to build up an additional class (for example, the 0^{th} class) to represent the rejection category, so that

If ($Vote(d_k \mid X) = 1$) and ($1 \le k \le n$) then X is classified;

If ($Vote(d_k \mid X) = 1$) and ($k == 0$) then X is rejected.

The optimum rule is to reject the pattern if the maximum of the a posteriori probabilities is less than the defined threshold. According to Bayesian probability theory, the optimum rule has the following two conditions:

1) To accept the pattern X for recognition and to identify it as belonging to the k-th pattern:

$$Vote(d_k \mid X) = 1$$

if and only if

$$P(\omega_k)F(X \mid \omega_k) \ge P(\omega_i)F(X \mid \omega_i)$$

and

$$P(\omega_k)F(X \mid \omega_k) \ge (1-t)\sum_{i=1}^{n} P(\omega_i)F(X \mid \omega_i) \qquad \ldots\ldots (6.1)$$

2) To reject the pattern X:

$$Vote(d_0 \mid X) = 1$$

whenever

$$\max_k [P(\omega_k)F(X \mid \omega_k) < (1-t)\sum_{i=1}^{n} P(\omega_i)F(X \mid \omega_i) \quad \ldots\ldots (6.2)$$

where n is the number of classes, $P(\omega_i)$ ($i=1,2,3,...n$) is a priori probability of observing class ω_i, $P(X/\omega_i)$ is the conditional probability density for X given the ith class, and t is a constant parameter between 0 and 1 $(0 \le t \le 1)$. The relationships among the error, rejection, and recognition rates are listed below:

The probability of error, or error rate, is:

$$E(t) = \int_v \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} Vote(d_j \mid X)P(\omega_i)F(X \mid \omega_i)dX \quad \ldots\ldots (6.3)$$

The probability of rejection or reject rate is:

$$R(t) = \int_v Vote(d_0 \mid X)\sum_{i=1}^{n} P(\omega_i)F(X \mid \omega_i)dX \quad \ldots\ldots (6.4)$$

The probability of a correct recognition rate is:

$$C(t) = \int_v \sum_{i=1}^{n} Vote(d_i \mid X)P(\omega_i)F(X \mid \omega_i)dX = 1 - E(t) - R(t) \quad \ldots\ldots (6.5)$$

From the above analysis, we know that the error, rejection, and correct recognition rates are implicit functions of the threshold parameter t.

The probability of the acceptance or acceptance rate is defined as:

$$A(t) = C(t) + E(t) \quad \ldots\ldots (6.6)$$

The reliability of the recognition system is denoted as:

$$\mathrm{Re}(t) = C(t) + R(t) \quad \ldots\ldots (6.7)$$

In a neural network classifier, the confidence threshold (*Conf*) can be related to the parameter t. The relation is denoted as:

$$ANN_{conf} = w * (1 - t) \quad \ldots\ldots (6.8)$$

where w is an empirical factor, which is selected based on how high the reliability is set. Normally, in our experiments, w is set between 1.0-2.0. Equation (6.8) demonstrates that an ANN classifier can introduce a rejection strategy by setting a high threshold of confidence value. Based on the above analyses, there are several ways to reduce both the rejection rate and the error rate:

- In order to reduce the error rate, we need to expand the "rejection region" by setting a smaller parameter t in equations (6.1) and (6.2). As a result, more patterns are rejected and fewer patterns are either falsely or correctly accepted. For example, in a neural network classifier, the confidence threshold ANN_{conf} can be set at a high value.
- Based on equation (6.2), in order to reduce the rejection rate, we can increase the value of $\max_k[P(\omega_k)F(X \mid \omega_k)$ and at the same time, we can reduce the value of $\sum_{i=1}^{n} p(\omega_i)F(X|\omega_i)$. In practical applications, when a feature vector (X), which is extracted from a labeled class *i*, is input into an ANN classifier, the ANN classifier should have the highest conditional probability density ($F(X \mid \omega_i)$ for the labeled class *i* and the lowest probability density in all other classes. This means that the more discriminative features play an important role in reducing the rejection rate.

For simplicity, as an example of two classification problems, each feature vector X is an m-dimensional vector. We assume that the two classes have the same a priori probability,

i.e., $P(\omega_1) = P(\omega_2) = \frac{1}{2}$, and the feature vectors are in the normal distribution with means u_1 and u_2, as well as the equal covariance σ^2. The following expressions exist:

$$\frac{w - ANN_{conf}}{ANN_{conf}} \le \frac{P(\omega_1)F(X \mid \omega_1)}{P(\omega_2)F(X \mid \omega_2)} \le \frac{ANN_{conf}}{w - ANN_{conf}} \quad \ldots\ldots \quad (6.9)$$

$$F(X \mid \omega_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{(X - U_i)^2}{2\sigma^2}] \quad \ldots\ldots \quad (6.10)$$

where i=1,2; w is a factor used in equation (6.8).

After some manipulations, the corresponding error and rejection rates can be formulated as follows:

$$E(ANN_{conf}) = \varphi(a)$$

$$R(ANN_{conf}) = \varphi(b) - \varphi(a)$$

$$\varphi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-(x^2/2)dx}$$

and

$$a = -\frac{1}{2}s - \frac{1}{s}\ln(\frac{w}{w - ANN_{conf}} - 1)$$

$$b = -\frac{1}{2}s + \frac{1}{s}\ln(\frac{w}{w - ANN_{conf}} - 1)$$

$$s = \frac{U_1 - U_2}{\sigma}$$

where s is the mean difference for two feature vectors

Fig. 6.1 shows the tradeoff curves of the error and rejection rates with their mean differences.

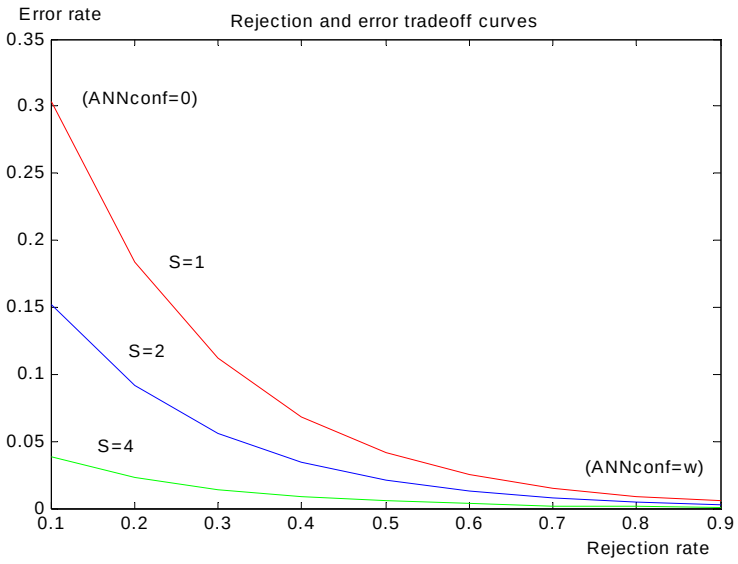


Fig. 6.1 Rejection and error tradeoff curves with mean differences

Based on the analysis, there is one way to reduce the rejection and the error rates: the larger mean difference (s) between two classes will result in a significant reduction in both the rejection rate and the error rate. It means that more discriminative feature vectors are helpful in suppressing both rejection rate and the error rate.

6.2 Analysis of an Ensemble Classifier

According to the principle of divide and conquer, a complex task can be solved by dividing it into a number of computationally simpler tasks. The simpler tasks can be achieved by distributing the tasks to a number of experts. For example, one way is to divide the input space into a set of subspaces. Each expert works on an individual subspace. The combination of experts is said to constitute an ensemble classifier. The responses of several experts are combined to produce an overall output.

Fig. 6.2 shows the block diagram of an ensemble classifier. For simplicity, it is assumed that input x_i, (i=1,2….n) is either an individual feature component or a feature vector, whose outputs are somehow combined to produce an overall output y.

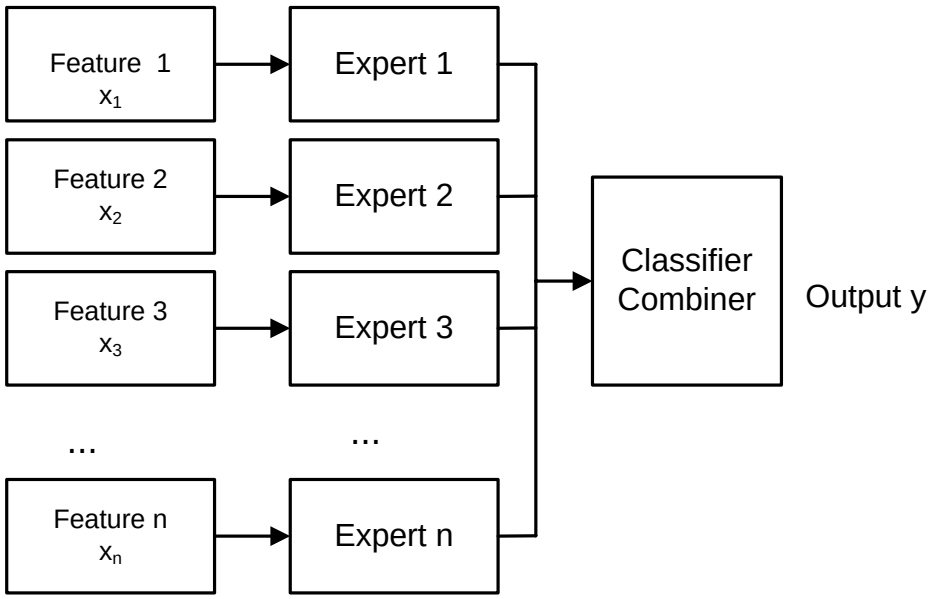


Fig. 6.2 Block diagram of an ensemble classifier

Generally speaking, in a classification problem, the goal of the classification is to predict the output value Y (where Y is a label vector $[y_1, y_{2,}...y_C]^T$ with C elements, which denotes C classes with one corresponding to the correct class, and all others corresponding to zero), given the values of a set of input features $X = \{x_1, x_2,...x_n\}$, simultaneously measured on the same system. We denote the approximation in the following equation:

$$Y = F(X) + \varepsilon \qquad \ldots\ldots (6.11)$$

Here, *F(X)* is an estimating function vector with *C* elements. For instance, for neural networks or gating networks, which we will discuss in the next chapter, all data in the vector should be equal to or larger than 0.0 and the sum of all data in the vector should be 1.0. ε is a random variable vector. The "training" data set $T = \{X, Y\}_1^N$ is used to "learn" a classification rule. Namely, the usual paradigm for accomplishing classification is to use the training data *T* to form an approximation of $F(X,T)$ for $F(X)$.

In this thesis, we use multilayer perceptron artificial neural networks and gating networks as classifiers, which minimize the mean squared error between the actual outputs (outputs of the networks) and the desired outputs (labels). Therefore, the recognition performance can be evaluated by the mean squared prediction errors, which come from the difference between the label *Y* and its estimation function $F(X,T)$, based on the training samples. Through the decompositions [28], the following equations exist:

$$E_\Omega[Y - F(X,T)]^2 = E_\Omega[F(X) - F(X,T)]^2 + E_\varepsilon[\varepsilon \mid X]^2 \quad \ldots\ldots (6.12)$$

$$E_\Omega[F(X) - F(X,T)]^2 = [F(X) - E_\Omega F(X,T)]^2 + E_\Omega[F(X,T) - E_\Omega F(X,T)]^2 \quad \ldots\ldots (6.13)$$

The random variable vector ε is independent of the training data *T* and the estimation function $F(X,T)$ with $E(\varepsilon \mid X) = 0$ [28]. In the following discussion, we will only analyze the estimation errors caused by $F(X,T)$. The expectation E_Ω takes over the training space Ω.

The quantity in equation (6.13) is equal to the bias and the variance:

Bias: $$B_\Omega(F(X)) = [F(X) - E_\Omega F(X,T)]^2 \quad \ldots\ldots (6.14)$$

Variance: $$V_\Omega(F(X)) = E_\Omega[(F(X,T) - E_\Omega F(X,T))^2] \quad \ldots\ldots (6.15)$$

Generally, the variance reflects the sensitivity of the estimation function $F(X,T)$ to the training samples *T*. The bias represents how closely on average the estimation function $F(X,T)$ is about to approximate the target function $F(X)$. The bias and the variance are two important factors affecting the estimation error or classification error in a pattern recognition problem. Equation (6.13) can be rewritten as:

$$E_{\Omega}[F(X)-F(X,T)]^2 = B_{\Omega}(F(X)) + V_{\Omega}(F(X))$$

We will discuss an ensemble logical "and" scheme and an ensemble averaging scheme in the next two subsections.

6.2.1 Ensemble Logical "and" Scheme

The overall recognition outputs are based on the logical "and" operation on different classifiers $F_i(X_i,T)$.

$$\overline{F}(X,T) = \bigwedge_{i=1}^{M} F_i(X_i,T) \qquad \ldots\ldots (6.16)$$

An ensemble logical "and" scheme can be represented as:

$$(\overline{F}(X,T) \Rightarrow d_j \mid_{C_{j,\overline{F}(X,T)} \geq threshold}) \leftrightarrow \forall_{i,i=1,\ldots,M}[F_i(X_i,T) \Rightarrow d_j \mid_{C_{j,F_i(X_i,T)} \geq threshold}]$$

$\ldots\ldots$ (6.17)

In equation (6.17), an ensemble logical "and" classifier votes for a recognizing digit, such as d_j, with the condition that its confidence value of *j*th output node is equal to or larger than a predefined threshold ($C_{j,\overline{F}(X,T)} \geq threshold$) ***if and only if*** for every classifier $F_i(X_i,T)$, (*i=1,2,...,M*) in the ensemble classifier, each $F_i(X_i,T)$ also votes for the same recognizing digit, such as d_j, with the confidence value of the *j*th output node

being equal to or larger than the predefined threshold ($C_{j,\overline{F}_i(X_i,T)} \geq threshold$); otherwise, the recognized numeral will be rejected by the ensemble logical "and" classifier.

In order to keep the overall recognition performance, when an ensemble logical "and" scheme is employed, it is necessary for each classifier $F_i(X_i,T)$ to have a similar recognition performance. The ensemble logical "and" classifier does not increase the recognition rate. Instead, it will reject those patterns with low confidence values in any individual classifier. As a result, the ensemble classifier scheme will enhance recognition reliability.

6.2.2 Ensemble Averaging Scheme

An ensemble averaging scheme with different feature sets of X_i on a training set T can be used in the sum voting scheme. The ensemble averaging scheme can be denoted as:

$$\overline{F}(X,T) = \tfrac{1}{M}\sum_{i=1}^{M} F_i(X_i,T) \qquad \ldots\ldots (6.18)$$

If we omit Ω and (X_i,T) in the second term on the RHS of equation (6.13), then, the variance becomes:

$$\begin{aligned} V_\Omega(\overline{F}) &= E_\Omega[(\overline{F} - E(\overline{F}))^2] = E[(\frac{1}{M}\sum_{i=1}^{M} F_i - E[\frac{1}{M}\sum_{i=1}^{M} F_i])^2] \\ &= E[(\frac{1}{M}\sum_{i=1}^{M} F_i)^2] - 2E(\frac{1}{M}\sum_{i=1}^{M} F_i) \bullet E(\frac{1}{M}\sum_{i=1}^{M} F_i) + (E[\frac{1}{M}\sum_{i=1}^{M} F_i])^2 \\ &= E[(\frac{1}{M}\sum_{i=1}^{M} F_i)^2] - (E[\frac{1}{M}\sum_{i=1}^{M} F_i])^2 \end{aligned} \qquad \ldots\ldots (6.19)$$

Thus, equation (6.19) can be rewritten as [78]:

$$\begin{aligned} V_\Omega(\overline{F}) &= E_\Omega[(\overline{F} - E(\overline{F}))^2] = E[(\frac{1}{M}\sum_{i=1}^{M} F_i)^2] - (E[\frac{1}{M}\sum_{i=1}^{M} F_i])^2 \\ &= \frac{1}{M^2}\sum_{i=1}^{M}\{E[F_i^2] - (E[F_i])^2\} + \frac{1}{M^2}\{\Delta(E[F_iF_j]) - \Delta(E[F_i]E[F_j])\} \end{aligned} \qquad \ldots\ldots (6.20)$$

It can be deduced that:

$$V_{\Omega}(\overline{F}) \le \frac{V_{\Omega}(F_i) + \max_{i,j}(\Delta E[F_i F_j] - \Delta E[F_i]E[F_j])}{M} \le \max_i V_{\Omega}(F_i) \quad \ldots\ldots \quad (6.21)$$

where, $V_{\Omega}(F_i) = E[(F_i)^2] - (E[F_i])^2$, M>1.

Equation (6.21) shows that the variance $V_{\Omega}(\overline{F})$ of the ensemble averaging scheme is less than that of any individual classifier.

The bias of the ensemble classifier is denoted as:

$$B_{\Omega}(\overline{F}(X)) = \{\frac{1}{M}\sum_{i=1}^{M} F_i(X_i) - E[\frac{1}{M}\sum_{i=1}^{M} F_i(X_i, T)]\}^2$$

As the output label of a character in an ensemble classifier is denoted as Y, according to equation (6.11), we assume that all the ideal estimating function $F_i(X_i)$ (without considering training samples T) for ith classifier in the ensemble classifier is approximately equal to Y, then we have :

$$F_1(X_1) \cong F_1(X_2) \ldots \cong F_M(X_M) \cong Y$$

where symbol "$\cong$" means being equal to or approximately being equal to.

If we design the feature sets X_i and the corresponding classifiers $F_i(X_i, T)$ trained by the training samples T, (i=1,2,...,M) with the same estimation (recognition) performance, then we readily see that the bias of an ensemble averaging classifier is approximately equal to that of any individual classifier in the ensemble scheme.

From the above analyses, the following solutions can be obtained:

1) When an ensemble logical "and" classifier is applied, the recognition reliability is enhanced.

2) A smaller variance in an ensemble averaging scheme will lead to a lower error rate (in comparison with any individual classifier).

In the next section, we will introduce a cascade recognition system for reducing the rejection rate and increasing the recognition rate.

6.3 Analysis of a Cascade Ensemble Classifier System

A cascade classifier system can be composed of several two-level classifier systems. In this section, we will only discuss a two-level cascade classifier scheme with two ensemble classifiers, which are shown in Fig. 6.3. The input to the second ensemble classifier consists of the rejected characters in the first ensemble classifier.

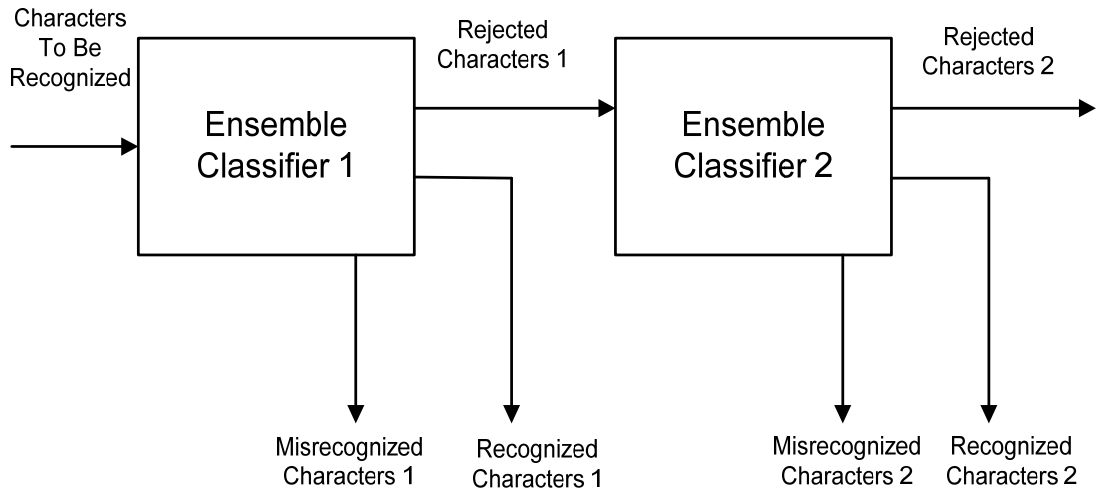


Fig. 6.3 A two-level cascade ensemble classifier scheme

In the two-level ensemble classifier, the following relations are satisfied:

I) Overall Recognized Characters=Recognized Characters 1 + Recognized Characters 2

II) Overall Misrecognized Characters=Misrecognized Characters 1 + Misrecognized Characters 2

III) Rejected Characters 2=Rejected Characters 1-Misrecognized Characters 2-Recognized Characters 2

Based on the above analysis, some conclusions can be drawn:

- The number of rejected characters after any two-level ensemble classifier is smaller than that of any one-level ensemble classifier. Namely, the rejection rate of any two-level ensemble classifier is lower than that of any one-level ensemble classifier.
- The correct recognition rate of the two-level ensemble classifiers is higher than that of any one-level ensemble classifier.
- The misrecognition rate of the two-level ensemble classifiers is the sum of the two one-level ensemble classifiers.

As discussed in section 6.1, the misrecognition (error) rate can be reduced by expanding the rejection space, or by setting a higher confidence threshold in the recognition system, or by using an ensemble logical "and" classifier. Correspondingly, the side effect of increasing the rejection rate can be suppressed by a multi-level cascade classifier system.

In conclusion, there are three ways to simultaneously reduce the error rate, the rejection rate, and at the same time, to increase the system's correct recognition rate:

1) extracting more discriminative features
2) using ensemble classifiers
3) employing a cascade classifier system

In our proposed ensemble classifier system, three ANNs and their gating networks are used to form an ensemble classifier based on the following: an ensemble classifier consisting of at least three classifiers can form a democratic voting system.

Chapter Seven

Cascade Ensemble Classifier System for the Recognition of Handwritten Numerals with Rejection Strategies

7.1 A Cascade Ensemble Classifier System

A novel cascade ensemble classifier scheme with rejection strategies is proposed in order to achieve the lowest error rate while pursuing the highest recognition rate for the recognition of handwritten numerals. The recognition scheme is shown in Fig.7.1.

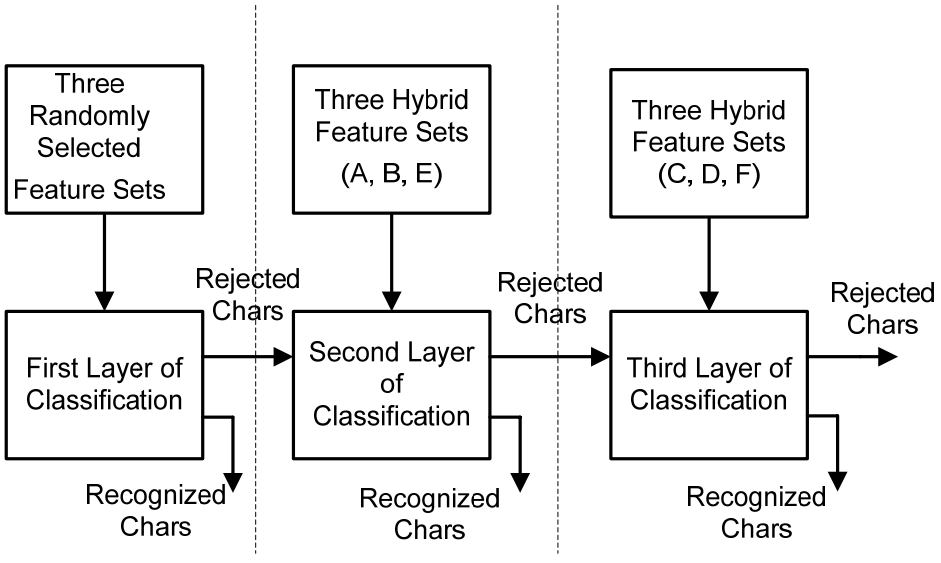


Fig. 7.1 A cascade recognition system with rejection strategy

Note: Chars: characters

The cascade system shown in Fig. 7.1 consists of three layers of classification, which are serially linked. Depending on the classification scheme which will be discussed in the next section, each layer is composed of four levels of Multi-ANNs with/without Gating Networks (MANNGN) ensemble classifiers. The schematic diagram of one layer of classification is shown in Fig. 7.2.

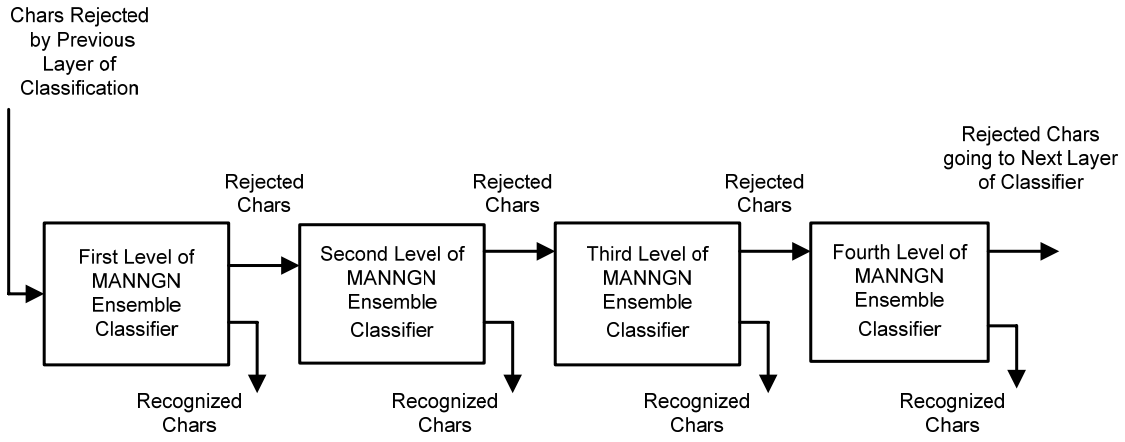


Fig. 7.2 Schematic diagram of one layer of classification in a cascade structure

We have implemented different frameworks of the combination of multi-ANN classifiers and their gating networks.

- **Training and Testing Procedures for the Cascade Ensemble Classifier System**

In the training procedure, for an ANN classifier at any level of any layer of the cascade classifier system, it is trained by the rejected characters in the previous level of the classifier. At the first level of the ANN classifier in each layer of classification, the ANN is trained by all the training samples with different feature sets as shown in Fig. 7.1, as follows: all the ANN classifiers in the first layer are trained by three randomly selected feature sets (I, II, III); all the ANN classifiers in the second layer are trained by the hybrid

feature sets (A, B, E); and all the ANN classifiers in the third layer are trained by the hybrid feature sets (C, D, F). Because the different feature sets are used at different layers of classification, it makes the classifiers complementary from the recognition point of view. The three randomly selected feature sets (I, II, III) and the six hybrid feature sets (A, B, C, D, E, F) have been discussed in chapter 4. At each level, the gating network is trained by the corresponding confidence values of the ANNs.

At the first level of the first layer of the cascade system, most of the characters should be correctly recognized; more difficult characters are rejected and sent to the higher level classifiers. In other words, we design classifiers at the higher levels and train them to recognize more difficult characters which are rejected by lower level classifiers.

Each ANN classifier is trained by pre-defined iterations and stopped when either iteration condition and/or the recognition accuracy are met. Therefore, the appropriate training samples for all higher level classifiers are needed in order to maintain the cascade recognition structure. About 15%-25% of training samples are rejected in the training level. The rejected samples are used as the training samples for the next higher level classifier.

In the testing procedure, the characters rejected by the previous level classifier are fed into the recognizer at the next higher level for further recognition. The recognized characters are directly output for display on the computer screen or they are saved into the database. We used the MNIST handwritten digit database to test our scheme.

- **Advantages of the Cascade Ensemble Classifier System**

As the cascade classifier scheme is applied, the recognition system can use a rejection strategy to reject those characters with relatively low confidence values rather than taking

a risk to misrecognize them. The rejected characters are sent to the higher level of classifiers for further recognition.

A novel framework with gating networks is proposed for congregating the outputs of the multi-classifiers. At the same time, the gating networks can remedy the setback of the ANN classifiers. The gating networks help to improve the recognition rate and the reliability of the cascade recognition system significantly.

We used three ANNs and three gating networks to form an ensemble classifier. The output was voted on by three ANNs and three gating networks rather than depending on only one ANN. This mechanism was based on the democratic voting system so as to achieve a more reliable performance.

It has been proven in the previous chapter that three new randomly selected feature sets have more distinct abilities than any other original feature set for recognition. We used three new randomly selected feature sets as the inputs of the ANN classifiers in the first layer in order to achieve a better recognition performance.

For the inputs, different feature sets are fed into different layers of classification. The correlation among the feature sets is relatively low and they are somewhat complementary in terms of discriminant ability for different numerals. By this process, we can explore more discriminant capabilities of feature sets for recognition. Experiments have demonstrated that the hybrid features are useful for achieving a better result.

7.2 Three Ensemble Classifier Schemes

In the design of the ensemble classifiers, we developed three schemes.

Scheme I: three parallel ANNs are combined by the majority vote or the sum vote.

Scheme II: a gating network (GN) is used to congregate the weighted outputs of three ANNs.

Scheme III: three gating networks (each classifier using one gating network) are used to congregate the weighted confidence values of the three ANNs, respectively. The final recognition result is based on the outputs of the three classifiers and the outputs of the three gating networks.

In scheme III, we call the new mixture of classifiers a "congregation scheme". The goal of the congregation scheme is to increase the recognition reliability by using a double-check mechanism (gating networks) on the confidence values of the three ANN classifiers.

When using a congregation scheme, each ANN is trained individually by different features extracted from the training samples. From our experimental results, it has been observed that for any ANN, it is very difficult to achieve a 100% recognition rate on a large scale training set (for example, 60,000 training samples of the MNIST dataset). Consequently, the recognition rate on the testing set will not achieve a 100% performance. In order to increase the recognition rate, a gating network is used to remedy the confidence values of the classifiers. The gating network serves two purposes. Firstly, it tries to correctly recognize those characters with low confidence values shown on any one or all of the classifiers. Secondly, it tries to correctly recognize the ones which are misclassified while testing.

1) Combination Scheme I

Scheme I consists of a combination of simple classifiers. Each classifier consists of three ANNs as shown in Fig. 7.3. The output is the combination of the three classifiers.

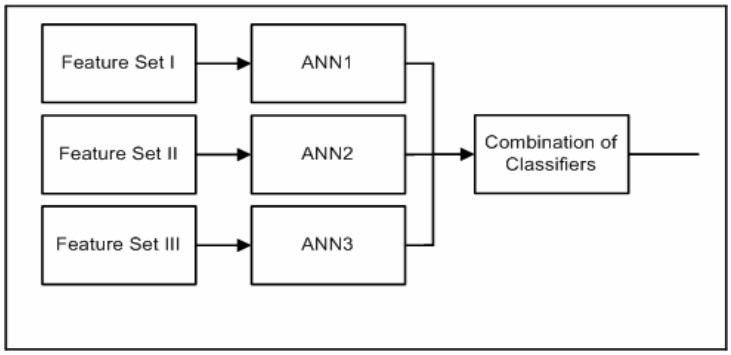


Fig. 7.3 One level of simple multi-ANN classifier

The rule for recognition (scheme I) is:

A numeral is accepted if: 1) three classifiers vote for the same numeral at the same time, where the sum of the confidence values is equal to or larger than 2.4, or the confidence value of each ANN is larger than 0.70; 2) the sum of the confidence values of any two ANNs is larger than 1.99 and they both vote for the same numeral; otherwise, the numeral is rejected.

- **Configuration of ANN**

For each ANN, we use a three-layer structure:

Input layer ----------Hidden layer ---------- Output layer

where, the number of nodes at the input layer = the number of the input features.

The number of nodes at output layer = 10 (representing 10 numerals).

The number of nodes at the hidden layer depends on the number of training samples. Table 7.1 lists the number of the hidden layer nodes with different numbers of training samples.

Table 7.1 List of no. of nodes at hidden layer with different no. of training samples

No. of training samples (x)	No. of nodes at hidden layer
$x \geq 20{,}000$	150
$10{,}000 \leq x < 20{,}000$	100
$5000 \leq x < 10{,}000$	50
$2000 \leq x < 5000$	30
$x < 2000$	10

As scheme I only uses a simple combination of three ANNs, we will propose other schemes with gating networks.

2) Classifier Scheme II

A new combination scheme of classifiers, which congregates three ANNs and a gating network, is proposed. The schematic diagram is shown in Fig. 7.4. The output confidence values of three ANNs are weighted by w_{10}~w_{19} for ANN1, w_{20}~w_{29} for ANN2, and w_{30}~w_{39} for ANN3 (note: w_{10}~w_{19} refers to the weights of the confidence values c_{10}~c_{19} of ANN1, and so on). A gating network is used to congregate the weighted confidence values.

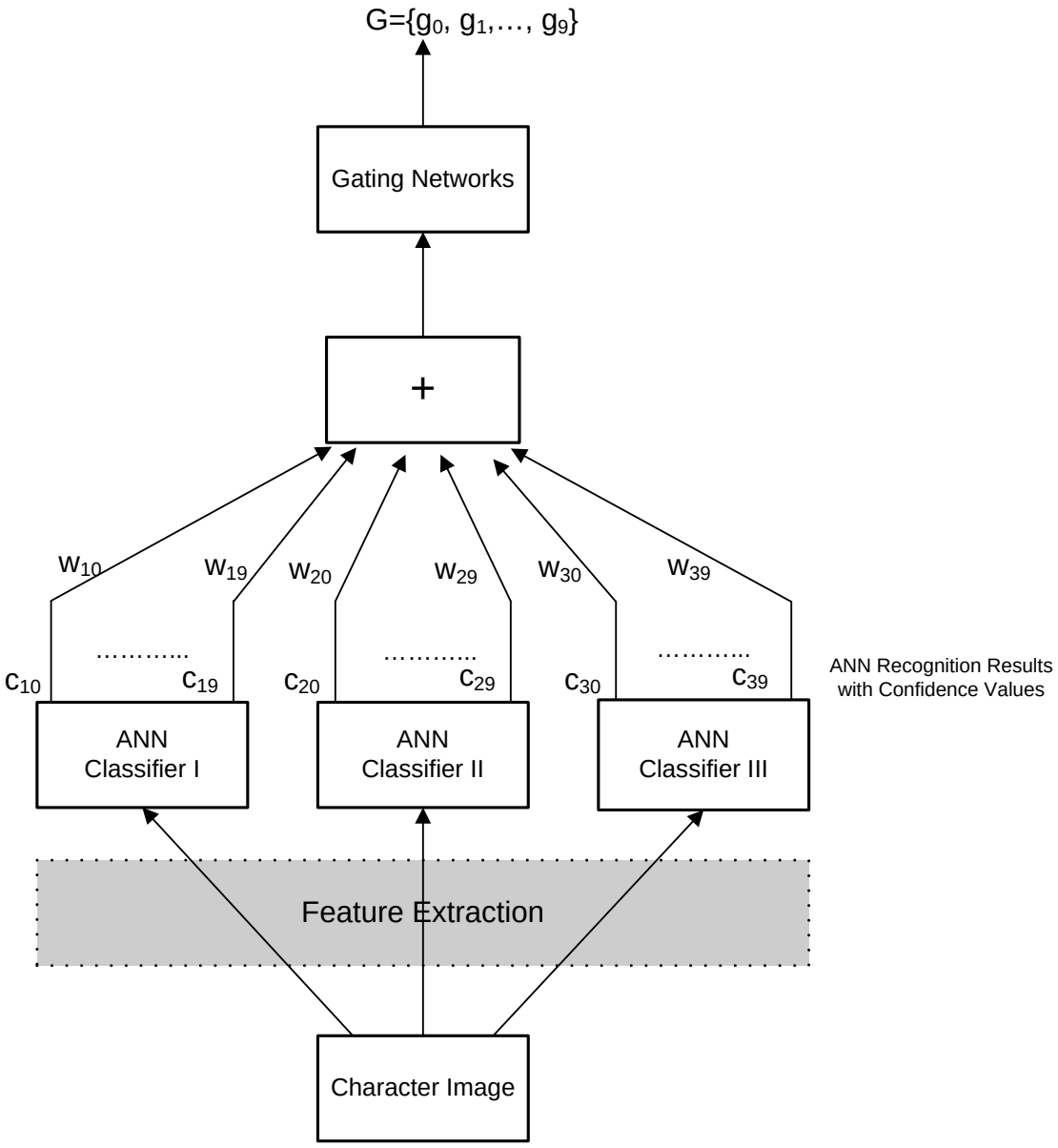


Fig. 7.4 An ensemble classifier consisting of three ANNs and one gating network

A genetic algorithm is used to evolve the optimal weights for the gating network from the confidence values of three ANNs.

Suppose the outputs of three ANNs are represented as: $\{c_{10}, c_{11},\ldots,c_{19}\}$, $\{c_{20}, c_{21},\ldots,c_{29}\}$, $\{c_{30}, c_{31},\ldots, c_{39}\}$, respectively.

The weighted outputs of the ANNs' confidence values can be calculated as follows:

$$X_i = W_i^T . C_i \qquad \ldots\ldots \quad (7.1)$$

where $W_i = [w_{i0}, w_{i1}, ..., w_{i9}]$, $C_i = [c_{i0}, c_{i1}, ..., c_{i9}]$ *i=1,2,3,* for three ANNs.

Add three weighted confidence values into a Y vector:

$$Y = \sum_{i=1}^{3} X_i \qquad \ldots\ldots \quad (7.2)$$

$$Y = [y_0, y_1, ... y_9]$$

Then a generalized gating network is applied to Y.

$$g_j = \frac{e^{y_i}}{\sum_k e^{y_k}} \qquad \ldots\ldots \quad (7.3)$$

$$G = [g_0, g_1, ..., g_9]^T$$

G is the output of the gating network.

Our goal is to pursue a lowest misrecognition rate and at the same time to seek the highest recognition performance. We can create a vector O_{target} with 10 elements. In the vector, the value of the corresponding label is set equal to 1.0, while others are set equal to 0.0. A fitness function f is chosen to minimize the difference between the output G and the corresponding training sample vector O_{target}, as follows:

$$f = | G - O_{target} |^2 \qquad \ldots\ldots \quad (7.4)$$

By minimizing the equation (7.4) through a genetic evolution, the weights tend to be optimal. Then, the recognition criterion is set as follows:

- A numeral is accepted if: 1) three ANN classifiers vote for the same numeral at the same time, where the sum of the confidence values is equal to or larger than 2.4, or 2) the gating network votes for a numeral, where the confidence value of the gating network is larger than 0.85, or 3) the sum of the confidence values of any two ANNs is larger than 1.99 and they both vote for the same numeral and the gating network votes for the same numeral. Otherwise, the numeral is rejected.

As only one gating network is used in this scheme, the congregated outputs of the gating network consist of ten nodes, which represent ten numerals. The confidence values of the three ANNs and the ten outputs of the gating network are used to make a decision about the final recognition results. However, there may still be some errors in the testing samples, so we propose another new scheme which includes three ANNs and three gating networks in the scheme III.

3) Classifier Combination Scheme III

We propose another classifier combining scheme shown in Fig. 7.5.

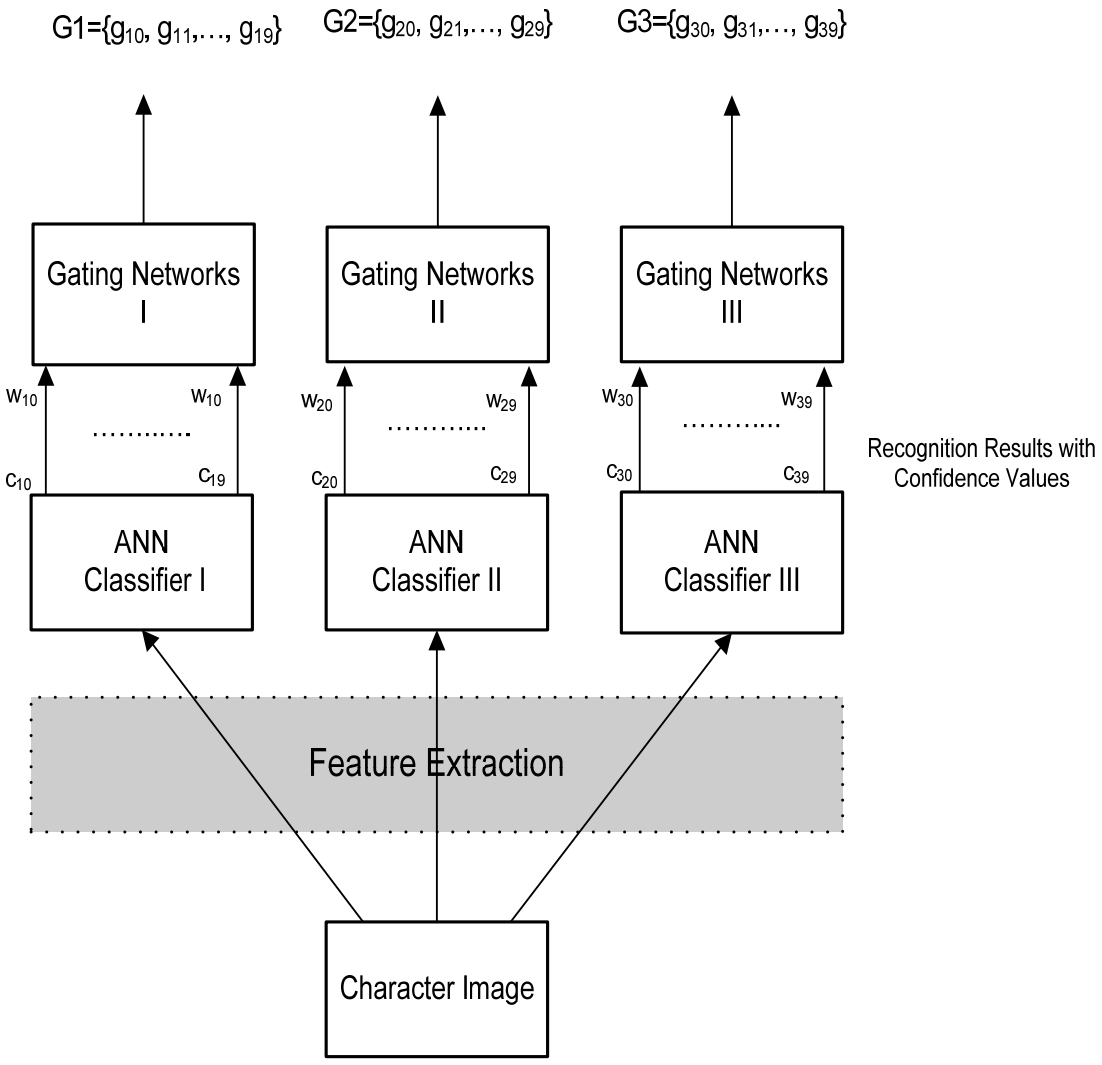


Fig. 7.5 An Ensemble classifier consisting of three ANNs and three gating networks

This new scheme includes three gating networks. Each gating network is used to congregate one of ANN's outputs.

Three ANN networks produce their outputs $C_i = \{ c_{ij} \}$ as a generalized confidence value vector ($0 < c_{ij} \leq 1.0$). Here i=1,2,3 for three ANNs; j=0,1,2,…9 for ten numerals; and c_{ij} denotes the confidence value for the jth nodes of the ith ANN.

The *i*th gating networks receive each classifier's confidence values C_i, where C_i is an input vector and the gating networks produce the weighted outputs. An intermediate variable ξ_{ij} can be calculated as:

$$\xi_{ij} = w_{ij} c_{ij} \quad \text{..... (7.5)}$$

Under the condition: $\sum_j c_{ij} = 1.0$; the weight w_{ij} is chosen in such a way that it optimizes the gating network's output in order to achieve the highest recognition rate in the system.

In order to generalize the output, the *ij*-th output g_{ij} of the gating networks is the "softmax" function of ξ_{ij}, as follows:

$$g_{ij} = \frac{e^{\xi_{ij}}}{\sum_k e^{\xi_{ik}}} \quad \text{..... (7.6)}$$

where k=0,1,…,9.

Equation (7.6) is the *j*th unit of the *i*th gating network and it satisfies the following condition:

$$\sum_j g_{ij} = 1 \quad \text{..... (7.7)}$$

where, we have: $G_i = \{g_{i0}, g_{i1}, ..., g_{i9}\}$, *i=1,2,3* for three classifiers, G_i is an output vector of the gating network *i*.

The following equation can be used to train three gating networks in order to obtain the best recognition performance:

$$\sum_{i=1}^{3} | O_{target} - G_i |^2 \Rightarrow 0 \quad \text{..... (7.8)}$$

From our experiments, we obtained the empirical criteria for recognition based on the combination of the outputs of the three ANNs and the outputs of the three gating networks, which are listed below:

1) If three ANNs vote for the same numeral at the same time and the three gating networks vote for the same numeral, which is the one that the three ANNs voted for, where each ANN's confidence value is at least 0.60, then the numeral is accepted.

2) If three gating networks vote for the same numeral and the confidence value of any of the two gating networks is greater than 0.65, while the confidence value of the third gating network is at least greater than 0.45 and it is the highest confidence value in the 10 nodes of the third gating network, then the numeral is accepted.

3) If two of three ANNs vote for the same numeral, where the sum of the confidence values of the two ANNs is greater than 1.99 and the remaining ANN may vote for another numeral and the confidence value of the remaining ANN is smaller than 0.6, then the numeral is accepted as the label chosen by the two ANNs.

4) If none of the above condition is met, the numeral is rejected.

The rejected numeral is sent to the higher level ensemble classifiers for recognition. The thresholds used in the criteria are empirically given based on experiments.

7.3 Genetic Algorithms for Training Gating Networks

Genetic algorithms (GAs) offer a particularly attractive approach to optimization since they generally perform an effective search of large, non-linear spaces [51, 86, 96, 104].

The genetic algorithm was developed based on Darwinian evolution and natural selection for solving optimization problems. GA applies evolution-based optimization techniques of selection, mutation, and crossover to a population for computing an optimal solution. The problem of the weight selection in the gating network is well suited to the evolution by GAs.

In the normal combination of classifiers such as majority voting or sum voting, the mutually dependent information among classifiers concerning their individual discriminating power is neglected. However, our proposed scheme congregates the classifiers' outputs with their confidence values through weighted linear combination and nonlinear generalization. GA-based gating networks have the following advantages:

1) Given a vector X, with n-dimensional random weights, the task of the GAs is to find a vector of the weights that minimizes the fitness function.

2) Given an appropriate criterion function, the GAs can evolve to an optimal solution globally at only a time complexity of $O(n)$.

In the handwritten character recognition area, the most difficult problem is to find a reasonable fitness function for a large set of training samples. The recognition rate can be used as a fitness criterion for a training classifier. However, it is unfeasible for an ANN to be used as a classifier because it needs huge computations for each generation of learning.

In this thesis, we use GAs to train the gating networks. When equation (7.4) or equation (7.9) is used as the fitness function, our GAs pursue the smallest difference between the gating networks' outputs and the target label vector O_{target}. The input of each gating

network is the corresponding ANN's output, which was trained by training samples beforehand. The following is a description of steps of our genetic algorithms.

Chromosome Representation

There are three ANNs in our system. Each ANN's outputs have 10 nodes. A chromosome is a vector consisting of 30 weights. Each chromosome component is a real number. The length of one chromosome is 10x3=30.

A chromosome is presented as:

$[\ w_{1,0}\ \ w_{1,1},\ \ \ldots,\ \ \ w_{1,9}\ \ \ w_{1,10}\ \ w_{1,11}\ \ \ldots\ \ w_{1,19}\ \ w_{1,20}\ \ w_{1,21}\ \ldots\ \ \ w_{1,29}\ ]$

|-10 weights for ANN1--| |-10 weights for ANN2-| |-10 weights for ANN3-|

Population Initialization

The initial chromosomes P (48 populations in this thesis), are randomly created (0.0~1.0):

Chromosome 1: $[w_{1,0}\ \ w_{1,1}\ \ldots\ w_{1,29}]$

Chromosome 2: $[w_{2,0}\ \ w_{2,1}\ \ldots\ w_{2,29}]$

……..

Chromosome P: $[w_{P,0}\ \ w_{P,1}\ \ldots\ w_{P,29}]$

Selection

The best 24 chromosomes with minimum fitness values, taken from 48 populations in each generation, are chosen to go into the mating pool.

Fitness Computation

In the MNIST database, there are 60,000 training samples. One target label vector O_{target} is created for each training sample. The inputs of the gating networks consist of the ANNs' confidence values. Equation (7.4) or equation (7.8) is used as a fitness function.

Crossover

Crossover occurs when information is exchanged between two parent chromosomes and the new information is introduced to children chromosomes. A single offspring parameter value, w_{new}, comes from a combination of the two corresponding parent parameter values. The crossover begins by randomly selecting a parameter *a* in a pair of parents, which is a crossover point. The crossover is calculated as follows:

$$a = roundup\{random * (M-1)\}$$

$$parent1(mother) = [w_{m0}, w_{m1}, w_{m3}, ...w_{ma}, ..., w_{mM-1}]$$
$$parent2(father) = [w_{d0}, w_{d1}, w_{d2}, ...w_{da}, ..., w_{dM-1}]$$

where *M* is the length of the weight vector. The subscripts *m* and *d* in the weight parameters (w_{mi},w_{di}) represent the mother and the father in the mating pool. Then, the selected parameters are combined to form new parameters. Two new weights are calculated as follows:

$$w_{new1} = w_{ma} - \beta[w_{ma} - w_{da}]$$
$$w_{new2} = w_{da} + \beta[w_{ma} - w_{da}] \qquad \ldots\ldots (7.10)$$

where ß is a random value between 0.0 and 1.0. The next step is to exchange the right parts of two parents, consisting of the crossover point to the end for each parent.

$$offsprings1 = [w_{m0,} w_{m1,} w_{m2,...,} w_{new1,}, ..., w_{dM-1}]$$

$$offsprings2 = [w_{d0,} w_{d1,} w_{d2,...,} w_{new2,}, ..., w_{mM-1}]$$

Mutation

In our experiment, the mutation rate is set at 0.01. According to the mutation rate, we randomly replace the w_{mi} (w_{di}) with a new weight element, which is produced by multiplying the old weight value with a new uniform random number (0.0-1.0).

Termination Criteria

In the training procedure, termination occurs when either the number of iterations reaches its defined number or the fitness value converged, so that the weights in chromosome pool are stable.

7.4 Experimental Results

We conducted five experiments based on the hybrid features and the various cascade recognition schemes. The MNIST database, which includes a set of 60,000 training samples, and a different set of 10,000 testing samples, are used in the following experiments. The experimental results are listed below:

1) Experiment One

Experiment one consisted of a series of six sub-experiments. The six sub-experiments were conducted only on the first layer of the cascade classification structure in Fig. 7.1, using different amounts of training samples with three sets of randomly selected features.

In a cascade classification structure, one layer of the cascade classification can be composed of more than three serial hierarchical classifiers depending on the number of training samples. A classifier consists of three ANNs without any gating network. The

rejected training samples in the classifier at any level will be regarded as the training samples for the next higher level classifier until the rejected samples go through all of the classifiers. The remaining characters may be recognized or rejected in the final stage. Three randomly selected feature sets: Feature Set I: 200 (number of feature dimensions; this notation will be used in the following experiments), Feature Set II: 218, Feature Set III: 240, as described in chapter 4, were used in Experiment One. We conducted a series of experiments with different numbers of training samples varying from 10,000 to 60,000. Here, we list six results.

- **Experiment I-A: 10,000 training samples used**

The hierarchical recognition system was trained by the first 10,000 training samples, and tested on the same set of 10,000 training samples and another set of 10,000 testing samples of the MNIST database, respectively. We used the rejection rule of the combination scheme I in Section 7.2. Three levels of classifications were used in the experiment. The recognition results conducted on the 10,000 training samples and the 10,000 testing samples are listed in Table 7.2.

Table 7.2 Recognition results of the hierarchical structure trained by 10,000 training samples

Testing category	Classifier I	Classifier II	Classifier III
No. of Rejection for 10,000 training samples	**1607**	**928**	**44**
No. of Misrecognition for 10,000 training samples	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**732**	**636**	**474**
No. of Misrecognition for 10,000 testing samples	**22**	**10**	**8**

In summary, the overall recognition results conducted on the 10,000 testing samples are listed below:

No. of Misrecognized Numerals = 40

No. of Rejected Numerals = 474

Correct Recognition Rate = 94.86%

Reliability Rate: (10000-40)/10000= 99.60%

- **Experiment I-B: 20,000 training samples used**

The ANN classifiers were trained by the first 20,000 training samples, and tested on 10,000 testing samples of the MNIST database. As the training samples were increased to 20,000, the number of hierarchical classifier levels was increased to five. Table 7.3 lists the recognition results conducted on the 20,000 training samples and 10,000 testing samples.

Table 7.3 Recognition results of hierarchical structure trained by 20,000 training samples

Testing category	Classifier I	Classifier II	Classifier III	Classifier IV	Classifier V
No. of Rejection for 20,000 training samples	**2778**	**866**	**388**	**158**	**100**
No. of Misrecognition for 20,000 training samples	**0**	**0**	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**505**	**445**	**406**	**375**	**323**
No. of Misrecognition for 10,000 testing samples	**9**	**2**	**2**	**9**	**6**

In summary, the overall recognition results conducted on the 10000 testing samples are listed below:

No. of Misrecognized Numerals = 28

No. of Rejected Numerals = 323

Correct Recognition Rate = 96.49%

Reliability Rate: (10000-28)/10000 = 99.72%

- **Experiment I-C: 30,000 training samples used**

The ANN classifiers were trained by the first 30,000 training samples, and tested on 10,000 testing samples of the MNIST database. Table 7.4 lists the recognition results conducted on the 30,000 training samples and 10,000 testing samples.

Table 7.4 Recognition results of hierarchical structure trained by 30,000 training samples

Testing category	Classifier I	Classifier II	Classifier III	Classifier IV	Classifier V
No. of Rejection for 30,000 training samples	**5320**	**2565**	**1589**	**520**	**70**
No. of Misrecognition for 30,000 training samples	**0**	**0**	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**450**	**408**	**375**	**335**	**254**
No. of Misrecognition for 10,000 testing samples	**8**	**2**	**2**	**7**	**5**

In summary, the overall recognition results conducted on the 10000 testing samples are listed below:

No. of Misrecognized Numerals = 24

No. of Rejected Numerals = 254

Correct Recognition Rate = 97.22%

Reliability Rate: (10000-24)/10000 = 99.76%

- **Experiment I-D: 40,000 training samples used**

The ANN classifiers were trained by the first 40,000 training samples, tested on 10,000 testing samples of the MNIST database. There were five hierarchical classifier levels for classification. The recognition results are shown in Table 7.5.

Table 7.5 Recognition results of hierarchical structure trained by 40,000 training samples

Testing category	Classifier I	Classifier II	Classifier III	Classifier IV	Classifier V
No. of Rejection for 40,000 training samples	**7582**	**4185**	**2783**	**748**	**50**
No. of Misrecognition for 40,000 training samples	**0**	**0**	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**338**	**311**	**271**	**246**	**189**
No. of Misrecognition for 10,000 testing samples	**8**	**5**	**3**	**2**	**2**

In summary, the overall recognition results conducted on the 10000 testing samples are listed below:

No. of Misrecognized Numerals = 20

No. of Rejected Numerals = 189

Correct Recognition Rate = 97.91%

Reliability Rate: (10000-20)/10000= 99.80%

- **Experiment I-E: 50,000 training samples used**

The ANN classifiers were trained by the first 50,000 training samples, and tested on 10,000 testing samples of the MNIST database. There were six hierarchical classifier levels. The recognition results are listed in Table 7.6.

Table 7.6 Recognition results of hierarchical structure trained by 50,000 training samples

Testing category	Classifier I	Classifier II	Classifier III	Classifier IV	Classifier V	Classifier VI
No. of Rejection for 50,000 training samples	**8532**	**5000**	**3620**	**2225**	**1256**	**155**
No. of Misrecognition for 50,000 training samples	**0**	**0**	**0**	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**324**	**296**	**268**	**221**	**186**	**163**
No. of Misrecognition for 10,000 testing samples	**3**	**3**	**2**	**2**	**2**	**2**

In summary, the overall recognition results conducted on the 10,000 testing samples are listed below:

No. of Misrecognized Numerals = 14

No. of Rejected Numerals = 163

Correct Recognition Rate = 98.23%

Reliability Rate: (10000-14)/10000= 99.86%

- **Experiment I-F: 60,000 training samples used**

The ANN classifiers were trained by 60,000 training samples, and tested on 10,000 testing samples of the MNIST database. There were seven hierarchical classifier levels. The recognition results are listed in Table 7.7.

Table 7.7 Recognition results of hierarchical structure trained by 60,000 training samples

Testing category	Classifier I	Classifier II	Classifier III	Classifier IV	Classifier V	Classifier VI	Classifier VII
No. of Rejection for 60,000 training samples	**9337**	**5227**	**4513**	**2249**	**1993**	**1200**	**150**
No. of Misrecognition for 60,000 training samples	**0**	**0**	**0**	**0**	**0**	**0**	**0**
No. of Rejection for 10,000 testing samples	**304**	**243**	**212**	**156**	**136**	**125**	**119**
No. of Misrecognition for 10,000 testing samples	**2**	**2**	**1**	**0**	**1**	**1**	**2**

In summary, the overall recognition results conducted on the 10,000 testing samples are listed below:

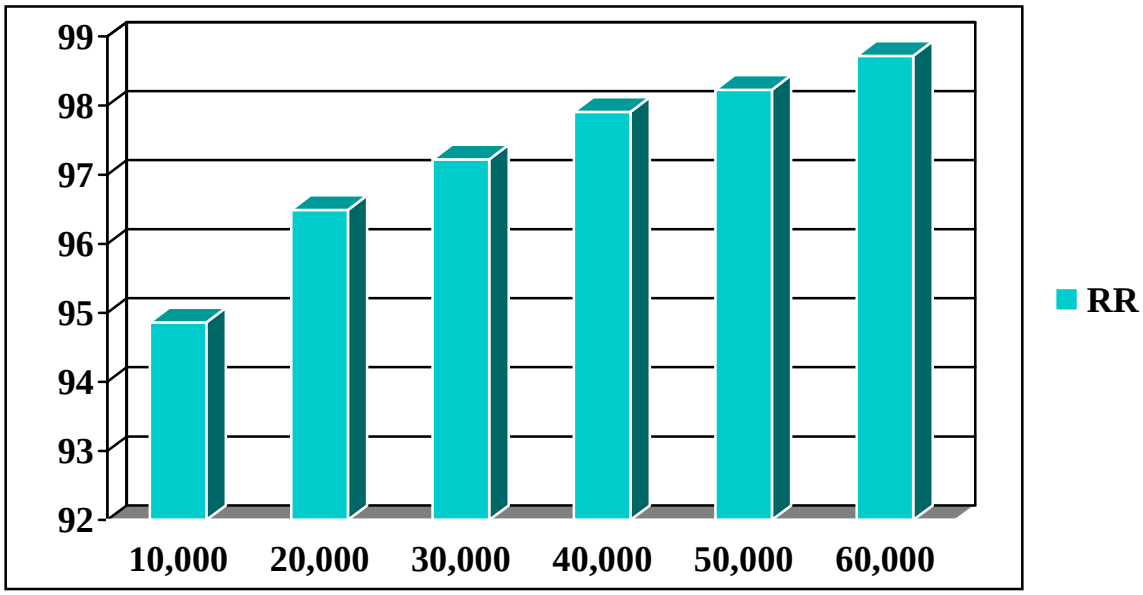
No. of Misrecognized Numerals = 9

No. of Rejected Numerals = 119

Correct Recognition Rate = 98.72%

Reliability Rate: (10000-9)/10000= 99.91%

Fig. 7.6 shows a bar graph of the recognition performances with different training samples.



Note: RR: Recognition Rate.

Fig. 7.6 Recognition rates conducted on the MNIST dataset with different training samples

From these sub-experiments, it can be concluded that when more training samples are used and more hierarchical levels of the classification are employed, the recognition rate will increase from 94.86% (10,000 training samples used) to 98.72% (60,000 training samples used). The reliability rate also increases from 99.60% to 99.91%. It means that more training samples help to increase recognition performance.

2) Experiment Two

The second experiment used three layers of the cascade recognition shown in Fig. 7.1. Each layer consisted of four levels of ensemble classifiers. Each ensemble classifier was composed of three ANNs without any gating networks.

Different feature sets were used at the different layers of classification in order to make the ensemble classifiers complementary. Three randomly selected feature sets (Feature Set I: 200, Feature Set II: 218, Feature Set III: 240) were used in the first layer. Three hybrid feature sets (Feature Sets A, B, E) were used in the second layer. Three hybrid feature sets (Feature Sets C, D, F) were used in the third layer. The majority or voting strategy was used to combine three classifiers. The rejection rule of the combination scheme I in section 7.2 was used.

During the testing procedure, the rejected characters at the previous layer were fed into the recognizer at the next layer for further recognition. The recognized characters were directly output for display on screen or saved into the database.

In summary, the testing results for each of the three layers are listed below:

First Layer

No. of Testing Samples = 10,000

No. of Misrecognized Chars = 5

No. of Rejected Chars = 214

Correct Recognition Rate = (10000-214-5)/10000=97.81%

Reliability Rate = (10000-5)/10000 = 99.95%

Second Layer

No. of Testing Samples=214

No. of Misrecognized Chars = 2

No. of Rejected Chars = 140

Accumulated Recognition Rate = (10000-140-7)/10000 = 98.53%

Accumulated Reliability Rate = (10000-7)/10000 = 99.93%

Third Layer

No. of Testing Samples = 140

No. of Misrecognized Chars = 2

No. of Rejected Chars = 103

Accumulated Recognition Rate = (10000-103-9)/10000 = 98.88%

Accumulated Reliability Rate = (10000-9)/10000 = 99.91%

Compared with the results of Experiment I-F with 60,000 training samples, the recognition rate of Experiment Two with 60,000 training samples has slightly increased from 98.72% to 98.88%. The final results showed that the total number of misrecognized numerals was nine in both experiments. We needed to further explore the classifier combination strategy in order to reduce the misrecognition rate while increasing the recognition rate.

3) Experiment Three

In Experiment Three, we used the same cascade scheme as in Experiment Two. Each ensemble classifier was composed of three ANNs and one gating network. The structure of the ensemble classifier was shown in Fig. 7.4.

This scheme congregated three ANN's recognition results into a gating network. As described in Section 7.2, the weights of the three ANNs' outputs were evolved by genetic algorithms in order to achieve ten optimal outputs for the gating network. The system was trained by 60,000 training samples, and tested on 10,000 testing samples. The decision rules were chosen from the combination scheme II in section 7.2. The results are listed below:

First Layer

No. of Testing Samples = 10,000

No. of Misrecognized Chars = 8

No. of Rejected Chars = 291

Recognition Rate = (10000-291-8)/10000= 97.01%

Reliability Rate = (10000-8)/10000 = 99.92%

Second Layer

No. of Testing Samples = 291

No. of Misrecognized Chars = 1

No. of Rejected Chars = 120

Accumulated Recognition Rate = (10000-120-8-1)/10000 = 98.71%

Accumulated Reliability Rate = (10000-8-1)/10000 = 99.91%

Third Layer

No. of Testing Samples = 120

No. of Misrecognized Chars = 0

No. of Rejected Chars = 89

Accumulated Recognition Rate = (10000-89-9)/10000 = 99.02%

Accumulated Reliability Rate = (10000-9)/10000 = 99.91%

Because a gating network was added to the outputs of three ANNs, the gating network was able to correct some errors that occurred in the ANNs' outputs. For example, if some confidence values of one ANN were relatively low, after the gating network's correction, the confidence values of the gating network were increased. Some rejected characters of ANNs were recognized by the gating network. Compared to Experiment Two, the recognition rate in Experiment Three increased from 98.88% to 99.02%. However, the misrecognition rate remained at the same level as Experiment Two.

4) Experiment Four

In Experiment Four, we used the same cascade recognition structure described in Experiments Two and Three, except that a new classification scheme shown in Fig. 7.5 was used. In this scheme, each ANN has a gating network for the confidence values' verifications and corrections.

The recognition rules, which consider three gating networks' outputs, were modified accordingly as described in the combination scheme III in Section 7.2. The experiment was conducted on the 60,000 training samples and tested on 10,000 testing samples of the MNIST dataset. The results are listed below:

First Layer

No. of Testing Samples = 10,000

No. of Misrecognized Chars = 4

No. of Rejected Chars = 168

Correct Recognition Rate = (10000-168-4)/10000 = 98.28%

Reliability Rate = (10000-4)/10000 = 99.96%

Second Layer

No. of Testing Samples = 168

No. of Misrecognized Chars = 0

No. of Rejected Chars = 94

Accumulated Recognition Rate = (10000-94-4)/10000 = 99.02%

Accumulated Reliability Rate = (10000-4)/10000 = 99.96%

Third Layer

No. of Testing Samples = 94

No. of Misrecognized Chars = 0

No. of Rejected Chars = 77

Accumulated Recognition Rate = (10000-77-4)/10000 = 99.19%

Accumulated Reliability Rate = (10000-4)/10000 = 99.96%

Up to now, the classifier, consisting of three ANNs and three gating networks, shows the best recognition performance in terms of the overall recognition accuracy and the overall reliability. The main reason for this improvement is that three gating networks are linked to the outputs of three ANNs (each ANN has one gating network). Therefore, each gating network can correct or remedy the errors of the corresponding ANN effectively. As a result, the overall recognition performance is increased.

5) Experiment Five

Sum Voting without rejection in the recognition of the rejected digits in Experiment Four

In Experiment Four, 81 characters (77 rejected characters in the third layer of the cascade system + 4 misrecognized characters in the first layer of the cascade system) were not correctly recognized in our proposed cascade ensemble recognition system. In order to investigate the tradeoff between the recognition, misrecognition and rejection rates on the 77 characters, different confidence thresholds (ANN_{conf}) in equation (6.8), which were related to reject parameter t in our error analysis, were chosen in this experiment.

Our sum voting experiment was conducted as follows: three random hybrid feature sets [Random Feature Set I (200), Random Feature Set II (218), Random Feature Set III (240)] were used as the inputs of three ensemble classifiers. The classifiers were trained

by 60,000 training samples of the MNIST dataset and were used to recognize 77 rejected characters by the sum voting scheme without any rejection option.

Fig. 7.7 shows the tradeoff of the numbers among the recognition, misrecognition and rejection categories conducted on the 77 rejected samples in Experiment Four by setting different confidence thresholds.

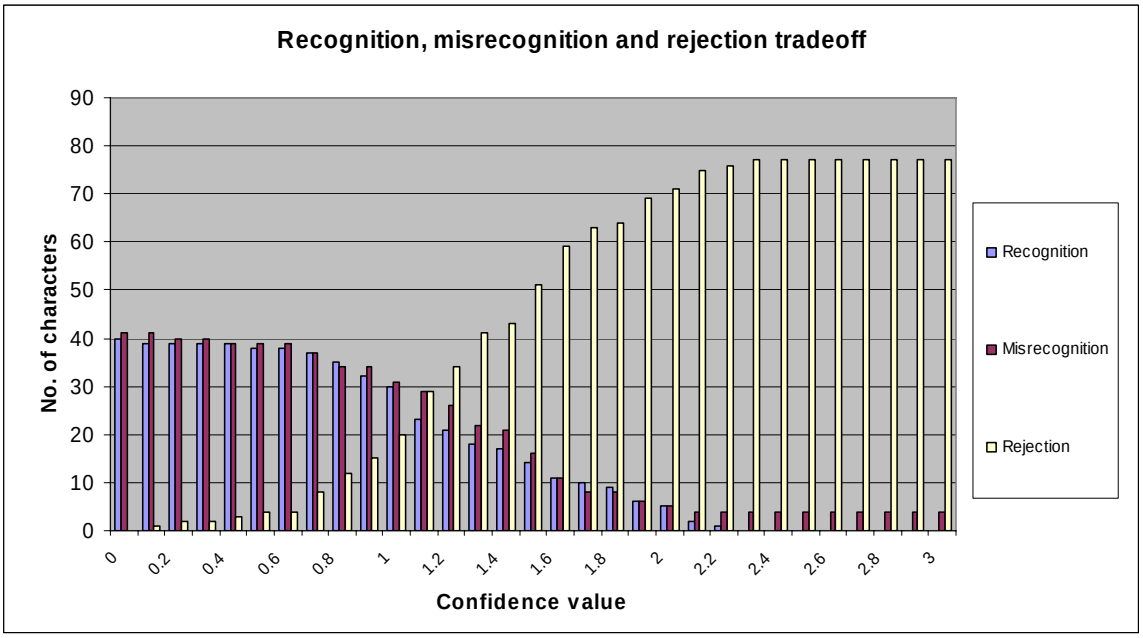


Fig. 7.7 Tradeoff among the recognition, misrecognition and rejection in the sum voting scheme without rejection option

In Fig. 7.7, if we reduced the confidence threshold from 3.0 to 0.0, the rejection number gradually decreased to 0; however, the misrecognition number increased from 4 to 41. At the same time, some of rejected digits in the previous level were correctly recognized.

Fig. 7.8 depicts the relationship curve between the substitution rate (misrecognition rate) and the rejection rate for our proposed cascade ensemble classifier system (Combining

scheme III in section 7.2: an ensemble classifier consists of three ANNs and three gating networks).

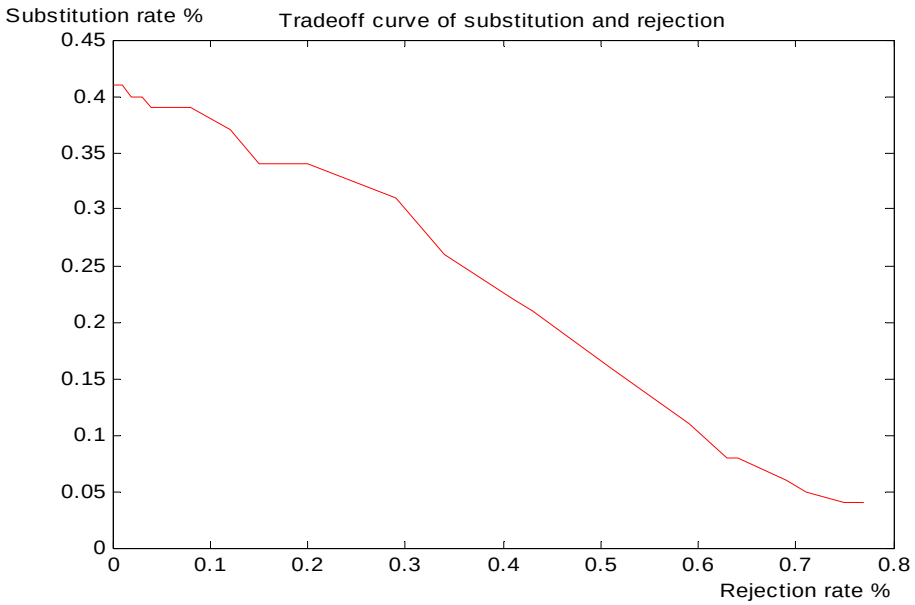


Fig. 7.8 Tradeoff curve between the rejection rate and substitution rate

Without rejection option, the recognition results conducted on the 77 rejected numerals using sum voting scheme are listed below:

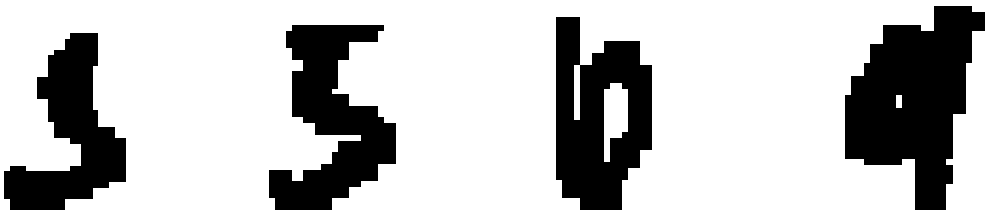
Number of misrecognized digits= 37

Number of correct recognition digits= 40

So the overall recognition rate of the cascade ensemble classifier system without rejection in the last layer of sum voting scheme is: (10000-37-4)/10000=99.59%.

The identification numbers (ID) and the original character images of the misrecognized digits in our proposed cascade ensemble classifiers are listed below:

- **Four digits were misrecognized in the first layer of the cascade system**

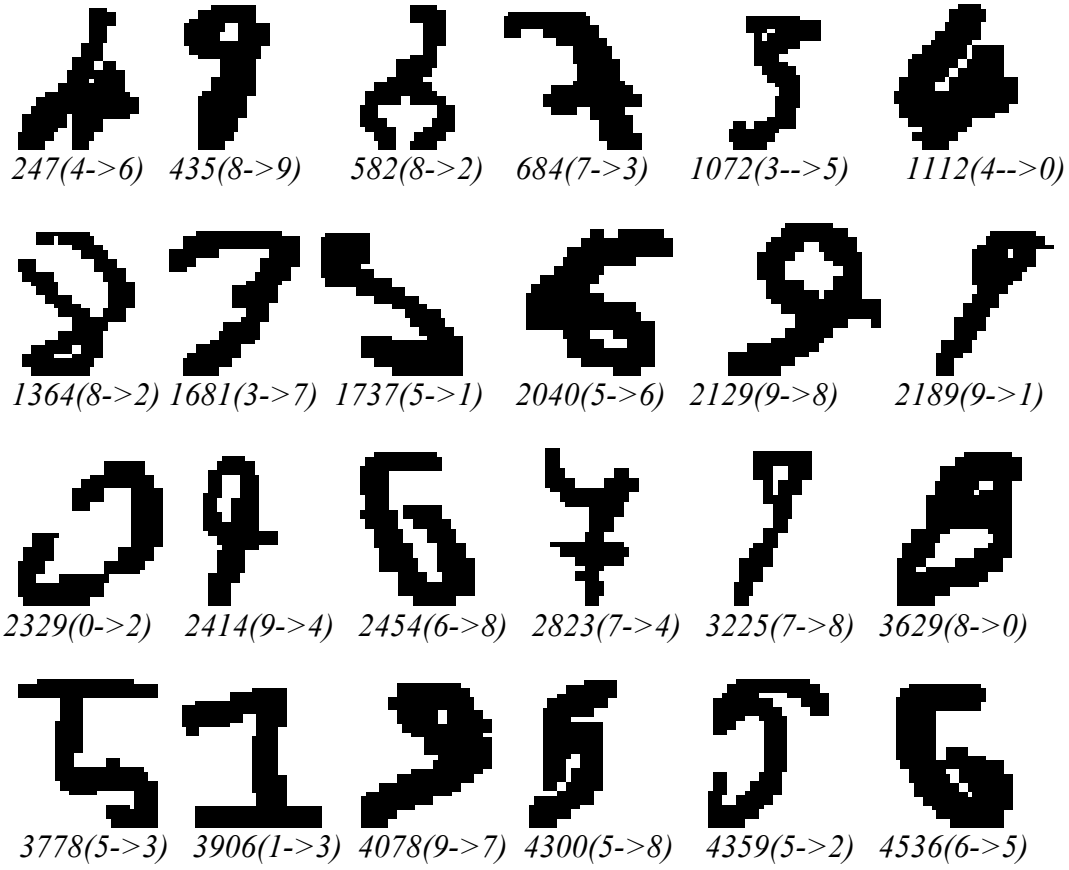


340(5->3) *674(5->3)* *3030(6->0)* *4911(4->9)*

Note: 1) four digits were misrecognized in the first layer of ensemble classifiers when the rejection strategy was applied.

2) *340(5->3)* means the digit ID 340, which was labeled as "5", but it has been misrecognized as "3". The testing ID is from 0 to 9999.

- **Here is the list of 37 misrecognized digits using the sum voting scheme for the recognition of the 77 rejected digits:**

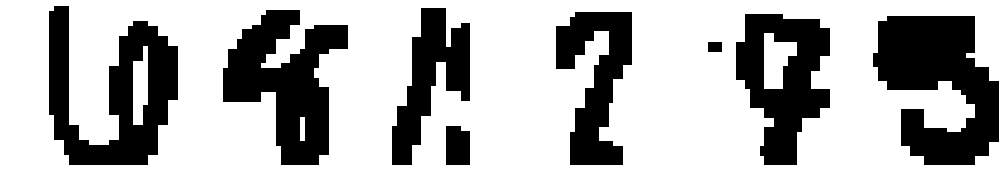


247(4->6) *435(8->9)* *582(8->2)* *684(7->3)* *1072(3-->5)* *1112(4-->0)*

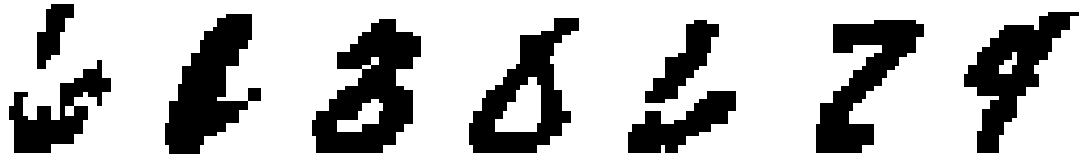
1364(8->2) *1681(3->7)* *1737(5->1)* *2040(5->6)* *2129(9->8)* *2189(9->1)*

2329(0->2) *2414(9->4)* *2454(6->8)* *2823(7->4)* *3225(7->8)* *3629(8->0)*

3778(5->3) *3906(1->3)* *4078(9->7)* *4300(5->8)* *4359(5->2)* *4536(6->5)*



4814(6->4) 4956(8->4) 5457(1->4) 5654(7->2) 5858(7->4) 6053(5->3)



6558(6->3) 6783(1->6) 7595(3->8) 8325(0->8) 9679(6->3) 9811(2->7) 9922(4->9)

From this experiment, we can achieve a very high recognition performance using a sum voting strategy without rejection option in the last layer of the cascade ensemble classifier system.

For those misrecognized characters shown above, we may use the geometrical features introduced in Section 4.1.7 to help improving the overall recognition rate. For example, for the recognition of the misrecognized digit “8” (ID: 1364) shown at P. 139, the middle line feature extraction method can be used to extract two loops on the character image vertically. Even the top part of the first loop is opened. The two loop features can be used to distinguish digit “8” from digit “2” easily because digit “2” has not two loops aligned vertically.

7.5 Comparison of Three Cascade Schemes

The effect of increasing the training samples has been discussed in the previous sections. Now, we will discuss the effects of recognition performance by using different ensemble classifiers in the cascade recognition system.

The comparisons of the recognition rates and the numbers of the misrecognition from Experiment Two to Experiment Four are shown in Figs. 7.9 and 7.10.

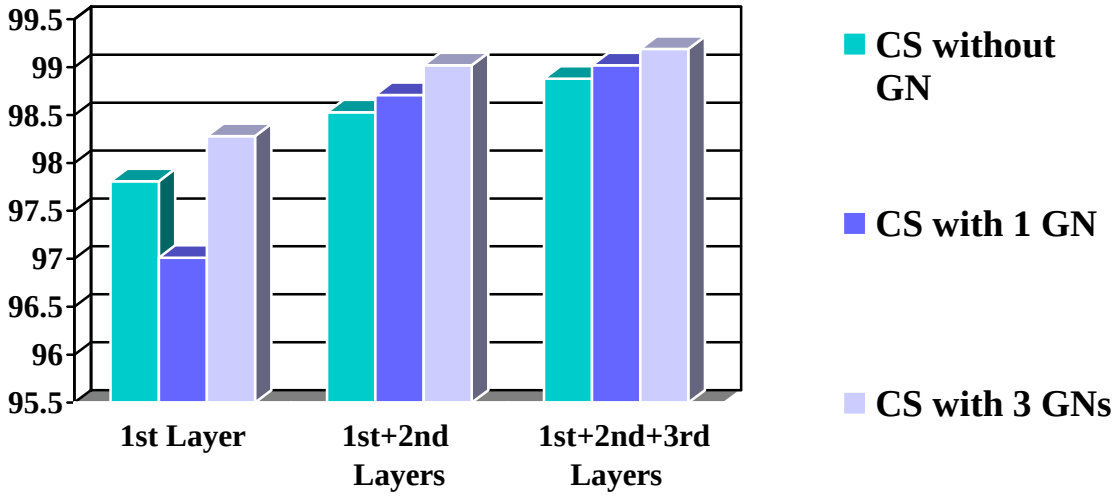


Fig. 7.9 Comparison of recognition rates for experiments two-four

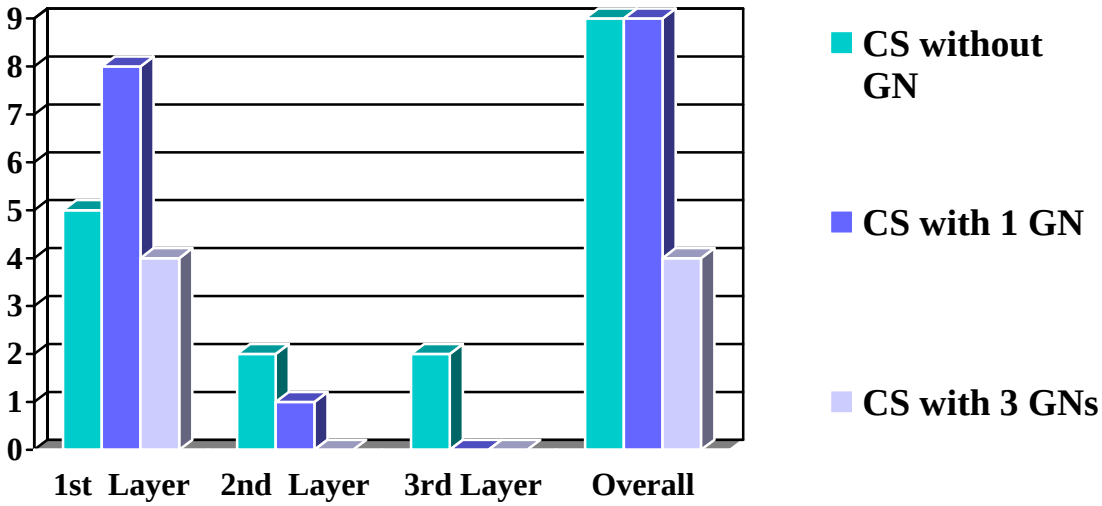


Fig. 7.10 Comparison of misrecognition numbers for experiments two-four

Notes: CS without GN: cascade recognition structure without gating network (Experiment Two);

CS with 1 GN: cascade recognition structure with one gating network (Experiment Three);

CS with 3 GN: cascade recognition structure with three gating networks (Experiment Four);

The detailed information for the four schemes is shown in Table 7.8.

Table 7.8 Recognition performances of three cascade schemes

Scheme	Recognition Rate (%)	No. of Rejected chars	No. of Errors	Recognition Reliability Rate (%)
CS without GN	98.88	103	9	99.91
CS with 1 GN	99.02	89	9	99.91
CS with 3GN	99.19	77	4	99.96

In Experiment Four, we developed three gating networks, which were linked to the three ANN's outputs, so as to congregate each ANN's confidence values individually. Therefore, the final recognition results were voted on both three ANNs' outputs and three gating networks' outputs, and the best recognition performance was achieved.

We conducted several experiments. The experiments demonstrated that: 1) the more training samples were used in training procedure, the higher the recognition rate could be achieved; 2) Hybrid feature sets were helpful in seeking a lower misrecognition rate; 3) The cascade ensemble classifier recognition with each ANN linking to one gating network (An ensemble classifier) can achieve the best recognition performance in terms of the highest recognition rate, and the highest reliability. Our proposed cascade ensemble classifier system can achieve 99.96% reliability and 99.19% recognition rate with rejection strategies or 99.59% recognition rate without rejection option in the last layer of the cascade classification system.

Table 7.9 lists the recognition performance comparison of our proposed cascade ensemble classifier system with other latest recognition systems published in the literature.

Table 7.9 Recognition comparison of our proposed cascade ensemble classifier system with other recognition systems with rejection strategy

Method	Database	Recognition Rate (%)	Rejection Rate (%)	Misrecognition Rate (%)	Reliability (%)
Hierarchical Classifier [98] Pattern Recognition, 2002	NIST	86.68%	13.23%	0.01%	99.99%
An Optimized Hill Climbing Algorithm [81] IWFHR'9, 2004	NIST SD 19	99.10%	0.00%	0.90%	99.10%
Rejection Strategy for Convolutional Neural Networks [12] ICDAR, 2005	MNIST	92.12%	7.63%	0.25%	99.75%
GP-based Secondary Classifiers [111] Pattern Recognition, 2005	NIST	90.00%	9.70%	0.03%	99.70%
Our proposed Cascade Ensemble Classifier System 2006	MNIST	99.19%	0.77%	0.04%	99.96%

7.6 Recognition Speed

We conducted an experiment to calculate the cascade ensemble classifier system's speed. For example, if we used three ANNs and three gating networks as an ensemble classifier, and we used the cascade structure shown in Figs. 7.1 and 7.2, for 10,000 testing samples, then the recognition time is about 100 seconds. This time includes the reading and saving of data from disk without considering the time for feature extraction. The classification speed for our system is 100s/10,000 digits = 10 ms/digit.

For the feature extraction and random feature selection, approximately, 10,000 characters took about 300s, namely:

The feature extraction speed: 300,000/10,000 digits = 30 ms/character

So the recognition time for feature extraction and classification is about 40ms/digit.

Our cascade ensemble classifier system can recognize 25 digits per second.

All of the experiments were conducted on a Pentium® 4 personal computer, CPU 2.80GHz, 1.00 GB of RAM.

Chapter Eight

Conclusions and Discussions

In this thesis, many efforts were devoted to the recognition and verification of handwritten numeral recognitions. In pursuit of the highest recognition accuracy and the lowest misrecognition rate, we introduce a hybrid feature extraction strategy and a multi-modal nonparametric analysis for feature dimensionality reduction (in order to obtain a faster and more stable classifier training procedure for verification). The design of a cascade ensemble classifier recognition system with rejection strategies is also introduced. From a practical perspective, the various recognizers and verifiers were designed and implemented using novel hybrid feature extraction algorithms and a newly designed ensemble cascade classifier system. The designed OCR engines were applied to handwritten numeral recognition. A summary of thesis contributions and discussions on future direction is also addressed.

8.1 Summary of Thesis Contributions

It is common sense that if an OCR system can achieve an excellent recognition performance, the following two aspects must have played an important role: feature extraction and classification. In this thesis, our research focuses on: feature extraction and the design of a cascade ensemble classifier system in order to increase the recognition accuracy and reliability. The main contributions of this thesis are summarized below:

1) The highest recognition reliability and a minimal error rate for the recognition of handwritten digits have been achieved by the proposal of a novel cascade ensemble classifier system with rejection strategies.

Based on a theoretical analysis of the tradeoff of the error, rejection, and recognition rates of a cascade ensemble classifier system, three solutions were proposed: (i) extracting more discriminative features to attain a high recognition rate, (ii) using ensemble classifiers to suppress the error rate, and (iii) employing a novel cascade system to enhance the recognition rate and to reduce the rejection rate. Based on these strategies, novel gating networks were used to congregate the confidence values of three parallel ANN classifiers. The weights of the gating networks were trained by Genetic Algorithms (GAs) to achieve the overall optimal performance. The novel framework with gating networks could remedy the drawback of the ANN classifiers. It led to the significant improvement of both the recognition rate and the reliability of the recognition system. The cascade ensemble classifier system has a lower rejection rate and a higher recognition rate compared to the one-level ensemble classifier system. The error rate can be reduced by expanding the rejection space, or by setting a higher confidence threshold in the recognition system, or by using an ensemble logical "and" classifier. In the training procedure, for any classifier at any level of the cascade recognition system, it is trained by the rejected characters in the previous level of the classifier. The trained classifier can recognize those characters, which are not recognized by previous classifiers. While a cascade recognition scheme is applied, the recognition system can use rejection strategies to reject those characters with relatively low confidence values rather than taking a risk to misrecognize them. The rejected characters are sent to a higher level of classifiers for

further recognition. The comprehensive experiments demonstrated that our proposed cascade ensemble classifier system with gating networks can achieve very encouraging results, i.e. 99.96% reliability with a 99.19% recognition rate with rejection strategy, or a 99.59% recognition rate without rejection.

2) A novel multi-modal nonparametric analysis for feature dimensionality reduction for the verification of handwritten digits has been proposed.

The novel multi-modal nonparametric method utilizes only those training samples on and near the effective decision boundary to compute the between-class scatter matrix for optimal discriminant analysis. As we adopt the multi-modal discriminant analysis, the training data in each cluster are more centralized and the within-class scatter matrix will be less scattered than the mono-modal one. For the computation of the between-class scatter matrix, our method uses the quick-sort algorithm to sort the *k-NN* for each cluster, corresponding to every cluster in another class, so as to get the training data along the effective decision boundary. The adjacent *k-NN* training samples of two clusters from different classes are used to calculate the between-class scatter matrix. The optimal Fisher criterion based on our proposed method maximizes the between-class separability and minimizes the within-class separability in order to improve the system's discriminant ability for classification. The computational complexity of our proposed algorithm for calculating the between-class scatter matrix S_b is $O(N \log N)$, which is much smaller than the computation complexity of $O(N^2 \log N)$ used in other similar nonparametric discriminant analysis approaches [5, 31], therefore less CPU time is required for classifier training. Experiments demonstrated that our proposed method could achieve a high

feature compression performance without sacrificing its discriminant ability. The results of dimensionality reduction make the ANNs converge more easily. For the verification of confusing handwritten numeral pairs, our proposed algorithm was used to congregate features, and it outperformed the PCA and compared favorably with other nonparametric discriminant analysis methods.

3) The exploration of various hybrid feature extraction methods is highlighted in this thesis.

As we know, feature extraction is a vital step in the recognition problems. In this thesis, seven sets of feature extraction algorithms are proposed. Among the seven sets of features, two dimensional real wavelets and complex wavelets are used to extract directional-based wavelet features. Medial axial transformation algorithm-based gradient features have shown their excellent discriminative ability in the recognition of the handwritten numerals.

A simple and effective multi-class divergence analysis has been proposed for hybrid feature ranking and selection. By applying a random feature selection scheme to the seven sets of ranked features, three randomly selected hybrid feature sets were formed, which demonstrated a better recognition performance compared to the original feature sets.

8.2 Future Research Directions

Handwritten character recognition has been extensively researched for the past few decades and has always been a challenging topic. Basically, two goals are needed in the theoretical research and the practical implementations: to achieve the highest recognition rate and, at the same time, to maintain the lowest misrecognition rate, or the highest reliability. In the future, further research can include in the following aspects:

Theory can be further developed towards the cascade multi-class rejection rules. In our current system, the rejection rules, although effective, are mostly derived from empirical experiments. A systematic study of rejection strategies will be desirable for further theoretical work.

For the hybrid feature extraction methods, the fine features should be further investigated. Different features have different discrimination merits for different recognition purposes. When designing a verifier, the fine features can be employed. More importantly, the development of a new theory and algorithm in feature extraction is also important.

Finally, the combination of different classifiers, such as SVM, ANN, and *K-NN* embedded in a recognition system is also a suitable method of increasing recognition performance, because different classifier combinations have different merits in dealing with discriminant problems. The mechanism and structures of different classifier combinations need to be investigated.

Bibliography

1. F. M. Alkoot and J. Kittler, Experimental Evaluation of Expert Fusion Strategies, Pattern Recognition Letters, Vol. 20, No. 11, 1999, pp. 1361-1369.

2. A. Amin, H. B. Al-Sadoun, and S. Fischer, Hand-printed Arabic Character Recognition System Using An Artificial Network, Pattern Recognition Vol. 29, No. 4, 1996, pp. 663-675.

3. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Tree, Wadsworth International, 1984.

4. M. Bressan and J. Vitria, On the Selection and Classification of Independent Features, IEEE Transactions on PAMI, Vol. 25, No. 10, October 2003, pp. 1312-1317.

5. M. Bressan and J. Vitria, Nonparametric Discriminant Analysis and Nearest Neighbor Classification, Pattern Recognition Letters, Vol. 24, No. 5, 2003, pp. 2743-2749.

6. A. de S. Britto, R. Sabourin, F. Bortolozzi, and C. Y. Suen, The Recognition of Handwritten Numeral Strings Using A Two-stage HMM-based Method, International Journal on Document Analysis and Recognition, Vol. 5, No. 2, 2003, pp. 102-117.

7. R. M. Brown, T. H. Fay, and C. L. Walker, Handprinted Symbol Recognition System, Pattern Recognition, Vol. 21, No. 2, 1988, pp. 91-118.

8. C. J. C. Burges and B. Scholkopf, Improving the Accuracy and Speed of Support vector Learning Machine, Advanced in Neural Information Processing Systems 9, MIT Press, Cambridge, MA, 1997, PP. 375-381.

9. J. Cai, M. Ahmadi, and M. Shridhar, Recognition of Handwritten Numerals with Multiple Feature and Multi-stage Classifier, Pattern Recognition, VOL. 28, No. 2, 1995, pp. 153-160.

10. J. Cai, M. Ahmadi, and M. Shridhar, A Hierarchical Neural Network Architecture for Handwritten Numeral Recognition, Pattern Recognition, Vol. 30, No. 2, 1997, pp. 289-294.

11. M. Cannon, M. Fugate, D. R. Hush, and C. Scovel, Selecting a Restoration Technique to Minimize OCR Error, IEEE Transactions on Neural Networks, Vol. 14, No. 3, May 2003, pp. 478-490.

12. H. Cecotti and A. Belaid, Rejection Strategy for Convolutional Neural Network by Adaptive Topology Applied to Handwritten Digits Recognition, Proc. of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, August 2005, pp.765-769.

13. G. Y. Chen, T. D. Bui, and A. Krzyzak, Contour-Based Handwritten Numeral Recognition Using Multiwavelets and Neural Networks, Pattern Recognition, Vol. 36, No. 7, 2003, pp. 1997-1604.

14. Z. Chi, Q. Wang, and W. C. Siu, Hierarchical Content Classification and Script Determination for Automatic Document Image Processing, Pattern Recognition, Vol. 36, No. 11, November 2003, pp. 2483-2500.

15. S. B. Cho, Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals, IEEE Transactions on Neural Networks, Vol. 8, No. 1, 1997, pp.43-53.

16. S. Cho and J. H. Kim, Multiple Network Fusion Using Fuzzy Logic, IEEE Transactions on Neural Networks, Vol. 6, No. 2, 1995, pp.497-501.

17. C. K. Chow, On Optimum Recognition Error and Reject Tradeoff, IEEE Transactions on Information Theory, Vol.16, No. 1, January 1970, pp. 40-46.

18. C. K. Chu, Wavelets: A Mathematical Tool for Signal Processing, Philadelphia: Society for Industrial and Applied Mathematics, 1997.

19. L. Cun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, Vol. 86, No. 11, November 1998, pp. 2278-2324.

20. R. A. DeVore, Nonlinear Approximation, Acta Numerica, 1998, pages 51-151.

21. D. Decoste and B. Scholkopf, Training Invariant Support Vector Machines, Machine Learning, Vol. 46, No. 1-3, 2002, pp. 160-190.

22. J. X. Dong, Speed and Accuracy: Large-Scale Machine Learning Algorithms and Their Applications, Doctoral thesis, Computer Science Department, Concordia University, Montreal, October 2003.

23. J. X. Dong, A. Krzyzak, and C.Y. Suen, Fast SVM Training Algorithm with Decomposition on Very Large Datasets, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, No. 4, April 2005, pp. 603-618.

24. B. Dubuisson and M. Masson, A Statistical Decision Rule with Incomplete Knowledge About Classes, Pattern Recognition, Vol. 26, No. 1, Jan. 1993, pp. 155-165.

25. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, John Wiley & Sons, Inc., Wiley-Interscience, Second Edition, 2000.

26. D. H. Foley and J. W. Sammon, An Optimal Set of Discriminant Vectors, IEEE Transaction on Computer, Vol. C-24, No. 3, March 1975, pp. 281-289.

27. C. Frelicot and L. Mascarilla, Reject Strategies Driven Combination of Pattern Classifiers, Pattern Analysis and Applications, Vol. 5, No. 2, 2002, pp. 234-243.

28. J. H. Friedman, On Bias Variance 0/1-Loss and the Curse-of-Dimensionality, Data Mining and Knowledge Discovery, Vol. 1, No. 1, 1997, pp. 55-77.

29. K. Fukumizu, F. R. Bach, and M. I. Jordan, Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces, Journal of Machine Learning Research, Vol. 5, 2004, pp.73-99.

30. K. Fukunaga and W. L. G. Koontz, Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering, IEEE Transaction on Computer, Vol. C-19, No. 4, April 1970, pp. 311-318.

31. K. Fukunaga and J. M. Mantock, Nonparametric Discriminant Analysis, IEEE Transaction on PAMI, Vol. 5, No. 6, Nov. 1983, pp. 671-677.

32. K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, Inc, 2nd ed., 1990.

33. P. D. Gader and M. A. Khabou, Automatic Feature Generation for Handwritten Digit Recognition, IEEE Transactions on PAMI, Vol. 18, No. 12, pp. 1256-1261.

34. J. Gao and X. Q. Ding, On Improvement of Feature Extraction Algorithms for Discriminative Pattern Classification, 15th International Conference on Pattern Recognition, 2000, pp. 2101-2104.

35. M. D. Garris, C. L. Wilson, and J. L. Blue, Neural Network-Based Systems for Handprint OCR Applications, IEEE Transactions on Image Processing, Vol. 7, No. 8, August 1998, pp 1097-1112.

36. N. Giusti, F. Masuli, and A. Sperduti, Theoretical and Experimental Analysis of A Two-stage System for Classification, IEEE Transactions on PAMI, Vol. 24, No. 7, 2002, pp. 893-904.

37. S. Gunter and H. Bunke, Optimization of Weights in a Multiple Classifier Handwritten Word Recognition System Using a Genetic Algorithm, Electronic Letters on Computer Vision and Image Analysis Vol. 3, No. 1, 2004, pp. 25-41.

38. I. Guyon and A. Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, Vol. 3, 2003, pp. 1157-1182.

39. T. Hastie, R. Tibshirani, and A. Buja, Flexible Discriminant and Mixture Models, Proceedings of Neural Networks and Statistics Conference, Edinburgh, J. Kay and D. Titterington, Eds., Oxford University Press, 1995.

40. T. Hastie and R. Tibshirani, Discriminant Analysis by Gaussian Mixtures, Journal of the Royal Statistical Society: Series B (Statistical Methodology), Jan. 1996.

41. T. Hastie and R. Tibshirani, Discriminant Adaptive Nearest Neighbor Classification, IEEE Transactions on PAMI, Vol. 18, No. 6, 1996, pp. 607-616.

42. C. L. He, P. Zhang, J. X. Dong, C. Y. Suen, and T. D. Bui, The Role of Size Normalization on the Recognition Rate of Handwritten Numerals, Proceedings of Neural Networks and Learning in Document Analysis and Recognition, Seoul, Korea, August 29, 2005.

43. B. Heisele, T. Serre, S. Prentice, and T. Poggio, Hierarchical Classification and Feature Reduction for Fast Detection with Support Vector Machines, Pattern Recognition, Vol. 36, No. 9, 2003, pp. 2007-2017.

44. T. K. Ho and J. Hull, Decision Combination in Multiple Classifier Systems, IEEE Transactions on PAMI, Vol. 16, No. 1, Jan. 1994, pp.66-75.

45. T. K. Ho, The Random Subspace Method for Constructing Decision Forests, IEEE Transactions on PAMI, Vol. 20, No. 8, August 1998, pp. 832-844.

46. M. K. Hu, Visual Pattern Recognition by Moment Invariants, IRE Transactions on Information Theory, Vol. 8, February 1962, pp. 179-187.

47. Y. S. Huang and C. Y. Suen, An Optimal Method of Combining Multiple Classifiers for Unconstrained Handwritten Numeral Recognition, Proceedings of 3rd International Workshop on Frontiers in Handwriting Recognition, May 1993.

48. Y. S. Huang and C. Y. Suen, A Method of Combining Experts for the Recognition of Unconstrained Handwritten Numerals, IEEE Transactions on PAMI, Vol. 17, No. 1, 1995, pp. 90-94.

49. D. Ingrid, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, 1992.

50. M. I. Jordan and R. A. Jacobs, Hierarchical Mixtures of Experts and the EM algorithm, Neural Computation, Vol. 6, No. 2, 1994, pp. 181-214.

51. G. Kim and S. Kim, Feature Selection Using Genetic Algorithm for Handwritten Character Recognition, Proceedings of 7th International Workshop on Frontiers of Handwriting Recognition, Amsterdam, Netherlands, 2000, pp. 103-110.

52. N. G. Kingsbury, Image Processing with Complex Wavelets, Phil. Trans. R. Soc. Lond, A 357, 1999, pp. 2543-2560.

53. J. Kittler, J. Hatef, R. P. Duin, and J. Matas, On Combining Classifier, IEEE Transactions on PAMI, Vol. 20, No. 3, March 1998, pp. 226-239.

54. U. Krebel, Pairwise Classification and Support Vector Machines, Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 255-268.

55. A. Krzyzak, W. Dai and C. Y. Suen, Unconstrained Handwritten Character Recognition Using Modified Backpropagation Model, Proceedings of International Workshop Frontiers in Handwritten Recognition, April 1990, pp. 145-153.

56. M. Kudo and J. Sklansky, Comparison of Algorithms That Select Features for Pattern Classifiers, Pattern Recognition, Vol. 33, No. 1, January 2000, pp. 25-44.

57. L. I. Kuncheva and L. C. Jain, Designing Classifier Fusion System by Genetic Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 4, No. 4, Sept. 2002, pp. 327-336.

58. L. I. Kuncheva, A Theoretical Study on Six Classifier Fusion Strategies, IEEE Transactions on PAMI, Vol. 24, No. 2, Feb. 2002, pp. 281-286.

59. L. I. Kuncheva, Switching Between Selection and Fusion in Combining Classifiers: An Experiment, IEEE Transactions on System, Man, Cybernetics, Part-B, Vol. 32, No. 2, April 2002, pp. 146-156.

60. L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, Decision Templates for Multiple Classifier Fusion: An Experimental Comparison, Pattern Recognition, Vol. 34, No. 2, 2001, pp. 299-314.

61. E. Kussul and T. Baidyk, Improved Method of Handwritten Digit Recognition Tested on MNIST Database, Image and Vision Computing, Vol. 22, No. 12, 2004, pp. 971-981.

62. E. Kussul, T. Baidyk, and D. C. Wunsch II, Image Recognition Systems with Permutative Coding, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, August 2005, pp. 1788-1793.

63. L. Lam, Q. Xu, and C. Y. Suen, Differentiation Between Alphabetic and Numeric Data Using NN Ensembles, Proceedings of 16th International Conference on Pattern Recognition, Quebec City, Aug. 2002, pp. 40-43.

64. L. Lam and C. Y. Suen, Optimal Combinations of Pattern Classifiers, Pattern Recognition Letters, Vol. 16, No. 9, Sept. 1995, pp. 945-954.

65. F. Lauer, G. Bloch, and C. Y. Suen, Increasing the Recognition Rate of Handwritten Digit Classifiers, Technical Report, CENPARMI, Concordia University, 2005.

66. C. H. Lee and D. A. Langdgrebe, Feature Extraction Based on Decision Boundaries, TEEE Transaction on PAMI, Vol. 15, No. 4, April 1993, pp. 388-400.

67. S. W. Lee, Off-Line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network, IEEE Transactions on PAMI, Vol. 18, No. 6, June 1996, pp. 648-652.

68. S. W. Lee, C. H. Kim, H. Ma, and Y. Y. Tang, Multiresolution Recognition of Unconstrained Handwritten Numerals with Wavelet Transform and Multilayer Cluster Neural Networks, Pattern Recognition, Vol. 29, No. 12, 1996, pp. 1953-1961.

69. C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, Handwritten Digit Recognition Using State-of-the-art Techniques, Proceedings of the 8th International Workshop on Frontiers in Handwritten Recognition, Ontario, Canada, August 2002, pp.320-325.

70. C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques, Pattern Recognition, Vol. 36, No. 10, 2003, pp. 2271-2285.

71. C. L. Liu, H. W. Hao, and H. Sako, Confidence Transformation for Combining Classifiers, Pattern Analysis and Applications, Vol. 7, No. 1, 2004, pp. 2-17.

72. D. Lokenath, Wavelet Transforms and Their Applications, Boston: Birkhauser, 2002.

73. S. Mallat, A Wavelet Tour of Signal Processing, Second Edition, Academic Press, 1999.

74. S. G. Mallat, A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Transactions on PAMI, Vol. 11, No. 7, 1989, pp. 674-693.

75. G. Mayraz and G. E. Hinton, Recognizing Handwritten Digits Using Hierarchical Products of Experts, IEEE Transactions on PAMI, Vol. 24, No. 2, Feb. 2002, pp. 189-197.

76. P. Mitra, C. A. Muthy, and S. K. Pal, Unsupervised Feature Selection Using Feature Similarity, IEEE Transactions on PAMI, Vol. 24, No. 3., March 2002, pp. 301-312.

77. B. T. Mittchell and A. M. Gillies, A Model-Based Computer Vision System for Recognizing Handwritten ZIP Codes, Machine Vision and Applications, Vol. 21, No. 4, 1989, pp.231-243.

78. U. Naftaly, N. Intrator, and D. Horn, Optimal Ensemble Averaging of Neural Networks, Network: Computation in Neural Systems, Vol. 8, No. 3, 1997, pp. 283-296.

79. G. Nagy, State of the Art in Pattern Recognition, Proceedings of the IEEE, Vol. 56 No. 5, 1968, pp. 536-562.

80. P. M. Narendra and K. Fukunaga, A Branch and Bound Algorithm for Feature Subset Selection, IEEE Transaction on Computers, Vol. C-26, No. 9, Sept. 1997, pp. 917-922.

81. C. M. Nunes, A. de S. Britto, C. A. A. Kaestner, and R. Sabourin, An Optimized Hill Climbing Algorithm for Feature Subset Selection: Evaluation on Handwritten Character Recognition, Proceedings of Ninth International Workshop on Frontiers in Handwriting Recognition, October 2004, Tokyo, Japan, pp. 365-370.

82. II-Seok Oh and C. Y. Suen, Distance Features for Neural Network-Based Recognition of Handwritten Characters, International Journal on Document Analysis and Recognition, Vol. 1, No. 1, 1998, pp. 73-88.

83. II-Seok Oh, J. S. Lee, and C. Y. Suen, Analysis of Class Separation and Combination of Class-Dependent Features for Handwriting Recognition, IEEE Transaction on PAMI, Vol. 21, No. 10, Oct. 1999, pp. 1089-1094.

84. II-Seok Oh, J. S. Lee, and B. R. Moon, Hybrid Genetic Algorithms for Feature Selection, IEEE Transaction on PAMI, Vol. 26, No. 11, November 2004, pp. 1424-1437.

85. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, Impacts of Verification on A Numeral String Recognition System, Pattern Recognition Letters, Vol. 24, No. 7, July 2002, pp. 1023-1031.

86. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 17, No. 6, 2003, pp. 903-929.

87. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, Automatic Recognition of Handwritten Numeral Strings: A Recognition and Verification Strategy, IEEE Transactions on PAMI, Vol. 24, No. 11, November 2002, pp. 1438-1455.

88. H, S. Park and S. W. Lee, A Truly 2-D Hidden Markov Model for Off-Line Handwritten Character Recognition, Pattern Recognition, Vol. 31, No. 12, 1998, pp. 1849-1864.

89. J. C. Platt, Fast Training of Support Vector Machine Using Sequential Minimal Optimization, Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185-208.

90. L. Prasad, Wavelet Analysis with Applications to Image Processing, Boca Raton: CRC Press, 1997.

91. P. Pudil, J. Novovicova, and J. Kittler, Floating Search Methods in Feature Selection, Pattern Recognition Letters, Vol. 15, No. 11, Nov. 1994, pp. 119-1125.

92. J. R. Quinlan, C4.5: Programming for Machine Learning, Morgan Kaufmann, San Mateo, Caligornia, 1993.

93. L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition, Proceedings of the IEEE, Vol. 77, No. 2, 1989, pp. 257-286.

94. A. F. R. Rahman and M. C. Fairhurst, an Evaluation of Multi-Expert Configuration for the Recognition of Handwritten Numerals, Pattern Recognition, Vol. 31, No. 9, 1998, pp. 1255-1273.

95. A. F. R. Rahman, W. G. J. Howells, and M. C. Fairhurst, A Multi-Expert Framework for Character Recognition: A Novel Application of Clifford Networks, IEEE Transactions on Neural Networks, Vol. 12, No. 1, Jan. 2001, pp. 101-112.

96. M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, Dimensionality Reduction Using Genetic Algorithms, IEEE Trans. Evolutionary Computation, Vol. 4, No. 2, July 2000, pp. 164-171.

97. M. D. Richard and R. P. Lippmann, Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities, Neural Computation, Vol. 3, No. 4, 1991, pp. 461-483.

98. C. Rodriguez, I. Soraluze, J. Muguerza, J. I. Martin, and G. Alvarez, Hierarchical Classifiers Based on Neighborhood Criteria with Adaptive Computational Cost, Pattern Recognition, Vol. 35, No. 12, Dec. 2002, pp. 2761-2769.

99. R. Schettini, C. Brambilla, G. Ciocca, A. Valsasna, and M. De Ponit, A Hierarchical Classification Strategy for Digital Documents, Pattern Recognition, Vol. 35, No. 8, 2002, pp. 1759-1769.

100. J. Schurmann, Pattern Classification - A Unified View of Statistical and Neural Approaches, Wiley-Interscience, 1996.

101. C. Shapiro, A Hierarchical Multiple Classifier Learning Algorithm, Pattern Analysis and Applications, Vol. 6, No. 3, 2003, pp.285-300.

102. M. Shi, Y. Fujisawa, T. Wakbayashi, and F. Kimura, Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image, Patter Recognition, Vol. 35, No. 10, 2002, pp. 2051-2059.

103. K. J. Siddiqui, Y. H. Liu, D. R. Hay, and C. Y. Suen, Feature Selection Using a Proximity-Index Optimization Model, Pattern Recognition Letters, Vol. 15, No. 11, Nov. 1994, pp. 1137-1141.

104. W. Siedlecki and J. Sklansky, A Note on Genetic Algorithm for Large-Scale Feature Selection, Pattern Recognition Letters, Vol. 10, No. 5, November 1989, pp. 335-347.

105. P. Y. Simard, D. Steinkraus, and J. C. Platt, Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, Proceedings of 7th International Conference on Document Analysis and Recognition, 2003, pp. 958-962.

106. L. Stringa, A New Set of Constraint-Free Character Recognition Grammars, IEEE Transactions on PAMI, Vol. 12, No. 12, 1990, pp. 1210-1217.

107. L. Stringa, Efficient Classification of Totally Unconstrained Handwritten Numerals with a Trainable Multilayer Network, Pattern Recognition Letters, Vol. 10, No. 10, 1989, pp. 273-280.

108. C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, Computer Recognition of Unconstrained Handwritten Numerals, Proceedings of the IEEE, Vol. 80, No. 70, July 1992, pp.1162-1180.

109. Y. Tao, R. C. M. Lam, and Y. Y. Tang, Feature Extraction Using Wavelet and Fractal, Pattern Recognition Letters Vol. 22, No. 1, 2001, pp. 271-287.

110. L.N. Teow and K. F. Loe, Robust Vision-Based Feature and Classification Schemes for Off-Line Handwritten Digit Recognition, Pattern Recognition, Vol. 35, No.1, 2002, pp. 2355-2364.

111. A. Teredesai and V. Govindaraju, GP-Based Secondary Classifiers, Pattern Recognition, Vol. 38, No. 4, 2005, pp. 505-512.

112. K. Torkkola, Feature Extraction by Non-Parametric Mutual Information Maximization, Journal of Machine Learning Research, Vol. 3, 2003, pp.1415-1438.

113. Ø. D. Trier, A. K. Jain, and T. Taxt, Feature Extraction Methods for Character Recognition-A Survey, Pattern Recognition, Vol. 29, No. 4, pp. 641-662, 1996.

114. W. E. Weideman, M. T. Manry, H. C. Yau, and W. Gong, Comparisons of a Neural Network and a Nearest-Neighbor Classifier Via the Numeric Handprint Recognition

Problem, IEEE Transactions on Neural Networks, Vol. 6, No. 6, November, 1995, pp. 1524-1531.

115. K. Woods, W. P. Kegelmeyer. Jr., and K. Bowyer, Combination of Multiple Classifier Using Local Accuracy Estimates, IEEE Transactions on PAMI, Vol. 19, No. 4, April 1997, pp. 405-410.

116. L. Xu, A. Krzyzak, and C. Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwritten Recognition, IEEE Transactions on Systems, Man, Cybernetics, Vol. 22, No. 3, 1992, pp. 418-435.

117. L. H. Yang, C. Y. Suen, T. D. Bui, and P. Zhang, Discrimination of Similar Handwritten Numerals Based on Invariant Curvature Features, Pattern Recognition, Vol. 38, No. 7, July 2005, pp. 947-963.

118. B. Zhang, M. Fu, H. Yan, and M. A. Jabri, Handwritten Digit Recognition by Adaptive-Subspace Self-Organizing Map, IEEE Transactions on Neural Networks, Vol. 10, No. 4, 1999, pp. 939-953.

119. P. Zhang, T. D. Bui, and C. Y. Suen, Nonlinear Feature Dimensionality Reduction for Handwritten Numeral Verification, Pattern Analysis and Applications, Vol. 7, No. 3, Dec. 2004, pp. 296-307.

120. P. Zhang, T. D. Bui, and C. Y. Suen, Wavelet Feature Extraction for the Recognition and Verification of Handwritten Numerals, Keynote Address in the Book: Wavelet Analysis and Active Media Technology, World Scientific Publisher, 2005.

121. P. Zhang, T. D. Bui, and C. Y. Suen, Hybrid Feature Extraction and Forest Feature Selection for Increasing Recognition Accuracy of Handwritten Numerals, Proceedings of 8th International Conference on Document Analysis and Recognition, Seoul, S. Korea, Aug. 29-Sept. 1, 2005.

122. P. Zhang, T. D. Bui, and C. Y. Suen, Extraction of Hybrid Complex Wavelet Features for the Verification of Handwritten Numerals, Proceedings of International

Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan, Oct. 2004, pp. 347-350.

123. P. Zhang, T. D. Bui, and C. Y. Suen, Multi-Modal Nonlinear Feature Reduction for the Recognition of Handwritten Numerals, Proceedings of First Canadian Conference on Computer and Robot Vision (CRV'04), London, Ontario, Canada, May 2004, pp. 393-400.

124. P. Zhang, T. D. Bui, and C. Y. Suen, Recognition of Similar Objects Using 2-D Wavelet-Fractal Feature Extraction, Proceedings of 16th International Conference on Pattern Recognition, Vol. 2, Quebec, Canada, Aug. 2002, pp. 316-319.

125. J. Zhou, A. Krzyzak, and C. Y. Suen, Verification - A Method of Enhancing the Recognizers of Isolated and Touching Handwritten Numerals, Pattern Recognition, Vol. 35, No. 5, 2002, pp. 1179-1189.

126. M. Zimmermann, R. Bertolami, and H. Bunke, Rejection Strategies for Offline Handwritten Sentence Recognition, Proceedings of 17th International Conference on Pattern Recognition, Cambridge, England, 2004, pp. 550-553.