

ERROR ANALYSIS OF A HYBRID MULTIPLE CLASSIFIER SYSTEM FOR
RECOGNIZING UNCONSTRAINED HANDWRITTEN NUMERALS

Chun Lei He

A Thesis
In
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Master Science
(Computer Science and Software Engineering) at
Concordia University
Montreal, Quebec, Canada

July 2005

©Chun Lei He, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 0-494-10287-X

Our file *Notre référence*

ISBN: 0-494-10287-X

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

ABSTRACT

ERROR ANALYSIS OF A HYBRID MULTIPLE CLASSIFIER SYSTEM
FOR RECOGNIZING UNCONSTRAINED HANDWRITTEN NUMERALS

Chun Lei He

Since the early 1990s, many research communities, amongst the pattern recognition and machine learning, have shown a growing interest in Multiple Classifier Systems (MCSs), particularly for the recognition of handwritten words and numerals.

This thesis is divided into two parts. First, we construct an effective hybrid MCS (HMCS) of handwritten numeral recognition in order to raise the reliability of the entire system. This HMCS is proposed by integrating the cooperation (serial topology) and combination (parallel topology) of three classifiers: SVM, MQDF, and LeNet-5. In cooperation, patterns rejected from the previous classifier become the input of the next classifier. Based on the principles of different classifiers, effective measurements for the rejection options – First Rank Measurement (FRM), Differential Measurement (DM), and Probability Measurement (PM) are defined. In combination, Weighted Borda Count (WBC) at the rank level, which reflects *confidence* and *preference* of different ranks in different classes with different classifiers, is applied. Second, we analyze factors that cause the errors in HMCS. In this process, we focus mainly on the role of size normalization on the recognition of handwritten numerals. We have conducted experiments to investigate its effects and have found that the performance of handwritten numeral recognition systems deteriorates dramatically as the size resolution lowers.

The experiment was conducted on the MNIST database, which is a widely known handwritten digit recognition benchmark. The MNIST database of handwritten digits has a training set of 60,000 samples, and a test set of 10,000 samples. The final recognition rate of this system ranges from 95.54% to 99.11%, with a reliability of 99.93% to 99.11%. Hence, we conclude that, comparing to other systems, the proposed system has successfully achieved a high reliability while maintaining a reasonable recognition rate.

successfully achieved a high reliability while maintaining a reasonable recognition rate. For the MNIST dataset, this study shows that enlarging the size from $20 * 20$ to $26 * 26$ can improve the performance significantly. After constructing a smaller database of difficult original patterns from NIST, we prove that normalizing the original data to a size larger than $20 * 20$ in MNIST increases the recognition rate further.

Acknowledgements

First and foremost, I would like to take this opportunity to express my sincerest thanks to my supervisor, Dr. Ching Y. Suen, Founder and Director of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI). I feel extremely fortunate to have been guided by him because he has always encouraged me to do my very best. He has been a great teacher and role model.

I would also like to thank all my friends, especially those at CENPARMI. I enjoyed the monthly meetings of the Handwritten Numeral Recognition Group: Dr. Richard Zannibi, Ping Zhang, Javad Sadri, and Sapargali Kamar. I learnt a lot from the stimulating discussions and their helpful suggestions. I also appreciate the assistance I received from Ms. Shira Katz, who helped me to correct the grammar in my thesis and to enhance my English writing skills. Moreover, I would like to thank my friends, Dr. Qizhi Xu, Dr. Jianxiong Dong, Jinna Tan, Yan Zhang, Guiling Guo, Wumo Pan, Wu Ding, Yun Li, Ning Wang, and Nicola Nobile, who shared the good times with me in the past two years; their friendship has made this experience a memorable one.

Finally, I would like to thank my husband, Hao Meng, my parents, my parents-in-law, and my sister, Chunwei He, for being with me, loving me, and believing in me. Without their continuous support and encouragement, I would not have been able to complete my work.

CONTENTS

LIST OF FIGURES	IX
LIST OF TABLES	X
1. INTRODUCTION.....	1
1.1 Motivation and Objectives.....	1
1.2 State of the Art.....	4
1.3 Outline of Thesis.....	10
2. CLASSIFIERS	13
2.1 Classification Methods.....	13
2.2 SVM.....	16
2.3 LeNet-5	17
2.4 MQDF`	21
3. REJECTION MEASUREMENTS FOR COOPERATION	26
3.1 Rejection Option	26
3.2 First Rank Measurement (FRM).....	27
3.3 Differential Measurement (DM).....	29
3.4 Probability Measurement (PM).....	30
4. HYBRID MULTIPLE CLASSIFIER SYSTEM (HMCS).....	32
4.1 Integration Methods.....	32
4.2.1 Cooperation.....	32
4.2.2 Combination.....	34
4.2 HMCS in this paper	43

5. ERROR ANALYSIS	45
5.1 Introduction.....	45
5.2 Pre-processing & Feature extraction.....	46
5.2.1 Pre-processing.....	47
5.2.2 Feature extraction.....	49
5.3 Recognizing images in MNIST with different sizes.....	51
5.4 Finding the originals to construct a small database	52
5.5 Comparing the substitution rates of the small database.....	59
6. EXPERIMENTAL RESULTS	60
6.1 Database.....	60
6.2 Experimental Results of the Entire System	61
6.2.1 Experimental Results of the Cooperation.....	62
6.2.2 Experimental Results of the Combination.....	67
6.2.3 Experimental Results of the HMCS	68
6.3 Experimental Results of Error Analysis	69
6.3.1 Error Rates at Various Sizes.....	69
6.3.2 Experimental Results of NIST.....	71
6.3.3 Experimental Results of the Small Database.....	73
7. CONCLUSION	75
7.1 Summary	75
7.2 Future Research	77
REFERENCES.....	80
APPENDICES.....	85
APPENDIX I:.....	86
APPENDIX II:.....	87
APPENDIX III:.....	88
APPENDIX IV:	93

APPENDIX V:95

List of Figures

Figure 1. System of classifiers carrying out the cooperation of classifiers.....	5
Figure 2. System of classifiers carrying out the combination of classifiers.....	6
Figure 3. System of classifiers carrying out the selection of classifiers.....	6
Figure 4. Example of the output information in three levels.....	8
Figure 5: Architecture of LeNet-5.....	18
Figure 6. Flowchart of cooperation among three classifiers.....	34
Figure 7. Example pattern in MNIST (label = 7).....	36
Figure 8. Flowchart of combination among three classifiers.....	42
Figure 9. A Hybrid system of multiple classifiers.....	43
Figure 10. Sample images in size normalization.....	48
Figure 11. An example of bilinear interpolation.....	48
Figure 13. An example that candidate images are considered.....	58
Figure 14. An example where the aspect ratios are considered.....	59
Figure 15. Samples of MNIST.....	61
Figure 16. Distributions of recognition results of classifiers SVM using FRM.....	63
Figure 17. Distributions of recognition results of classifiers SVM using DM.....	63
Figure 18: Distributions of recognition results of classifier MQDF using DM.....	65
Figure 19: Data distributions of LeNet-5 using DM.....	65
Figure 20: Distributions for LeNet-5 using PM.....	66
Figure 21. Distribution of patterns incorrectly recognized in SVM in LeNet-5 with PM.....	67
Figure 22. Substitution rates at different normalization sizes of MNIST in SVM.....	70
Figure 23. Substitution rates at different normalization sizes of MNIST in MQDF.....	70
Figure 24. Number of errors in the small normalized database from different sources.....	74

List of Tables

<i>Table 1: Connection matrix of C_3 feature maps and S_2 feature maps</i>	19
<i>Table 2: Example of outputs from three classifiers</i>	36
<i>Table 3: Scores of an example pattern in BC</i>	38
<i>Table 4: Confusion matrix of three classifiers</i>	39
<i>Table 5: Scores of an example pattern in WBC</i>	41
<i>Table 6: Numbers of error images with different sizes</i>	52
<i>Table 7: Distribution of samples of each class in small database</i>	52
<i>Table 8: Thresholds for rejection of SVM based on the confidence value of FRM</i>	64
<i>Table 9: Thresholds for rejection of SVM based on the confidence value of DM</i>	64
<i>Table 10: Recognition Results of WBC, BC, and Majority vote</i>	68
<i>Table 11: Comparison of the performance of HMCS with individual classifier</i>	69
<i>Table 12: Numbers of each class distributed in NIST SD 19</i>	72
<i>Table 13: Summary of the width and height of the samples in the Training and Test sets of NIST SD 19</i>	73
<i>Table 14: Substitution numbers of a smaller database with different normalization sources</i>	74

Chapter 1

Introduction

In this chapter, the motivation and objective of this thesis are introduced. We also review the state of the art in Optical Character Recognition (OCR), classifiers, Multiple Classifier Systems (MCSs), and even other disciplines using MCS. As prior knowledge, the outputs of classifiers in three levels and common combination rules are introduced respectively. Finally, I introduce the structure of this thesis in this chapter.

1.1 Motivation and Objectives

In this thesis, we focus on proposing a hybrid MCS (HMCS), effectively integrating the (1) cooperation and (2) combination of multiple classifiers based on ranks from the outputs in order to achieve a high reliability while maintaining a reasonable recognition rate. We also analyze factors of causing the errors in HMCS. In this process, we mainly focus on the role of size normalization in the recognition of handwritten numerals.

In the early days of pattern recognition, a lot of research was focused on printed and handwritten character recognition. Characters were easy to work with, and were therefore regarded as a recognition problem that could be solved easily. However, when the research advanced from printed to handwritten character recognition, a great deal of challenge in solving this problem became apparent because of the variety of unconstrained handwritten numerals, ambiguity of handwritten numerals, different writing styles, different kinds of noise that may break the strokes in the characters or change their topology, and so on. Even though the problem is intrinsically complicated, many researchers continue developing and implementing algorithms for recognizing unconstrained handwritten characters, including numerals. The researchers now expect that an OCR machine will achieve a high recognition rate and a low or even zero substitution rate.

There are a number of classification algorithms to be applied in handwritten character recognition. These algorithms are based on different theories and methodologies [6]. Broadly speaking, we now have two large groups of classification methods, namely, feature-vector based methods and syntactic-and-structural methods. Furthermore, each group of methods includes many algorithms that are based on a variety of other methodologies, e.g., for the first group, there exist Bayes classifier, k -NN classifier, various distance classifiers and neural network based classifiers, etc.

Usually, for a specific application problem, each of these classifiers could attain a different degree of success, but perhaps none of them is totally perfect, or not as good as expected for practical applications. In China, we always say that “Three cobblers with their wits combined equal Zhuge Liang the master mind.” Thus, there is a need to study

the methodology of integrating the results of a number of different classification algorithms so that a better result could be obtained.

Extensive research has been carried out in the last decade on the use of MCSs for complex classification problems and the potential for performance improvement has been proven. In financial applications, errors are less tolerable than rejections since much extra effort is required to detect and correct the errors; therefore, very high reliability is desired. Hence, an HMCS with a high reliability while maintaining a reasonable recognition rate is proposed in this thesis.

In order to reduce misrecognition, we mainly investigate size's effects on the performance of handwritten numeric recognition systems. Some researchers studied the misclassified data in different numeral databases, such as MNIST, CENARMI Database, USPS, and NIST SD 19 [55] to deduce the reasons of misrecognition and probe the probability of avoiding the errors. Suen *et al.* [55] divided the errors into three categories based on their quality and analyze their distributions according to category. Category 1 is for the images that are easily confused with other numerals because of the similarity of their primitives and structures. Category 2 is for the images that humans have difficulty in identifying them because of noise, filled loop, cursive writing or over-segmentation, etc. Category 3 is for the images that are easily recognized by humans without any ambiguity.

Figuring out the reasons and recognizing errors in Category 3 become a challenging and interesting problem. According to a long-period of observation and experiments, we suspected that low resolution reduces the recognition rates of OCR systems dramatically.

In this thesis, we conduct experiments to investigate in detail its effects and identify the role of size resolution in handwritten numeric recognition systems.

1.2 State of the Art

In this section, we review the state of the art in several aspects: Optical Character Recognition (OCR), classifiers, MCS, and even other disciplines using MCS. Moreover, as prior knowledge, the outputs of classifiers in three levels and common combination rules are introduced respectively.

OCR is one of the most successful applications of automatic pattern recognition. OCR has been under research investigation since the mid 1950's. Since then, there has been steady research efforts in OCR devoted to the automatic processing and recognition of handwritten characters, such as letters and numerals. For instance, recognition of unconstrained isolated handwritten numerals is an important aspect of OCR. It has applications in numerous fields including automatic postal sorting, automatic bank cheque, financial slip processing, and so on [4][17][18][21].

A variety of multiple classifier systems have been studied since the late 1950's. For example, a head-demo (the combiner) would select the demon that "shouted the loudest", a scheme that is nowadays called a "winner-take-all" solution [20]. This area became a hot topic in the 90's with significant theoretical advances as well as numerous successful practical applications. Since the early 1990s, Multiple Classifier Systems (MCS), particularly for the recognition of words and handwritten digits [4-6], have been studied frequently to achieve higher accuracy and reliability.

Theoretically, MCS for character recognition is based on the idea that classifiers with different methodologies or different features are often complementary to each other; hence, the combination of different and complementary classifiers may reduce errors considerably and achieve a higher performance accuracy, just as the decision of a panel of human experts is usually superior to that of a single individual [38].

Gunes et al. [10] distinguish the Multiple Classifier Systems by using three categories, including (a) cooperation of classifiers (serial topology) (Figure 1), (b) combination of classifiers (parallel topology) (Figure 2), and (c) selection of classifiers (Figure 3), according to the types of operation among the classifiers. Moreover, if a system operates with several types of associations, the system is known as a hybrid or mixed system.

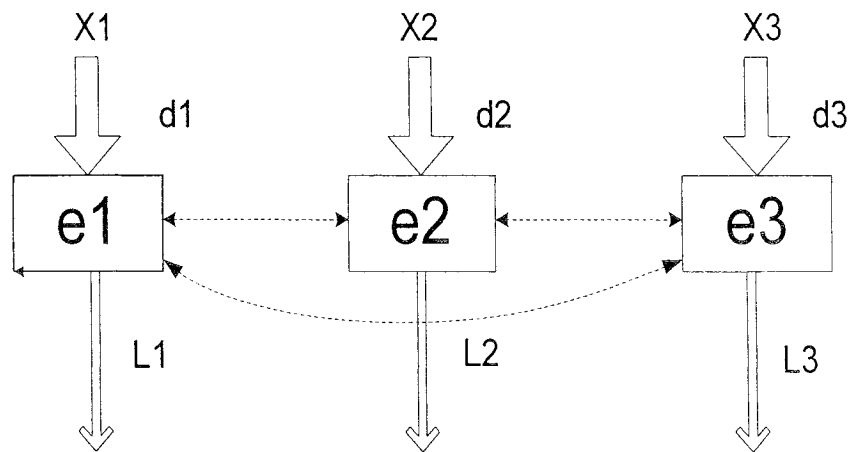


Figure 1. System of classifiers carrying out the cooperation of classifiers

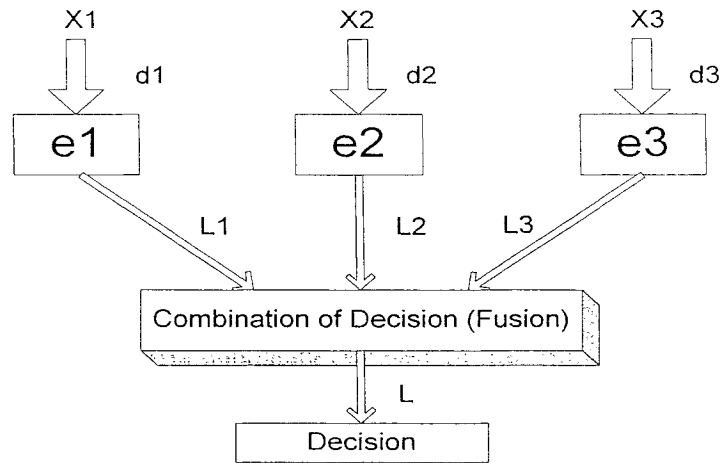


Figure 2. System of classifiers carrying out the combination of classifiers

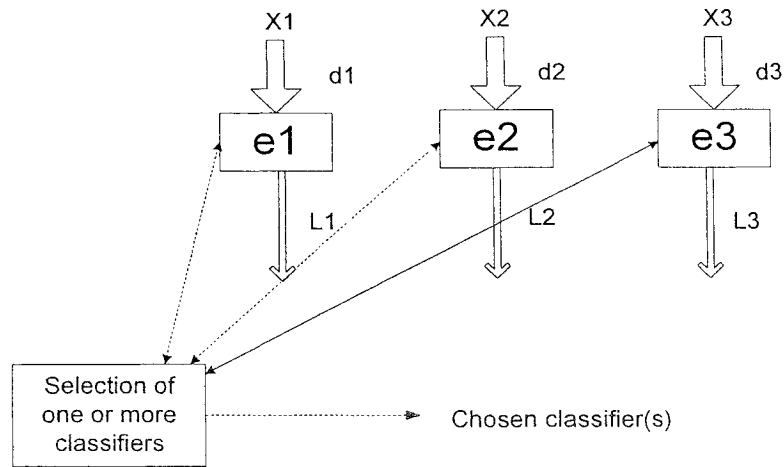


Figure 3. System of classifiers carrying out the selection of classifiers

Till now, many combination techniques have been proposed [6]; essentially, they depend on the information provided by the classifiers. Some researchers use combination rules based on the voting principle [7], others use rules based on the Bayesian theory [6], on belief functions and Dempster-shafer theory of evidence [6], on fuzzy rules [8], on Behaviour Knowledge Space [9], and so on.

Problems similar to MCS have been studied in other disciplines. For instance, in statistics, the idea of *popular opinion* or *consensus* was a means of social and economic organization for many years. The problem of reaching a consensus arises when a group of

people try to reach a common decision from individual opinions. Although the problems may be slightly different in forms, their essence is similar. The consensus problem is similar to MCS because it considers the general problem of combining multiple distributions into a single distribution [30 - 32].

Stone [36] called his model an “opinion pool” in which a consensus was formed through the weighted combination of individual opinions. When the weights were equal, a democratic scheme, or what we call the “voting principle”, was formed. He was interested in the condition under which the consensus achieved a higher “utility” function than the worst opinion of the individual person. DeGroot [34] formalized the theory for linear combination in order to reach a consensus. He proposed a method in which each individual expert was assigned a degree of reliability to every other expert. The experts modified their own opinions after hearing the opinions of others. A consensus was reached when experts stop changing their opinions.

However, some combination techniques also work for aggregating point estimations [33-35]. Numerous combination schemes have been published. Winkler [33] experimented with simple combination strategies in football betting. He noticed that the score increased consistently as more and more assessors are involved in the decision combination process. Although the experiment was simple, and it involved a lot of human factors, he identified three major issues. He observed that a strong correlation exists among assessors working independently, that a combination of some sort improved the score over the average of individuals, and that score improved as more and more assessors are considered.

Generally speaking, the output information that various classification algorithms supply or are able to supply can be divided into three levels:

- 1) The abstract level: a classifier e only outputs a unique label j , or for some extension, e outputs a subset $J \subset \Lambda$.
- 2) The rank level: e ranks all the labels in Λ or (a subset $J \subset \Lambda$) in a queue with the label at the top being the first choice.
- 3) The measurement level: e attributes each label in Λ a measurement value to address the degree that x has the label.

Among the three levels, the measurement level contains the most of information and the abstract level contains the least. From the measurements attributed to each label, we could rank all the labels in Λ according to a rank rule (e.g., ascending or descending order). By choosing the label at the top rank, or directly by choosing the label with the maximal or minimal value at the measurement level, we can assign a unique label to x . In other words, from the measurement level to the abstract level there is an information reduction process or abstraction process.

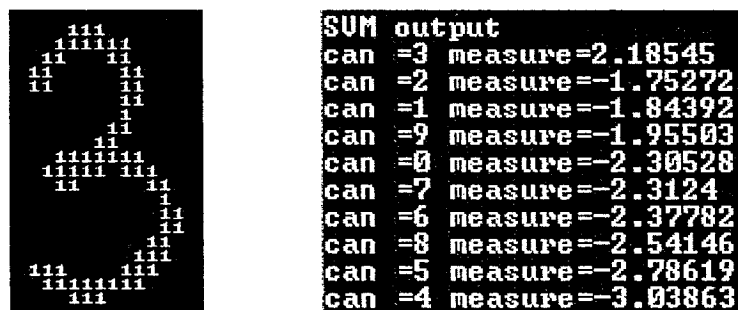


Figure 4. Example of the output information in three levels

Let us see an example. In Figure 4, the output in the abstract lever is $\{3\}$ as 3 is the first rank of the output; the output in the abstract level is $\{3, 2, 1, 9, 0, 7, 6, 8, 5, 4\}$,

which are only ranks of the output; the output in the measurement level is $\{3, 2.18545\}$, $\{2, -1.75272\}$, ..., $\{4, -3.03863\}$, which are ranks with measurements.

For common combination, the following rules are usually used. To generalize, let us define $v_{is}(x_k)$ as a numerical value calculated by the classifier e_s for the class C_i during the classification of a pattern x_k and w_i as a numerical value calculated by the system of classifiers for the class C_i . Usually the most usually employed rules are:

- The Maximum rule: $\forall i \in [1, l], w_i(x_k) = \max_{s=1}^S v_{is}(x_k)$

- The Minimum rule: $\forall i \in [1, l], w_i(x_k) = \min_{s=1}^S v_{is}(x_k)$

- The Sum rule: $\forall i \in [1, l], w_i(x_k) = \sum_{s=1}^S v_{is}(x_k)$

- The Mean rule: $\forall i \in [1, l], w_i(x_k) = \frac{1}{K} \sum_{s=1}^S v_{is}(x_k)$

- The Median rule: $\forall i \in [1, l], w_i(x_k) = \text{median}_{s=1}^S v_{is}(x_k)$

Usually, the decision rule is defined by a function $SC(x_k)$ such as:

$$SC(x_k) = \begin{cases} C_j, & \text{if } w_j(x_k) = \max_{i=1}^l w_i(x_k) \text{ and } w_j(x_k) \geq T \\ C_{l+1}, & \text{else.} \end{cases}$$

In order to reduce misrecognition, we mainly investigate multi-resolution on the performance of handwritten numeral recognition systems. Although some researchers designed some features based on multi-resolution, the role of resolution on the performance of handwritten numeral recognition systems is unclear. Formerly, some studies based on multi-resolution features were presented [57] [25]. In [57], the authors focus on sub-images with multi-resolution. As an example mentioned in this paper, if ‘3’

are the top contenders at a particular recursive stage, the features from the upper zone of the test pattern holds greater discriminatory power and should be examined at a finer resolution. In [58], the authors proposed a scheme in the feature extraction stage, which extracts multi-resolution features with wavelet transform. They only find a fine resolution for sub-image or extract multi-resolution features. These features are based on similarities among classes or dependency among features. In this thesis, the role of resolution on the performance of entire system is studied in depth.

1.3 Outline of Thesis

The thesis is organized into seven chapters.

- In Chapter 1, I introduce the motivation and objective of this thesis. I also review the state of the art in Optical Character Recognition (OCR), classifiers, MCS, and even other disciplines using MCS. As prior knowledge, the outputs of classifiers in three levels and common combination rules are introduced respectively. Moreover, I introduce the structure of this thesis in this chapter.
- In Chapter 2, three classifiers are introduced. These include Support Vector Machine (SVM), Modified Quadratic Discriminant Function (MQDF), and LeNet-5. As these three classifiers have excellent generalization performance in a wide variety of learning applications such as handwritten digit recognition and object recognition, they were chosen to design the HMCS.
- In Chapter 3, we define three effective rejection measurements. They are First Rank Measurement (FRM), Differential Measurement (DM), and Probability

Measurement (PM). These measurements are classifiers used in cooperation of HMCS.

- In Chapter 4, a hybrid Multiple Classifier System (HMCS), including cooperation and combination of classifiers, is detailed. In combination, compared to Majority Vote and Borda Count (BC), we propose a more effective combination method at the rank level — Weighted Borda Count (WBC). Afterwards, we describe the entire structure of this HMCS.
- In Chapter 5, we analyze errors and observe the relationship between normalization sizes and recognition rates in this HMCS. We propose to analyze the substitution images with different normalization sizes, but the same classifier and same features extracted to predict the relationship between normalization sizes and recognition rates. After constructing a smaller database of difficult original patterns from NIST, we found that normalizing the original data to a size larger than $20 * 20$ in MNIST further increases the recognition rate.
- In Chapter 6, we introduce the database used in this thesis, and demonstrate the experimental results of HMCS and error analysis, respectively. In HMCS, we show not only each classifier's experimental results of rejection option in cooperation, but also experimental results of different combination methods. Afterwards, we compare the performance of HMCS and each individual classifier. In error analysis, we show the experimental results of recognition rates with different normalization sizes in SVM and MQDF. Moreover, we did some statistics on sizes of patterns in NIST. Finally, we compare the error rates in the

small database with various sizes normalized from patterns in both MNIST and NIST.

- Conclusions are drawn in Chapter 7. In this chapter, we summarize contribution of this thesis and present some future work in this research direction. In this thesis, we not only proposed an effective hybrid Multiple Classifier System but also investigate the relationship between the performance of handwritten numeral recognition systems and size resolution. Thus, we have some future work in these two facets.

There are several appendices in this thesis.

- Appendix I contains index of images in MNIST, which are substituted in different sizes.
- Appendix II contains index of images in NIST and MNIST, which are substituted in at least 2 different sizes.
- Patterns with maximum or minimum width, height or area in NIST SD 19 are demonstrated in Appendix III.
- In Appendix IV, original images of patterns in NIST, which are incorrectly recognized in MNIST with HMCS.
- In Appendix V, Total Probability Theorem is introduced.

Chapter 2

Classifiers

In this chapter, three classifiers are introduced. These include Support Vector Machine (SVM), Modified Quadratic Discriminant Function (MQDF), and LeNet-5. As these three classifiers have excellent generalization performance in a wide variety of learning applications such as handwritten digit recognition and object recognition, they were chosen to design the HMCS.

2.1 Classification Methods

There are a number of classification algorithms to be applied in handwritten character recognition. These algorithms are based on different theories and methodologies. The classifiers include (1) support vector machine (SVM) classifiers, (2) nearest-neighbor classifiers, (3) Bayesian classifiers, (4) polynomial discriminant classifiers, (5) neural network classifiers, (6) tree classifiers, (7) syntactic approaches, (8) Hidden Markov Model (HMM), etc. They typically use feature descriptors in the form of vectors and return a class identity.

- SVM is a large margin linear classifier on a feature space defined by the kernel function. The weight vector is the interpolation of the learning patterns. The coefficients are determined on the learning patterns by solving a quadratic optimization problem, in which a pre-specified upper bound of coefficients controls the tolerance of learning errors. After optimization, only a small portion of the learning patterns, which are called support vectors (SVs), have a non-zero coefficient.
- The nearest-neighbor classifier performs direct prototype matching using a predefined distance to measure the similarity between a pattern and those prototypes in a class. The distance function can be a Euclidean or a Hamming distance. The problem with the method is that there is a high computation cost when classification is conducted [39]. There are many variants of this approach with the intention to reduce the complexity. A famous one is the k-nearest neighbor which finds k closest matches and uses a voting scheme to decide on the class. When pixel value is used directly, the method is referred to as template matching.
- The Bayesian classifier assigns a pattern to a class with the maximum posterior probability. The class prototypes are used in a training stage to estimate the class-conditional probability density function for a feature vector [40 - 41].
- The polynomial discriminant classifier [42] assigns a pattern to a class with the maximum discriminant value, which is computed by a polynomial in the components of a feature vector. The class models are implicitly represented by the coefficients in the polynomial.

- Syntactic classifiers [43] use grammars at all levels in the Chomsky hierarchy to describe class models. These grammars take in a high level descriptor such as a symbol string instead of feature vectors. The class models are abstracted as grammatical rules that can be used to generate the prototypes.
- Tree classifiers are motivated by the need to reduce the complexity in prototype matching. There are many design strategies [44 - 45], but generally it is difficult to control the growing and pruning of trees. Commonly used control methods are mutual information, probability models or entropy values. The most famous tree classifiers are CART [46] and C4.5 [47]. Ho [48] extends the work to C4.5 Decision Forests and reports good results.
- Hidden Markov Model (HMM) [49] is a statistical framework for modeling sequential input by state transitions. It has been widely used in speech recognition and online handwritten recognition. Its applications to offline handwritten recognition have been growing. Cai *et al.* [52] define the state of a given observation in HMM as a micro-state and the collections of individual micro-states as macro-states. The statistical information of a handwritten numeral is represented by micro-states using HMMs, and the structural information is modeled by relationships between macro-states. Park *et al.* [53] use a 2-D HMM for character recognition.

2.2 SVM

In the past several years, support vector machine (SVM) has played an increasingly important role in the pattern recognition system due to its excellent generalization performance in a wide variety of learning applications such as handwritten digit recognition and object recognition.

Given that training sample $\{X_i, y_i\}$, $y_i \in \{-1, 1\}$, $X_i \in R^n$ where y_i is the class label and $i=1, \dots, N$.

The support vector machine first maps the data to a Hilbert space H , which can be considered as a generalization of Euclidean space, using a mapping Φ ,

$$\Phi: R^n \rightarrow H$$

The mapping Φ depends on a kernel function K that satisfies the Mercer's conditions [40, 41] such that $K(X_i, X_j) = \Phi(X_i) \bullet \Phi(X_j)$. Then, in the space H , we need to find an optimal hyperplane by maximizing the margin and bounding the number of training errors. More specifically, we need to compute the sign of $f(X)$, where

$$\begin{aligned} f(X) &= W \Phi(X) + b \\ &= \sum_{i=1}^N \alpha_i y_i \Phi(X_i) \Phi(X) + b \\ &= \sum_{i=1}^N \alpha_i y_i K(X_i, X) + b \end{aligned} \tag{2.1}$$

and b is a threshold. The data X_i for which $\alpha_i > 0$ are called support vectors (SV). We can avoid computing $\Phi(X)$ explicitly and use the kernel function K instead.

Training an SVM is to find $\alpha_i, i = 1, \dots, N$, which can be obtained by minimizing the following quadratic cost function:

$$L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(X_i, X_j) \quad (2.2)$$

subject to $0 \leq \alpha_i \leq C \quad i=1 \dots N$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (2.3)$$

where C is a parameter decided by the user. The larger the value of C , the higher penalty allocated to the training errors.

As the training of SVM is slow, we applied a fast SVM training algorithm [1]. The algorithm applies Keerthi *et al*'s [37] SMO to solve the optimization of a sub-problem in a working set, which is a subset of the training set, in combination with some effective techniques such as kernel caching, “digest” strategy, shrinking strategies, and the queue technique of selecting a new working set.

2.3 LeNet-5

Convolutional Neural Networks [28] are specialized neural network architectures which incorporate knowledge about the invariances of 2D shapes by using local connection patterns, and by imposing constraints on the weights. LeNet-5 is a convolutional neural network.

LeNet-5 comprises of seven layers, not counting the input, all of which contain trainable parameters (weights). The input is a 32 * 32 pixel image. This is significantly

larger than the largest character in the database (at most 20×20 pixels centered in a 28×28 field). It is desirable for potential distinctive features such as stroke end-points or centers to appear in the center of the receptive field of the highest-level feature detectors. In LeNet-5 the set of centers of receptive fields of the last convolutional layer (C3, see below) form a 20×20 area in the center of the 32×32 input. The values of the input pixels are normalized so that the background level (white) corresponds to a value of -0.1 and the foreground (black) corresponds to a value of 1.175. This makes the mean input roughly 0, and the variance roughly 1, which accelerates learning [29].

The seven layers contain convolutional layers (labeled as C_i), sub-sampling layers (labeled as S_i), and fully-connected layers (labeled as F_i). The architecture of the LeNet-5 is shown in Figure 5.

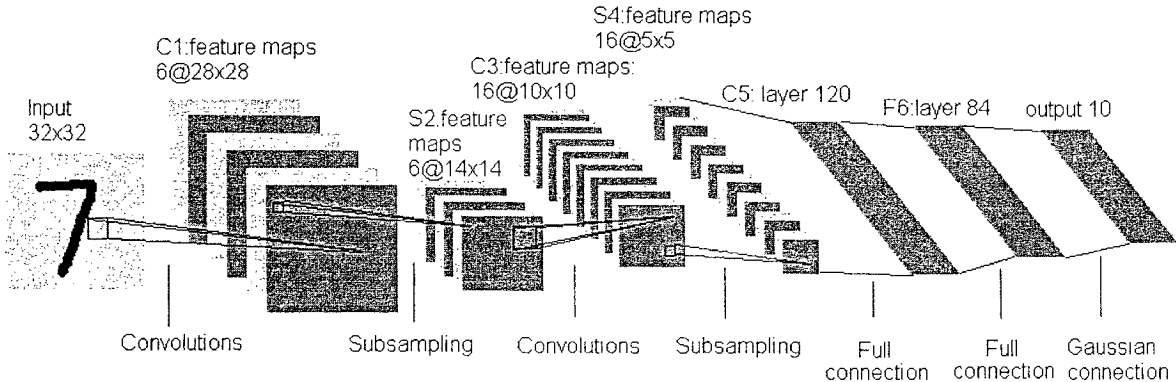


Figure 5: Architecture of LeNet-5

The layer C_1 is a convolutional layer, which is composed of six feature maps of the size of 28×28 with different weight vectors. A unit in a feature map has 25 inputs connected to a 5 by 5 area of the input, which is the receptive field of the unit. The receptive fields of neighboring units are overlapped. The layer C_1 contains 156 trainable parameters and 122,304 connections.

The layer S_2 is a sub-sampling layer with six feature maps of size 14×14 . Each unit is connected to the 2×2 area in the corresponding feature maps of C_1 . The reason for sub-sampling is to reduce the precision of some feature information. Precise information of certain features is harmful because the information is likely to vary in different samples of a numeral. For instance, we only need to know that there is an end point in the top left corner of the numeral seven, and we do not need to know the exact coordination of this end point since the coordination varies in instances of the numeral seven. The value of a unit is obtained by adding the four inputs from C_1 , then multiplying the result by a trainable coefficient, and adding it to a trainable bias. The final result is passed through a sigmoid function. The layer S_2 has 12 trainable parameters and 5,880 connections.

The layer C_3 is a convolutional layer with 16 feature maps of size 10×10 . Each unit is connected to 5×5 neighbors at the identical locations of S_2 's feature maps. The connection matrix of C_3 features maps and S_2 feature maps are shown in Table 1. The layer C_3 has 1,516 trainable parameters and 156,000 connections.

Table 1: Connection matrix of C_3 feature maps and S_2 feature maps

The columns indicate the feature map in C_3 ; the rows indicate the feature maps in S_2 . The table indicates which feature maps in S_2 are combined with a particular feature map of C_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

The layer S_4 is a sub-sampling layer with 16 feature maps of size 5×5 . Each unit in each feature map is connected to the feature maps of the previous layer in a similar way as in C_1 and S_2 . Layer S_4 has 32 trainable parameters and 2,000 connections.

The layer C_5 is a convolutional layer with 120 feature maps of size 1×1 . Each unit is connected to 5×5 neighbors at the identical locations of all of S_4 's feature maps. Layer C_5 contains 48,120 trainable connections.

The layer F_6 is a fully-connected layer with 84 units. It has 10,164 trainable parameters. Up to layer F_6 , the state of unit i , denoted by x_i , is computed by a sigmoid squashing function:

$$x_i = f(a_i) \quad (2.4)$$

where a_i is the weighted sum of the unit i . For each unit, the weighted sum between the input vector and its weight vector is produced by adding a trainable bias to a dot product.

The squashing function is defined as

$$f(a) = A \tanh(Sa) \quad (2.5)$$

where A is the amplitude of the function and S determines its slope at the origin. Here, we set at $A=1.7159$ and $S=\frac{2}{3}$.

Finally, the output layer is composed of Euclidean Radial Basis Function units (RBF), one for each class, with 84 inputs each. The RBF unit is computed using the following formula:

$$y_i = \sum_j (x_j - w_{ij})^2 \quad (2.6)$$

The larger the RBF is, the less fitness is between the input pattern and the model of the class associated with the RBF.

The loss function of this neural network is:

$$E(W) = \frac{1}{P} \sum_{p=1}^P \{y_{D^p}(Z^p, W) + \log(e^{-j} + \sum_i e^{-y_i(Z^p, W)})\} \quad (2.7)$$

where y_{D^p} is the output of the D_p th RBF unit, Z^p is the p -input pattern, and W represents the collection of adjustable parameters in the system.

The first term of loss function is an MSE criterion, which pushes down the penalty of the correct class; the second term plays a “competitive” role, as it pulls up the penalties of the incorrect classes.

2.4 MQDF

In this section, we briefly review the modified Quadratic Discriminant Function (MQDF) proposed by Kimura *et al.* [3]. The QDF is obtained under the assumption of multivariate Gaussian density for each class of numerals. The MQDF aims to improve the computation efficiency and classification performance of QDF via eigenvalue smoothing [50].

The parameters of MQDF are estimated via the maximum likelihood (ML) estimation of covariance matrices followed by a K-L transformation. The MQDF is different from the QDF in that the eigenvalues of minor axes are set to a constant. The motivation behind this is to smoothe the parameters that compensate for the estimation error on finite sample size.

Let us start with the Bayesian decision rule, which classifies the input pattern to the class of a maximum a posteriori (MAP) probability out of M classes. Representing a pattern with a feature vector $x=(x_1, \dots, x_d)^T$, the a posteriori probability is computed by Bayes rule:

$$P(w_i | x) = \frac{P(w_i)p(x | w_i)}{p(x)}, \quad i = 1, \dots, M \quad (2.8)$$

where $P(w_i)$ is the a priori probability of class w_i , $p(x | w_i)$ is the class probability density function (pdf) and $p(x)$ is the mixture density function. Since $p(x)$ is independent of a class label, the nominator of the above formula can be used as the discriminant function for classification:

$$g(x) = P(w_i)p(x | w_i). \quad (2.9)$$

The Bayesian classifier is reduced to Linear Discriminant Function (LDF) or Quadratic Discriminant Function (QDF) under the Gaussian density assumption with varying restrictions. Assume that the pdf of each class is a multivariate Gaussian function:

$$p(x | w_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}{2}\right] \quad (2.10)$$

where μ_i and Σ_i denote the mean vector and the covariance matrix of class ω_i , respectively. Inserting the above formula to $g(x)$, taking the negative logarithm and omitting the common terms under equal a priori probabilities, the QDF is obtained as:

$$g_0^{(i)}(x) = (x - \mu^{(i)})^T \{\Sigma^{(i)}\}^{-1} (x - \mu^{(i)}) + \log |\Sigma^{(i)}| - 2 \log P(\omega^{(i)}). \quad (2.11)$$

for a class $\omega^{(i)}$ where μ_i and Σ_i denote the mean vector and the covariance matrix for Σ_i in the class ω_i , respectively, and Σ_i is the a priori probability for the class ω_i . The

QDF is actually a distance metric in the sense that the class of minimum distance is assigned to the input pattern.

Meanwhile, a QDF can be written in the orthogonal expansion form:

$$g_0(x) = \sum_{i=1}^N \frac{1}{\lambda_i} \{\varphi_i^t(x - \mu_M)\}^2 + \log \prod_{i=1}^n \lambda_i \quad (2.12)$$

by using the equation:

$$\Sigma_M = \sum_{i=1}^N \lambda_i \varphi_i \varphi_i^t \quad (2.13)$$

where the μ_M and Σ_M denote the maximum likelihood estimates of the mean and the covariance, respectively, and λ_i ($\lambda_i \geq \lambda_{i+1}$) and φ_i denote the i th eigenvalue and the eigenvector of the matrix Σ_M respectively.

However, the QDF uses the maximum likelihood estimate of the covariance matrix, which is sensitive to the estimation error of the covariance matrix. Thus, a MQDF employs a kind of a pseudo Bayesian estimate of the covariance matrix. MQDF is less sensitive to the error and requires less computation time and storage while achieving a better performance.

Performance of the discriminant function is improved by using the following pseudo-Bayesian estimate [51] of the covariance:

$$\Sigma_p = \Sigma_M + h^2 I \quad (2.14)$$

where I is the identity matrix and h^2 is a constant.

From the above formula:

$$\varphi_i \Sigma_p \varphi_i^t = \varphi_i \Sigma_M \varphi_i^t + h^2 = \lambda_i + h^2 \quad (2.15)$$

The i th eigenvalue and eigenvector of Σ_p are equal to $\lambda_i + h^2$ and φ_i , respectively. A modified quadratic MQDF1 is given as:

$$g_1(x) = \sum_{i=1}^N \frac{1}{\lambda_i + h^2} \{\varphi_i^t(x - \mu_M)\}^2 + \log \prod_{i=1}^n (\lambda_i + h^2) \quad (2.16)$$

MQDF1 is less sensitive than QDF to the error of the estimate of the covariance matrix, but it requires $O(n^2)$ computation time and storage as QDF does. In order to decrease the computation time and storage, another modification MQDF2 of the discriminant function has been designed. By substituting h^2 for all of the eigenvalues λ_i , $i \geq k+1$ of Σ_M in QDF, we obtain the MQDF2:

$$g_2(x) = \sum_{i=1}^k \left(\frac{1}{\lambda_i}\right) \{\varphi_i^t(x - \mu_M)\}^2 + \sum_{i=k+1}^n \left(\frac{1}{h^2}\right) \{\varphi_i^t(x - \mu_M)\}^2 + \log(h^{2(n-k)} \prod_{i=1}^n \lambda_i) \quad (2.17)$$

By using the equation:

$$\sum_{i=1}^n \{\varphi_i^t(x - \mu_M)\}^2 = \|x - \mu_M\|^2 \quad (2.18)$$

The MQDF2 is rewritten as:

$$g_2(x) = \frac{1}{h^2} [\|x - \mu_M\|^2 - \sum_{i=1}^k \left(1 - \frac{h^2}{\lambda_i}\right) \{\varphi_i^t(x - \mu_M)\}^2] + \log(h^{2(n-k)} \prod_{i=1}^n \lambda_i) \quad (2.19)$$

It is obvious that the required computation time and storage of the MQDF2 are about k/n times those of the QDF and the MQDF1. Also, the MQDF2 as well as the MQDF1 are less sensitive to the estimation error of the covariance matrix if h^2 and k are suitably chosen.

The advantages of the MQDF2 are multiple. First, it overcomes the bias of minor eigenvalues (which are underestimated on small sample sizes), allowing the classification performance to be improved. Second, for computing the MQDF2, only the principal

eigenvectors and eigenvalues are to be stored so that the memory space is reduced. Third, the computation effort decreases because the projections to minor axes are not computed [39].

Chapter 3

Rejection Measurements for Cooperation

In this chapter, we define three effective rejection measurements. They are First Rank Measurement (FRM), Differential Measurement (DM), and Probability Measurement (PM). These measurements are used by classifiers in cooperation of HMCS.

3.1 Rejection Option

The reject option can be very useful in preventing excessive misclassifications in applications that require high classification reliability [11]. Rejection measurements based on the outputs of classifiers are vital, especially in cooperation with a hybrid Multiple Classifier System (HMCS), because only the rejected patterns from a previous classifier are classified by the next one.

We propose three measures: First Rank Measurement (FRM), Differential Measurement (DM), and the Probability Measurement (PM) for classifiers in this HMCS. However, we only use the DM and the PM for three classifiers as the DM is modified on

the FRM and has better performance of isolated handwritten numeral recognition. Therefore, if the outputs of the classifiers are numerical scores that are the values of an arbitrary discriminant (MQDF) or of distances to margins (SVM), then the DM is a more effective method to measure the rejected patterns. However, if the outputs of classifiers are the distances to prototypes (LeNet-5), then the PM is better.

If we consider measuring the outputs of classifiers for the rejection option as a two-class problem, acceptable patterns and rejection classification, and thereby the outputs at the measure level can be considered as features for measurements of rejection option.

3.2 First Rank Measurement (FRM)

The first idea of measuring classifier is to compute the statistics for all the patterns that have the negative confidence values of the first rank, especially for SVM. According to the principle of SVM, a negative confidence value means that the given pattern does not belong to the current class, if we only consider two classes (*is* or *isn't*). Hence, if the confidence values of the patterns' first rank are negative, it means that SVM could not classify them very well. In this case, we need to find other classifiers or another method to classify these patterns.

It is true that the patterns we expect to reject, which are wrong in SVM, have negative confidence values or small positive values of their first ranks. Moreover, the distribution of the confidence value of the first rank in SVM has a Gaussian shape. (Its distribution is shown in Chapter 6.) It seems that we can reject all the patterns if their first ranks are negative or if they have small positive confidence values.

However, FRM has some drawbacks. According to the experiments, we find that most patterns with small positive values or negative confidence values in the first rank are not bad. On the contrary, the number of correctly recognized patterns is greater than number of incorrectly recognized patterns in the same range of the first rank's confidence values (showed in tables in Chapter 6). If we reject patterns based on the confidence values of the first rank, the rejection rate is too large at 18.8%.

In addition, FRM does not distinguish patterns whose confidence values of the first rank in SVM are similar, but the values of second rank are either positive or negative. These two kinds of patterns are not the same. If the confidence values of patterns' first rank are positive, and the values of their second rank are negative, we assume that these patterns are successfully classified by the current classifier, and they belong to the same classes as their first ranks. Nevertheless, if both the confidence values of patterns' first rank and the values of their second rank are positive or negative, we can hardly determine if they belong to the class of the first rank or the class of the second rank.

Even if the first ranks' confidence values are negative, but they are much greater than their second ranks' values, they should belong to good classified patterns because they are closer to the boundaries of the first class than to the boundaries of the second class. Sometimes, the errors – negative values in the first rank in SVM, are made by the calculation of boundaries between classes.

Therefore, we develop a new measurement, which uses distribution and relationship among all the confidence values instead of only the confidence of the first rank, to overcome these drawbacks.

3.3 Differential Measurement (DM)

The performance of the first rank's confidence values is known to determine whether the pattern should be rejected. Nonetheless, using the distribution and the relationship among all the confidence values proves to be more reliable.

Generally, we assume that classes are internally cohesive but isolated from the others. Hence, in each classifier, we expect that the first rank of the output with a higher confidence value is correct. Accordingly, the ratio of the difference between the first rank and the center of other ranks $c(x)$ and the center, $\Phi(x)$, is defined as follows:

$$\Phi(x) = \frac{|v_1(x) - c(x)|}{c(x)} \quad (3.1)$$

where

$$c(x) = \frac{1}{N-2} \sum_{i=2}^{N-1} |v_i(x) - v_{i+1}(x)|, \quad (3.2)$$

and $V(x) = \{v_1(x), v_2(x), \dots, v_N(x)\}$ is the output of the given pattern x , and N is the total number of classes.

Actually, we even do not expect $v_2(x)$ to have a competitive confidence value with $v_1(x)$. Thus, we modify $\Phi(x)$ as:

$$\Phi_1(x) = \frac{|v_1(x) - v_2(x)|}{c(x)} \quad (3.3)$$

Using a classifier implementing linear decision functions to a two-class problem:

$$f(x, \alpha) = \text{sign}(w \cdot x + b), \quad (3.4)$$

the corresponding decision function can then be defined as follows:

$$f(x, \alpha) = \begin{cases} +1, & \text{if } \Phi_1(x) \geq T_D \\ -1, & \text{if } \Phi_1(x) < T_D \end{cases} \quad (3.5)$$

where T_D is a threshold derived from the training data.

We call the above measurement method Differential Measurement (DM). As explained, DM has a better performance than FRM in SVM. If the outputs of the classifiers are numerical scores that are the values of an arbitrary discriminant (MQDF) or that are distances to margins (SVM), DM is a more effective method to measure the rejected patterns. However, DM does not perform very well in LeNet-5 (Details in Chapter 6).

3.4 Probability Measurement (PM)

Because DM cannot perform very well in LeNet-5, we need to define another new measurement, called Probability Measurement (PM) for it. We apply the following formula to the output vector $V(x) = \{v_1(x), v_2(x), \dots, v_N(x)\}$, and we derive $P(x) = \{p_1(x), p_2(x), \dots, p_N(x)\}$:

$$p_k(x) = \frac{1/v_k(x)}{\sum_{i=1}^N 1/v_i(x)} \quad k = 1, 2, \dots, N \quad (3.6)$$

For each pattern, if the output vectors of a classifier are arranged in ascending order, then $p_1(x)$ has the maximum value. The corresponding decision function is then defined as follows:

$$f(x, \alpha) = \begin{cases} +1, & \text{if } p_1(x) \geq T_p \\ -1, & \text{if } p_1(x) < T_p \end{cases} \quad (3.7)$$

where T_p is a threshold obtained from the training data.

In fact, the definition of $p_k(x)$ is not new. Formerly, researchers use $p_k(x)$, which obey the three basic axioms of probability theory, as apparent post-probabilities and put them into (3.8) for combination in measure level.

$$P_E(x \in C_i / x) = \frac{1}{K} \sum_{k=1}^K P_k(x \in C_i / x), \quad i = 1, \dots, N \quad (3.8)$$

In this system, we employ $p_k(x)$ for a rejection measurement for classifiers as the outputs of LeNet-5 are the distances to prototypes. Experimentally, PM is better than both FRM and DM in LeNet-5. In Chapter 6, more experimental results of DM and PM for three different classifiers are well demonstrated.

Chapter 4

Hybrid Multiple Classifier System

(HMCS)

In this chapter, a hybrid Multiple Classifier System (HMCS), including cooperation and combination of classifiers, is detailed. In combination, compared to Majority Vote and Borda Count (BC), we propose a more effective combination method at the rank level — Weighted Borda Count (WBC). Afterwards, we describe the entire structure of this HMCS.

4.1 Integration Methods

4.2.1 Cooperation

Cooperation is a serial architecture (as opposed to a combination or parallel architecture) [10]. This method uses the decisions and/or the results of a classifier applied to

handwritten numeral images for better guidance of whether to use one or more other classifiers. Using this topology, classifiers are applied in succession, with each classifier producing a reduced set of possible classes for each pattern, so that the individual classifiers or experts can become increasingly focused [22].

Choosing an optimal sequence of three classifiers is a problem to solve. The general schema of cooperation in this system is described in Figure 6. As we knew before, we measure SVM and MQDF with DM, LeNet-5 with PM. Since SVM has the highest accuracy, we use SVM as the first one, and secondly, we use MQDF or LeNet-5 later. Therefore, analyzing the distribution of patterns rejected from SVM with MQDF and LeNet-5 is necessary. The process of cooperation in this system is: If SVM does not show acceptable outputs, MQDF (or LeNet-5) is applied to the former patterns; or else we consider the SVM's output as the final results. Similarly, if the confidence value in MQDF (or LeNet-5) is high, we accept it and look at it as a final result; or else, we reject it and forward it to LeNet-5 (or MQDF). Finally, if LeNet-5 does not recognize it very well, we reject it; or else we accept its performance in LeNet-5.

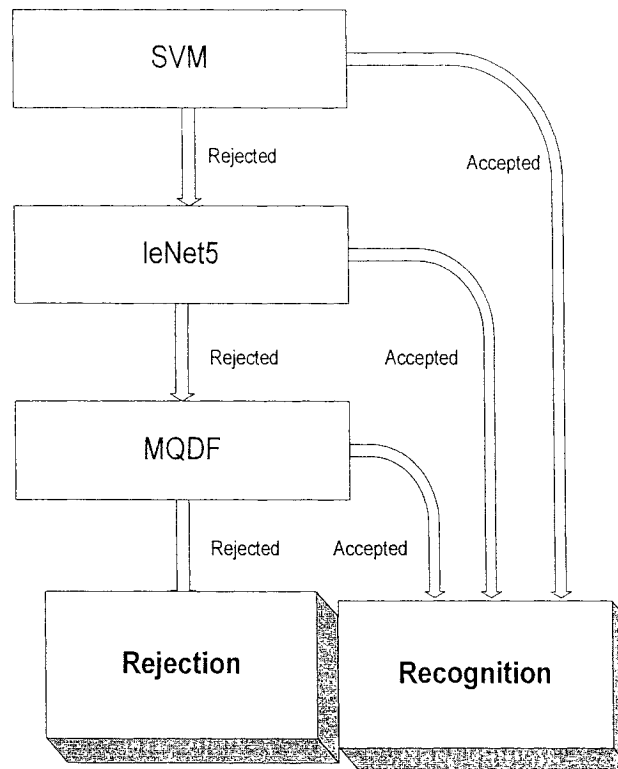


Figure 6. Flowchart of cooperation among three classifiers

Although the reliability is high, the recognition rate is not as high as we expected. Therefore, we design a combination (parallel topology) of three classifiers.

4.2.2 Combination

In combination, as we know, we can combine multiple classifiers in three ways: combination in abstract level; combination in rank level; and combination in measure level.

Classifiers are combined only at the rank level in this research study because it is more general and reasonable. Because voting in abstract level only includes little information from the three classifiers' outputs, the final result of majority vote in abstract level should

not be good enough for patterns' recognition. On the other hand, treating equally all classes generated by all classifiers is not preferable even though we may normalize the final results in measure level. This is because the individual classifier has its own capabilities,

In order to compare the Weighted Borda Count (WBC) algorithm with Majority Vote and Borda Count, we implement a combination in abstract level and other combination algorithms in rank level at first.

4.2.2.1 Majority Vote

About a system in abstract level, majority vote is usually considered because this type of combination can be used for any type of classifier, whatever the type of outputs of these classifiers is.

Majority vote has been a much studied subject among mathematicians and social scientists since its origin in the Condorcet Jury Theorem (CJT) [23]. This theorem provided validity to the belief that the judgment of a group is superior to those of individuals, provided the individuals have reasonable competence [22].

For combinations of small numbers of classifiers by majority vote, another factor that merits attention is the trade-off recognition and error rate. It has been proved theoretically that combinations of even numbers of experts will produce both lower correct and error rates (and higher rejection rates) than combinations of odd numbers of experts.

Majority vote is used in this system as follows: If at least two classifiers agree to the same result, confirm it; or else, reject it. For example, for a pattern x in Figure 7, the outputs of SVM, MQDF, and LeNet-5 are all listed in the Table 2.

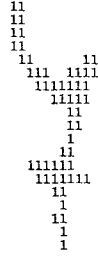


Figure 7. Example pattern in MNIST (label = 7)

In Table 2, as the outputs of SVM, MQDF, and LeNet-5 are $r_1 = 4, r_2 = 9, \text{ and } r_3 = 7$, Let μ be the output of combination, and the final result μ is rejected. Accordingly, only considering the outputs in abstract level is not enough.

Table 2: Example of outputs from three classifiers
[R, M] in the following table represents results and their measurements, respectively

Ranks	SVM	MQDF	LeNet-5
1	[4, -0.519872]	[9,306.227]	[7,36.5627]
2	[8, 0.545886]	[4,309.957]	[8,41.6108]
3	[7,-1015621]	[8,318.193]	[3,50.7307]
4	[3,-1.19321]	[7,348.69]	[6,61.7424]
5	[5,-1.55446]	[5,387.344]	[5,69.9872]
6	[2,-1.6244]	[3,388.129]	[2,73.0493]
7	[6,-1.70928]	[2,446.556]	[0,77.6559]
8	[0,-1.76175]	[1,601.243]	[9,77.8455]
9	[9,-2.04378]	[6,629.95]	[4,80.5389]
10	[1,-2.53281]	[0,653.848]	[1,94.7965]

4.2.2.2 Borda Count (BC)

Let μ be the output of combination. The BC is an equal-weight-voting scheme, where the score $T(\theta)$ of class θ is the (negative) sum of the ranks assigned to it by the constituents:

$$\mu = \text{arg}(\max T(\theta)), \quad (4.1)$$

$$T(\theta) \equiv f_{\theta}(r^{(1)}, \dots, r^{(J)}) = \sum_{j=1}^J -r^{(j)}(\theta) \quad (4.2)$$

This function was proposed by Borda [12], illustrating an inherent problem in objective rank combination, whereby the rank of a class can paradoxically be affected by classes with worse ranks if they are removed, which violates the principle of independence of irrelevant alternatives. This paradox is a precursor to the famous Arrow's impossibility theorem in social choice theory [13]. In order to avoid negative operation, we modify the above formula to:

$$T'(\theta) \equiv f_{\theta}(r^{(1)}, \dots, r^{(J)}) = \sum_{j=1}^J (S - r^{(j)}(\theta)) \quad (4.3)$$

where S is the total number of classes.

For example, for the same pattern x in Figure 7, the outputs of SVM, MQDF, and LeNet-5 are $[4,8,7,3,5,2,6,0,9,1]$, $[9,4,8,7,5,3,2,1,6,0]$, and $[7,8,3,6,5,2,0,9,4,1]$. The $T'(\theta)$ in each of the classifiers is as follows:

Table 3: Scores of an example pattern in BC

Classes($r^{(j)}(\theta)$)	0	1	2	3	4	5	6	7	8	9
Scores in SVM	2	0	4	6	9	5	3	7	8	1
Scores in MQDF	0	2	3	4	8	5	1	6	7	9
Scores in LeNet-5	3	0	4	7	1	5	6	9	8	2
$T'(\theta)$	5	2	11	17	18	15	10	22	23	12

Accordingly, the final result is $\mu = 8$. Although the final result is still incorrect, the correct one, which is $\mu = 7$, is ranked at the second.

In conclusion, the strength of the BC method is in its simplicity, where all classifiers are treated equally in combination and where the training of the combination rule is not required. These are also its problems as it does not argue or discount for superior or inferior classifiers nor does it differentiate between low and high ranks in its combination strategy.

4.2.2.3 Weighted Borda Count (WBC)

A mixture of classifiers of various types, numerical scores such as distances to prototypes, values of an arbitrary discriminant, and distances to margins are not directly usable because their scales are not compatible with each other. Thus, calculating a confusion matrix for each confidence level of the given pattern used as weights for different classes in each classifier is necessary. The confusion matrix (Table 4) is calculated from the overall performance of each classifier on the training set.

Table 4: Confusion matrix of three classifiers

	0	1	2	3	4	5	6	7	8	9
SVM	0.9990	0.9938	0.9961	0.9921	0.9919	0.9933	0.9917	0.9803	0.9908	0.9841
MQDF	0.9949	0.9859	0.9915	0.9891	0.9837	0.9922	0.9843	0.9786	0.9856	0.9554
LeNet-5	0.9898	0.9850	0.9864	0.9822	0.9786	0.9675	0.9823	0.9715	0.9846	0.9623

This model is a generalization of the BC in that it replaces the sum of ranks with a weighted sum of ranks. The WBC combiner assigns different weights to different component classifiers according to their performance characteristics. By assigning relatively higher weights to more accurate classifiers, the WBC shows preference for their rankings in its combination function. Similar to previous section, we define μ in (4.1) be the output of combination and M_i be the confidence value of class i of the combination. The WBC score function is

$$T(\theta) \equiv f_{\theta}(r^{(1)}, \dots, r^{(J)}) = \sum_{j=1}^J w_k r^{(j)}(\theta) \quad (4.4)$$

where

$$M_i = \sum_{k=1}^S r^{(j)}(\theta) \times w_k \quad (4.5)$$

If we apply Total Probability Theorem to the above formula, w_k and $r^{(j)}(\theta)$ are calculated as follows:

$$w_k = P_k(\text{classifier}_k(x) = j | x \in C_i) \cdot P_k(\text{classifier}_k(x)), \quad (4.6)$$

where

$$P_k(\text{classifier}_k(x) = j | x \in C_i) = \frac{C_{ij}^{(k)}}{N_i}, \quad (4.7)$$

$$P_k(\text{classifier}_k(x)) = \frac{D^{(k)}}{M}, \quad (4.8)$$

and

$$r^{(j)}(\theta) = \text{Conf}_{ik} = \begin{cases} 2 \times (10 - R_{ik}), & \text{if } R_{ik} = 1 \\ 10 - R_{ik}, & \text{if } R_{ik} = 2, 3, \dots, N \end{cases} \quad (4.9)$$

Conf_{ik} and R_{ik} represent the confidence value and the rank of a pattern, respectively. For Conf_{ik} and R_{ik} , k represents the classifier, and i represents the class. S is the number of classifiers, and N represents the total number of classes. Moreover, $C_y^{(k)}$ denotes the number of patterns with actual class i that is assigned to class j by the classifier k . N_i denotes the number of patterns whose actual class is i . $D^{(k)}$ denotes the number of patterns correctly classified by the classifier k . M denotes the total number of patterns in training set. Therefore, $C_y^{(k)}$ is a 10×10 confusion matrix in classifier k , and the conditional probabilities $P_k(\text{classifier}_k(x) = i | x \in C_i) \cdot P_k(\text{classifier}_k(x))$, $k = 1, 2, \dots, s$ are the probabilities calculated from the overall performance of each classifier on the training set. The confidence value of the first rank is doubled by the formula $2 \times (10 - R_{ik})$ because we assume that the first rank should have a higher probability than others to be the true label, hence it weighs double in these excellent classifiers.

The performance of each classifier $P_k(\text{classifier}_k(x))$ is calculated as follows: When $k = 1, 2, 3$, $P_k(\text{classifier}_k(x))$ stands by the performance of classifier SVM, MQDF, and LeNet-5, respectively.

$$P_1(\text{classifier}_1(x)) = 0.9923$$

$$P_2(\text{classifier}_2(x)) = 0.9844$$

$$P_3(\text{classifier}_3(x)) = 0.9802$$

Let us go back to the example of a pattern in Figure 7 and results in Table 2. in previous two sections. For the same pattern x in Figure 7, $T'(\theta)$ in each classifier is changed as follows:

Table 5: Scores of an example pattern in WBC

Classes(θ)	0	1	2	3	4	5	6	7	8	9
Scores in SVM	2	0	4	6	18	5	3	7	8	1
Scores in MQDF	0	2	3	4	8	5	1	6	7	18
Scores in LeNet-5	3	0	4	7	1	5	6	18	8	2
$r^{(j)}(\theta)$	5	2	11	17	27	15	10	31	23	21
$T'(\theta)$	4.9	1.9	10.8	16.5	26.4	14.6	9.7	29.7	22.4	19.8

Moreover, we need to calculate the final results with weights. We calculate $T'(7)$ and $T'(8)$ as examples:

$$\begin{aligned} T'(7) &= 0.9923 * 0.9803 * 7 + 0.9844 * 0.9786 * 6 + 0.9802 * 0.9715 * 18 \\ &= 29.7301 \end{aligned}$$

$$\begin{aligned} T'(8) &= 0.9923 * 0.99076 * 8 + 0.9844 * 0.9856 * 7 + 0.9802 * 0.9846 * 8 \\ &= 22.3775 \end{aligned}$$

Finally, we calculate the final result $\mu = 7$, which is correct. Accordingly, we find that considering the preference and confidence of each class in each classifier and rank is necessary in the combination of MCS.

The general schema of cooperation in this system is as follows (Figure 8).

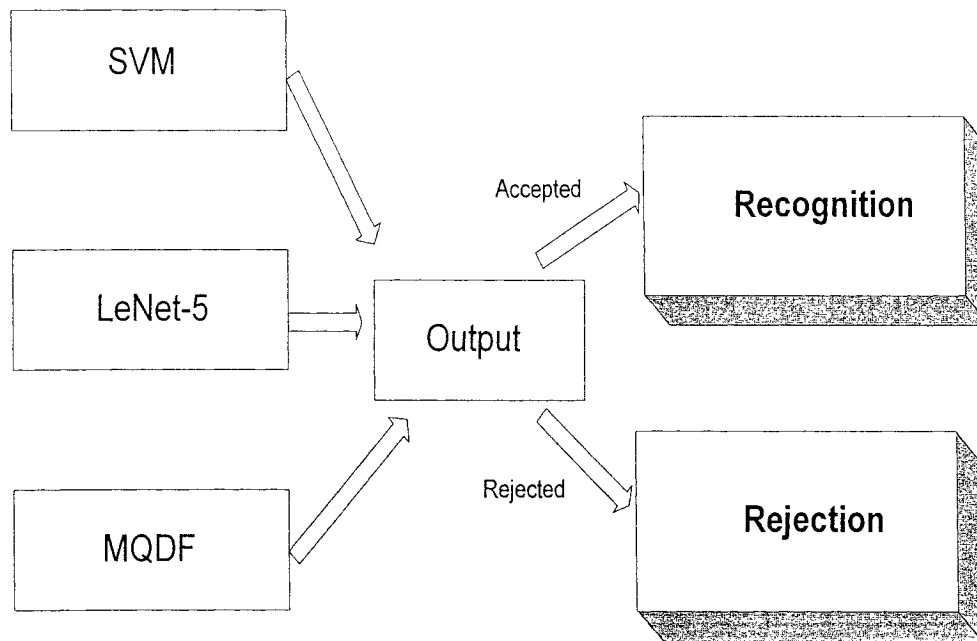


Figure 8. Flowchart of combination among three classifiers

The advantages of this WBC framework are multiple. At first, at the rank level, measurements of different types of classifiers are the same. Secondly, the most difficult patterns are filtered by several *experts* instead of one. Thirdly, both *confidence* (e.g. the first rank has double scores) and *preference* (e.g. preference for classifiers with higher general accuracy) are taken into consideration in combination, which is more reliable and reasonable in real life.

As a result, although WBC is better than cooperation of three classifiers, we want to find a better solution with a higher recognition rate while keeping high reliability. Therefore, we can integrate cooperation with combination.

4.2 HMCS in this chapter

Here, we study a hybrid system formed by the cooperation and combination of multiple classifiers. In other words, a classifier can transmit its decision, either acceptance or rejection, to one or more other classifiers [11]. The architecture of this hybrid system is shown in Figure 9.

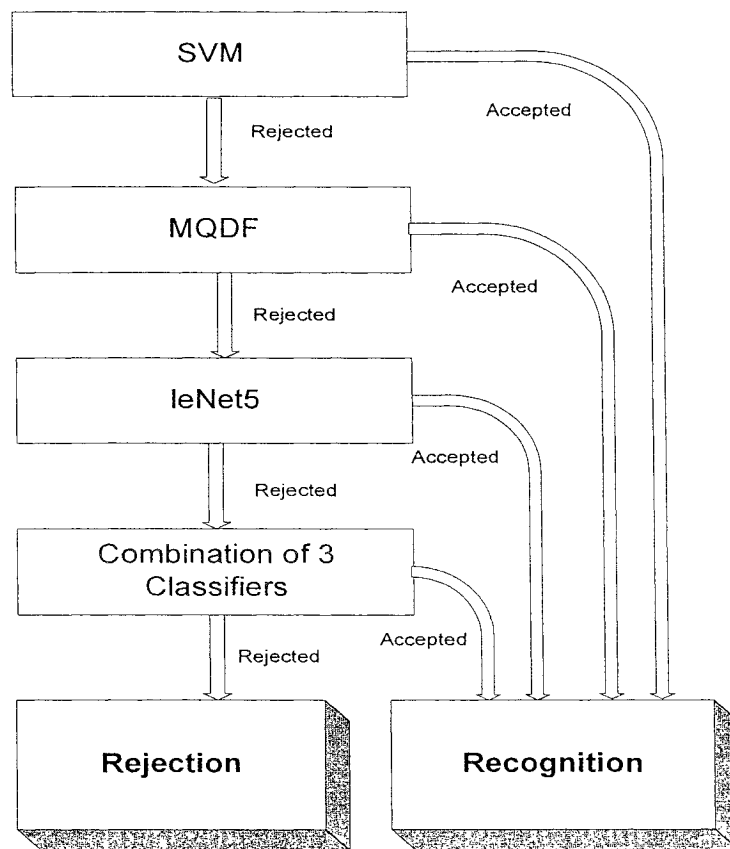


Figure 9. A Hybrid system of multiple classifiers

In cooperation, we choose a sequence of multiple classifiers according to their recognition accuracy. Firstly, we apply SVM, which has the highest accuracy, and recognizes the patterns with high measurement values. Next, we apply MQDF and LeNet-5 respectively to the patterns rejected by SVM. Finally, the patterns rejected by all

of the three classifiers are sent to the combination of MCS. As a result, HMCS has a better performance than cooperation and combination singly.

Chapter 5

Error Analysis

In this chapter, we analyze the errors and observe the relationship between normalization sizes and recognition rates in this HMCS. We propose to analyze the substitution images with different normalization sizes, but the same classifier and same features extracted to predict the relationship between normalization sizes and recognition rates. After constructing a smaller database of difficult original patterns from NIST, we found that normalizing the original data to a size larger (e.g. $26 * 26$) than $20 * 20$ in MNIST further increases the recognition rate.

5.1 Introduction

Generally, a character recognition system includes three main tasks: pre-processing, feature extraction, and classification. In pre-processing, researchers normally work on noise filtering, binarization, thinning [19], skew correction [18], slant normalization [17], etc. to enhance the quality of images and correct the distortion; in feature extraction,

various types of features and extraction techniques are available; in classification, a great number of classifiers can be used, such as statistical classifiers, support vector machines (SVMs), neural networks, and Multiple Classifier Systems (MCSs).

Although correctly selecting the options for each task helps to improve the overall recognition rate, one crucial factor of affecting the recognition rate is always ignored. This crucial factor is size normalization. Since some MNIST data are not noise-free, they are not good enough to be directly recognized by small images. In this process, we focus mainly on the role of size normalization on the recognition of handwritten numerals.

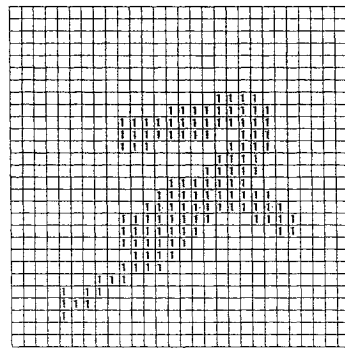
Most researchers agree that the substitution is mainly caused by the quality of images, or distortion of images; according to long-periods of observations and experiments, we suspected that low resolution is an important factor, which reduces the recognition rates of OCR systems.

5.2 Pre-processing & Feature extraction

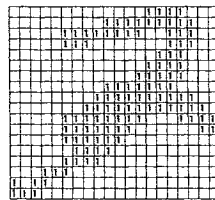
At first, we work on pre-processing, which includes size normalization. We normalize images of MNIST from $20 * 20$ to bigger sizes, and then extract gradient features from them in order to train and test the classifiers.

5.2.1 Pre-processing

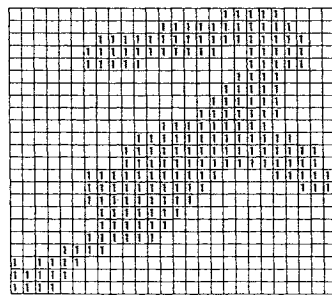
In size normalization, we keep the aspect ratio of the images and normalize them to a bigger size. First, we binarize and cut the original images (Figure 10(a)) of an MNIST numeral into a rectangle with the same height and width of the original patterns (Figure 10(b)). After that, we enlarge the images to a fixed size (e.g. $26 * 26$) using a bilinear interpolation algorithm (Figure 10(c)) [27]. Finally, we place the new normalized images at the center of an empty image with size $32 * 32$ (Figure 10(d)) for the extraction of gradient features.



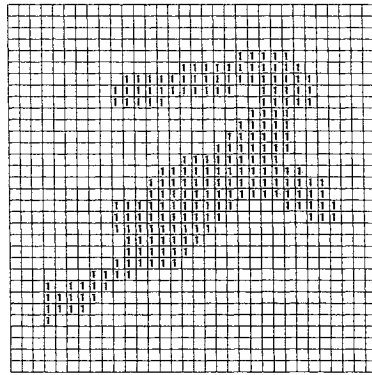
(a)



(b)



(c)



(d)

Figure 10. Sample images in size normalization

(a) an original image in MNIST; (b) a cut image from MNIST;
(c) an enlarged image; and (d) an image for feature extraction

In size normalization, we use bilinear interpolation for normalization. By translating and rescaling the coordinates, which will not change the interpolation, we may suppose the square is centered at (x, y) and the centers of the surrounding cells located at $(0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$, where they have values Z_{00} , Z_{10} , Z_{01} , and Z_{11} , respectively.

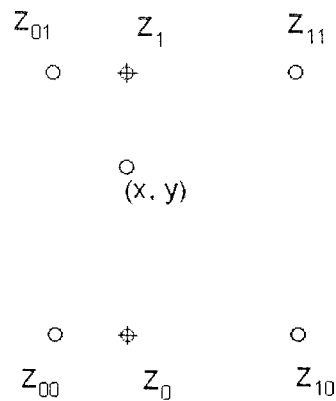


Figure 11. An example of bilinear interpolation

Linear interpolation on the bottom row of neighbors, between $(0, 0)$ and $(1, 0)$, estimates the value Z_0 at $(x, 0)$ as $x * Z_{10} + (1 - x) * Z_{00}$. Likewise, linear interpolation on the top row of neighbors, between $(0, 1)$ and $(1, 1)$, estimates the value Z_1 at $(x, 1)$ as

$x * Z_{11} + (1 - x) * Z_{01}$. Finally, linear interpolation between Z_0 and Z_1 estimates the value Z at (x, y) as $y * Z_1 + (1 - y) * Z_0$.

The key idea of bilinear interpolation is to perform linear interpolation first in one direction, and then in the other direction. By substituting the expressions for Z_0 and Z_1 into the previous formula you can see that the formula for Z is a polynomial involving powers of x and y no greater than 1, so it has four coefficients:

$$Z = a + b * x + c * y + d * x * y \quad (5.1)$$

Because these four coefficients were determined by four values (Z_{00}, \dots, Z_{11}), they are in general uniquely determined by the data. This immediately implies that the comparable procedure of first interpolating along columns (in the y -direction) and then interpolating the results in the x -direction will give the same result, because it, too, will have a similar formula with a unique solution.

Note that the term "bilinear" derives from the process of linear interpolation (twice in one direction, then once in the perpendicular direction), *not* from the formula for Z . The formula involves a term with $x * y$, which is not linear.

5.2.2 Feature extraction

Gradient features [21] are extracted from the binary images in the MNIST database in this study. In each pattern, a feature vector with a size 400 (5 horizontal, 5 vertical, 16 directions) is produced.

The gray-scale normalized image is standardized such that its mean and maximum values are 0 and 1.0, respectively. After centering a normalized image (e.g. $26 * 26$) into

a 32 * 32 box as mentioned in 5.2.1, Robert filter [24] is applied to calculate gradient strengths and directions. For example, the gradient magnitude and direction of pixel $g(m, n)$ are calculated as follows:

$$\Delta u = g(m, n) - g(m+1, n+1), \quad (5.2)$$

$$\Delta v = g(m, n+1) - g(m+1, n), \quad (5.3)$$

$$\theta(m, n) = \arctan\left(\frac{\Delta v}{\Delta u}\right), \quad (5.4)$$

$$s(m, n) = \sqrt{\Delta u^2 + \Delta v^2}, \quad (5.5)$$

where $\theta(m, n)$ and $s(m, n)$ specify the direction and gradient magnitude of pixel (m, n) , respectively.

We calculate the strength of the gradient as a feature vector. The direction of gradient is quantized to 32 levels with an interval $\pi/16$. The normalized character image is divided into 81 (9 horizontal * 9 vertical) blocks. The strength of the gradient in each of the 32 directions is accumulated in each block to produce 81 local joint spectra of direction and curvature.

The spatial resolution is reduced from 9*9 to 5*5 by down sampling every two horizontal and every two vertical blocks with a 5*5 Gaussian filter. Similarly, the directional resolution is reduced from 32 to 16 levels by down sampling with a weight vector $[1\ 4\ 6\ 4\ 1]^T$, to produce a feature vector of size 400 (5 horizontal, 5 vertical, and 16 directions).

Moreover, variable transformation ($y = x^{0.4}$) is applied to make the distribution of the feature Gaussian-like. The feature size is reduced to 400 by principal component analysis (KL transform).

Finally, we scale the feature vectors by a constant factor such that the values of feature components range from 0 to 1.0.

5.3 Recognizing images in MNIST with different sizes

We applied two different classifiers -- SVM and MQDF to test all the patterns in our MNIST test set to observe the recognition rate at different sizes. The reason for choosing two classifiers is to ensure that the normalization size affecting the recognition rate of the system is not happening because of these classifiers. LeNet-5 classifier cannot be used for this analysis because it has a fixed structure (detailed in Chapter 2), and normalizing the sizes of patterns may destroy its structure.

As a result, we find that the recognition rates rise with both SVM and MQDF when we enlarge the images (shown in Chapter 6). As images in MNIST have already been normalized, normalizing them to a bigger size is the second source of distortion of the originals. Even though the data underwent distortion (normalization) twice, the recognition rates still rise from **98.98% to 99.23% in SVM** and from **89.79% to 98.44% in MQDF** when we enlarge the images from $20 * 20$ to $30 * 30$. This suggests that if we normalize the original image to a bigger size than $20 * 20$, the recognition rate of the entire system will rise because we only need to normalize images from the originals once instead of twice. To prove whether this hypothesis is true or not, we constructed a small database with originals.

5.4 Finding the originals to construct a small database

Since the MNIST database was constructed from NIST's Special Database 3 and Special Database 1, which contain binary images of handwritten digits, and NIST Special Database 19, which includes NIST's Special Database 3 and Special Database 1, consequently we should be able to match all the images between the normalized images from MNIST and the original images from NIST SD 19.

In total, we found 417 substitution images with eight different sizes (Table 6). In order to create a database with the most difficult ones, we constructed a small database with images not recognized in at least two different sizes.

Table 6. Numbers of error images with different sizes

No. of sizes	8	7	6	5	4	3	2	1
No. of substituted patterns	12	21	23	15	18	33	59	236
Total of substituted patterns	181							236

Thus, we have in total 181 substitution images (An index of these images is listed in Appendix I). The distributions of each class in this small database are as follows (Table 7) (An index matched in NIST and MNIST is listed in Appendix II):

Table 7. Distribution of samples of each class in small database

Class	0	1	2	3	4	5	6	7	8	9	Total
No.	8	8	17	17	19	18	15	22	24	33	181

Since NIST SD 19 is too huge to match images one by one manually, we have implemented an automatic procedure to effectively apply template matching globally with some constraints. At first, we find all the substitution images in MNIST and sort them into ten classes (0,1, ..., 9); then we load one substitution image in a class. Subsequently, we cut the image to the real size; in other words, we remove four

boundaries and keep the real images; afterwards, we load the images in NIST SD 19 and normalize them to the same size of the cut image. Further, we match two images with template matching in order to choose a candidate image, and finally, we verify the candidate image with local structures. The details are shown in Figure 12.

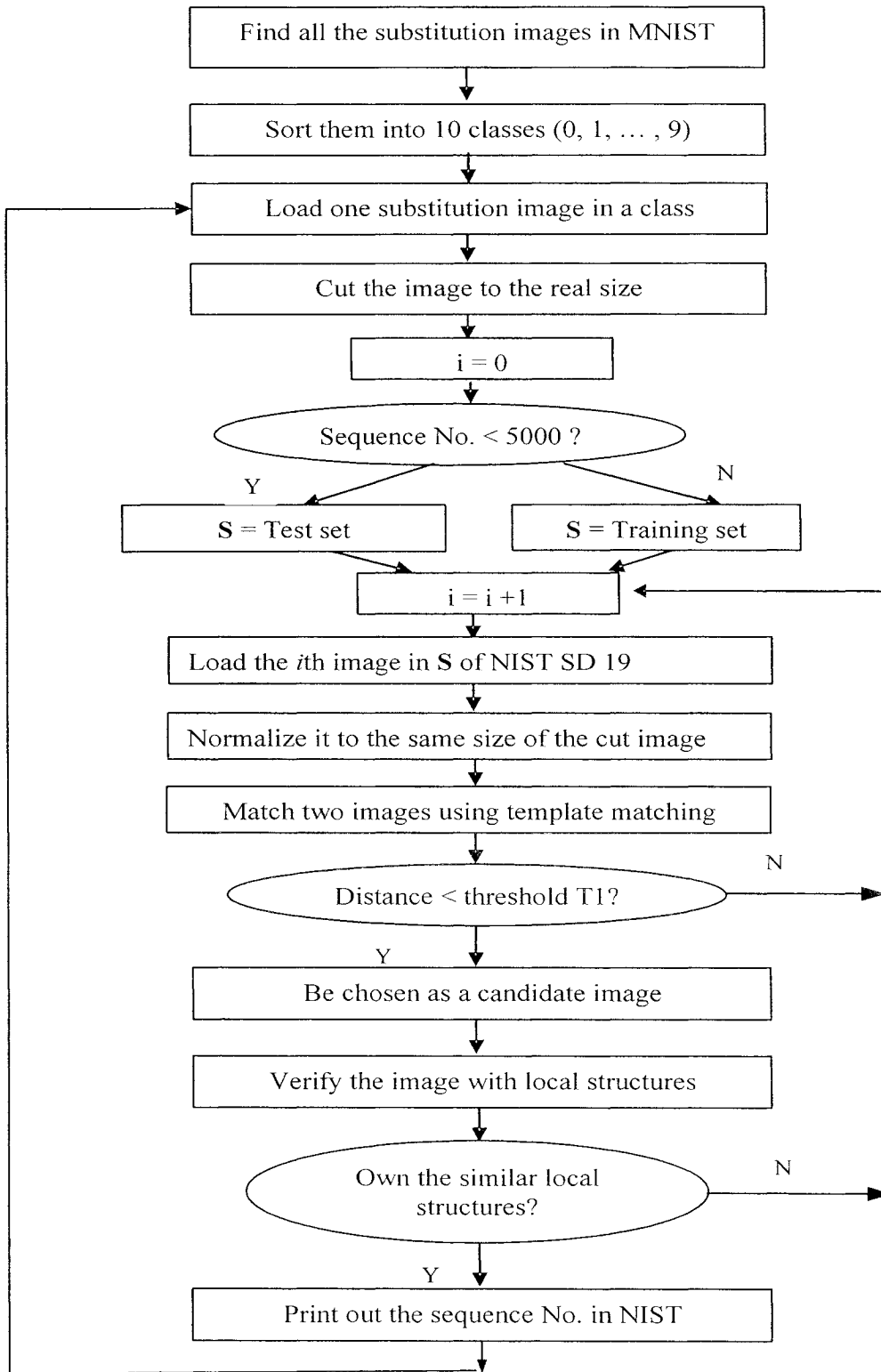


Figure 12. Flowchart of finding the originals of substitution images

During the procedure of matching two images, we apply a template matching algorithm. Template matching approaches have been quite popular in optical computing: frequency domain characteristics of convolution are used to simplify the computation. It can be simplified significantly in binary images. In the following, we introduce template matching first, and mention conditions for matching, such as constraints on number of dissimilar pixels and aspect ratio (height to width) in an image.

Suppose that we have a template $g[i, j]$ and we wish to detect its instances in an image $f[i, j]$. An obvious thing to do is to place the template at a location in an image and to detect its presence at that point by comparing intensity values in the template with the corresponding values in the image. Since it is rare that intensity values will match exactly, we require a measure of dissimilarity between the intensity values of the template and the corresponding values of the image. Several measures may be defined:

$$\max_{[i,j] \in R} |f - g| \quad (5.6)$$

$$\sum_{[i,j] \in R} |f - g| \quad (5.7)$$

$$\sum_{[i,j] \in R} (f - g)^2 \quad (5.8)$$

where R is the region of the template. Here, we take the entire error image as a template and calculate the similarities between error images and original images with formula (5.8) because the sum of the squared errors is the most popular measure. In the case of template matching, this measure can be computed indirectly and computational cost can be reduced. We can simplify:

$$\sum_{[i,j] \in R} (f - g)^2 = \sum_{[i,j] \in R} f^2 + \sum_{[i,j] \in R} g^2 - 2 \sum_{[i,j] \in R} fg \quad (5.9)$$

Our aim is to find patterns with minimum distances in (5.7) or patterns with distances smaller than a certain threshold value. As f and g are fixed, then $\sum fg$ gives a measure of mismatch in (5.9). Thus, we only need to find patterns with maximum values of $\sum fg$.

In the procedure of matching two images, the image has to satisfy the following constraints: (i) number of dissimilar pixels is not big and (ii) having similar aspect ratios. If any image satisfies (i) and (ii), it is considered as a candidate image; otherwise, if no image satisfies the two conditions, K in (i) need to be enlarged until an image or several images are found as candidate images.

(i). We use K in the following formula to represent the measure of similarity between two images:

$$K = \max \sum fg \quad (5.10)$$

Here, we need K to satisfy the following condition: $K \leq (h_{substituti\ on} * w_{substituti\ on}) / c_1$, $h_{substituti\ on}$ is the height of the current substituted image, $w_{substituti\ on}$ is the weight of the current substituted image; and c_1 is a constant. Experimentally, we set $c_1 = 6$.

(ii). The difference between the aspect ratio of the original images and the aspect ratio of the current error image should be small. In (5.11), we experimentally set $c_2 = 0.1$. $r_{original}$ is an aspect ratio of an original image, and $r_{original}$ is an aspect ratio of an error image.

$$|r_{original} - r_{template}| \leq c_2 \quad (5.11)$$

When verifying the original images from candidate images, we consider two situations. If the minimum distances in template matching are very small, we endorse the images with minimum distances as their originals. However, if the minimum distances are too big, we need to consider all the candidate images with their local structures in order to find their original images. We considered choosing several candidates instead of choosing one is because, during the verification, we found that two specific situations occurred when the minimum distances were large.

Conditions:

- a) If $d(x, y) = \|D_{\min}(x, y) - D_{2nd \min}(x, y)\| \leq T$, we will match all the candidate images; otherwise, we will assign the image with $D_{\min}(x, y)$ to the image in MNIST as a matching pair, where x is the pattern in MNIST, y is the pattern in NIST, $D_{\min}(x, y)$ is the distance between x and y with template matching, and $d(x, y)$ is the distance between $D_{\min}(x, y)$ and $D_{2nd \min}(x, y)$, and T is a constant.
- b) If $d(x, y_1) = d(x, y_2) \ \& \ r(x, y_1) \leq r(x, y_2)$, where $r(x, y_i) = \|R_{MNIST}(x) - R_{NIST}(y_i)\|$, ($i = 1, 2$), we assign the image with y_1 , where $R_{MNIST}(x)$ is an aspect ratio of pattern x in MNIST.

If all the images satisfy condition a), which means that the first candidate image is too similar to the second candidate image, we need to look and compare their local geometric structures to those of the image in MNIST. The patterns in Figure 13 serve as an example. Although the 1st candidate (far right image in NIST) has the minimum distance, the second middle candidate (centre image in NIST) is the real match of the image in MNIST (the left one). Accordingly, considering the candidate images is necessary.


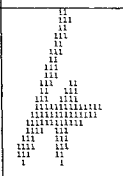

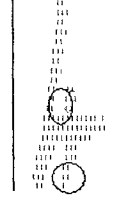
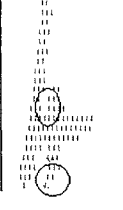
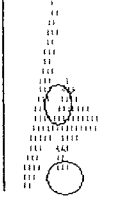
	Image in MNIST	Images in NIST	
Original Images			
Local Structure			
Distance	/	43	37
Results	/	√	×

Figure 13. An example that candidate images are considered

Another case is the situation that condition b) is satisfied. If condition b) is satisfied, which means that the matched image in MNIST owns two candidate images with minimum distance in NIST, the aspect ratio should be considered. Let us look at the patterns in Figure 14. We have determined that the matching image is the middle one because it has the same aspect ratio as the image in MNIST.




	Image in MNIST	Images in NIST	
Original Images			
Distance	/	44	44
Aspect Ratio (H/W)	$20/13=1.54$	$72/48=1.5$	$51/38=1.34$
Results	/	√	×

Figure 14. An example where the aspect ratios are considered

5.5 Comparing the substitution rates of the small database

While keeping their aspect ratios, we normalized the original images to various sizes and recognized the normalized images by the same feature extraction algorithm and classifier –SVM.

According to the statistical results shown in Chapter 6, we found that enlarging images helps to increase the recognition rate. When the images are normalized to the same sizes from both the originals and MNIST, images normalized from the originals have better performance and lower substitution rate. Hence, we conclude that:

- (1) Enlarging images helps to increase the recognition rate,
- (2) Normalizing images from the originals has a better performance than normalizing images from MNIST.

Chapter 6

Experimental Results

In this chapter, we introduce the database used in this thesis, and we demonstrate the experimental results of HMCS and error analysis, respectively. In HMCS, we show not only the results of each classifier concerning the rejection option in cooperation, but also the results of different combination methods. Afterwards, we compare the performance of HMCS with each individual classifier. In error analysis, we show the experimental results of recognition rates with different normalization sizes in SVM and in MQDF. Moreover, we show some statistics on sizes of patterns in NIST. Finally, we compare the error rates in the small database with various sizes normalized from patterns in either MNIST or NIST.

6.1 Database

The experiment was conducted on the MNIST database [2], which is a widely known handwritten digit recognition benchmark. The MNIST database of handwritten digits has a training set of 60,000 samples and a test set of 10,000 samples. The MNIST database is a subset of a larger set available from NIST. The digits of MNIST have been size-

normalized and centered into a fixed-size image. The original black and white (bi-level) images from NIST were size normalized to fit into a 20*20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. Each image was then centered in a 28*28 image by calculating the point, which is the center of the mass of all the pixels for each image, and shifting the image so as to position this point at the center of the 28*28 field. The following patterns are samples of MNIST (Figure 15).



Figure 15. Samples of MNIST

The formulae for calculating the recognition rate, substitution rate, rejection rate and reliability, respectively, are:

$$\text{Recognition rate} = \frac{\text{Number of correctly recognized samples}}{\text{Total number of test samples}} \times 100\%$$

$$\text{Substitution rate} = \frac{\text{Number of incorrectly recognized samples}}{\text{Total number of test samples}} \times 100\%$$

$$\text{Rejection rate} = \frac{\text{Number of rejected samples}}{\text{Total number of test samples}} \times 100\%$$

$$\text{Reliability} = \frac{\text{Recognition rate}}{100\% - \text{Rejection rate}}$$

6.2 Experimental Results of the Entire System

We designed an HMCS, which integrates the cooperation (serial topology) and combination (parallel topology) of three classifiers: SVM, MQDF, and LeNet-5 developed by Dong [1]. In cooperation, Differential Measurement (DM), Probability Measurement (PM) and First Rank Measurement (FRM) are defined for their rejection options on different types of classifiers. As we know, DM is more reliable than FRM as

DM uses distribution and relationship among all the confidence values. Thus, DM is applied to SVM and MQDF for their rejection options. Moreover, PM is applied to LeNet-5. In the process of combination, Weighted Borda Count (WBC) at the rank level with the Total Probability Theorem to the three classifiers is applied. WBC has better performance in comparison with Majority Vote, and Borda Count. In order to further improve the reliability of this HMCS, a verifier can be plugged into it.

6.2.1 Experimental Results of the Cooperation

In cooperation with this HMCS, SVM, LeNet-5, and MQDF were applied serially. As we look at rejected patterns from a previous classifier as input for the next classifier, measurements of the rejections are vital. In this section, we demonstrate the results from various measurements mentioned in Chapter 3 with different classifiers.

First, we compared the recognition results of SVM classifier using FRM or DM, as shown in Figure 16 and Figure 17. As both FRM and DM are designed for its rejection options, we chose an ideal measurement, which has a higher recognition rate while keeping the same low substitution rate for a classifier.

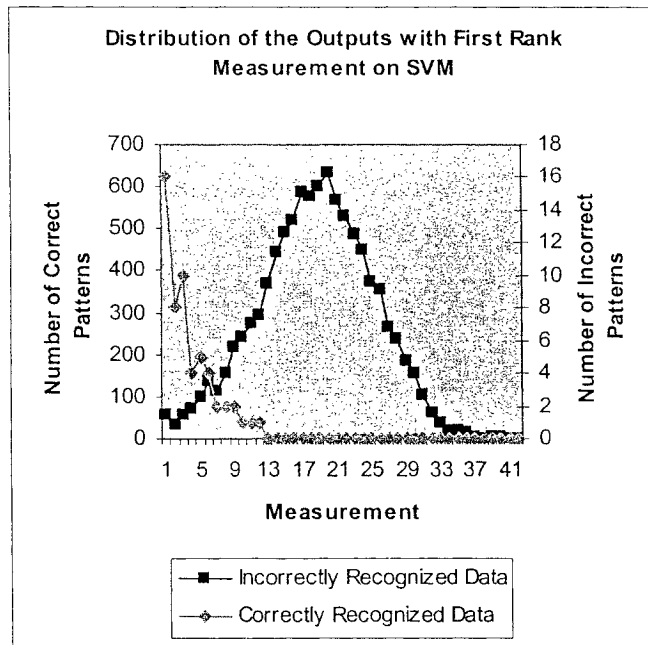


Figure 16. Distributions of recognition results of classifiers SVM using FRM

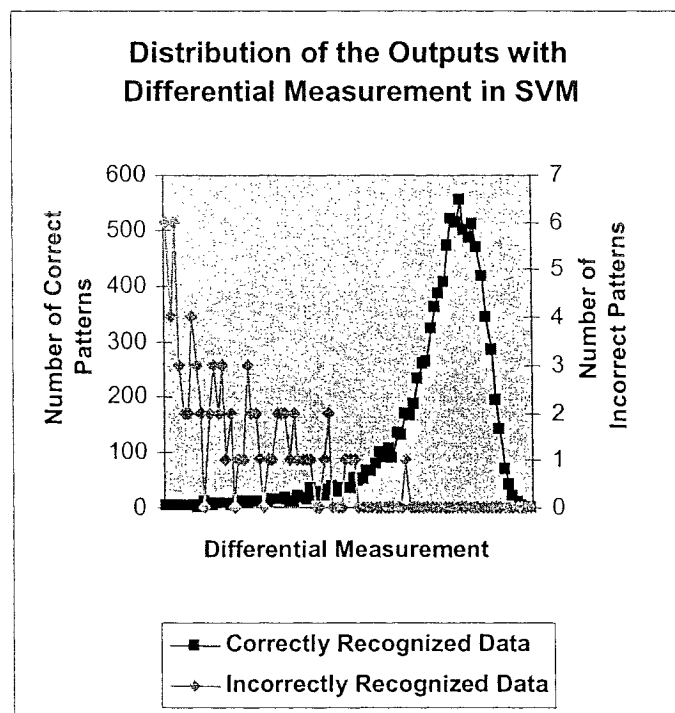


Figure 17. Distributions of recognition results of classifiers SVM using DM

According to the experiment on the training set of MNIST, the range of the confidence value of the FRM in SVM is from - 0.618123 to 4.15665. The maximum value of the FRM in the substitution set in SVM is 1.2. Table 8 shows the recognition and substitution rates of SVM against various thresholds of the FRM.

Table 8. Thresholds for rejection of SVM based on the confidence value of FRM

>Threshold	1.2	1.1	1.0	0.6	0	- 0.6
Recognition Rate	81.09%	85.49%	86.76%	94.01%	98.54%	99.23%
Substitution Rate	0%	0.01%	0.02%	0.09%	0.56%	0.77%

According to the experiment on the training set of MNIST, the range of the confidence value of the DM in SVM is from 0 to 8.6. The maximum value of the DM in the substitution set in SVM is 5.65. Table 9 shows the recognition and substitution rates of SVM against various thresholds of the DM.

Table 9. Thresholds for rejection of SVM based on the confidence value of DM

>Threshold	5.65	4.52	3.39	2.26	1.13	0
Recognition Rate	82.32%	93.42%	97.15%	98.55%	99.53%	99.23%
Substitution Rate	0%	0.01%	0.09%	0.22%	0.43%	0.77%

Based on the recognition results presented in Table 8 and Table 9, DM is better than FRM. The recognition rates are 93.42% in DM and only 85.49% in FRM, while both have the same substitution rate with 0.01%. Therefore, we used DM as a measurement for classifier SVM's rejection.

Similarly, DM is a good measurement for classifier MQDF. The distributions of the correctly recognized data and erroneous data are shown in Figure 18. As illustrated, the correctly recognized data are distributed in a Gaussian shape where the peak of the distribution is far away from the peak of the distribution of substitution patterns.

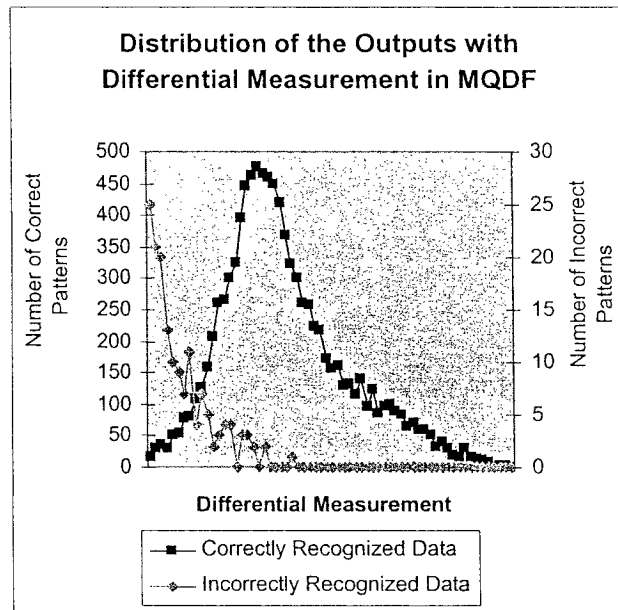


Figure 18: Distributions of recognition results of classifier MQDF using DM

Despite its benefits, DM is not a good measurement for classifier LeNet-5. As shown in Figure 19, it is hard for DM to find a threshold for rejection that can properly separate correctly recognized data and errors when used with LeNet-5.

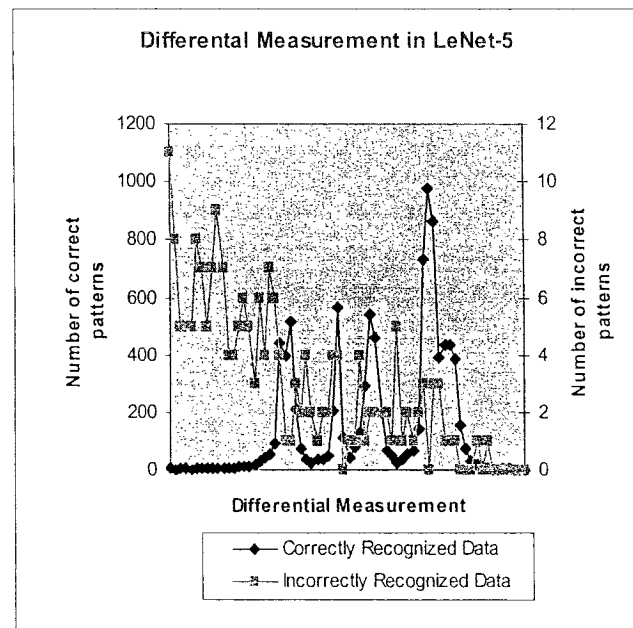


Figure 19: Data distributions of LeNet-5 using DM

On the other hand, PM is an effective tool to evaluate the outputs of LeNet-5, as shown in Figure 20. Over 60% of patterns in the entire test set have a PM greater than 0.99, while there is no substitution in that range.

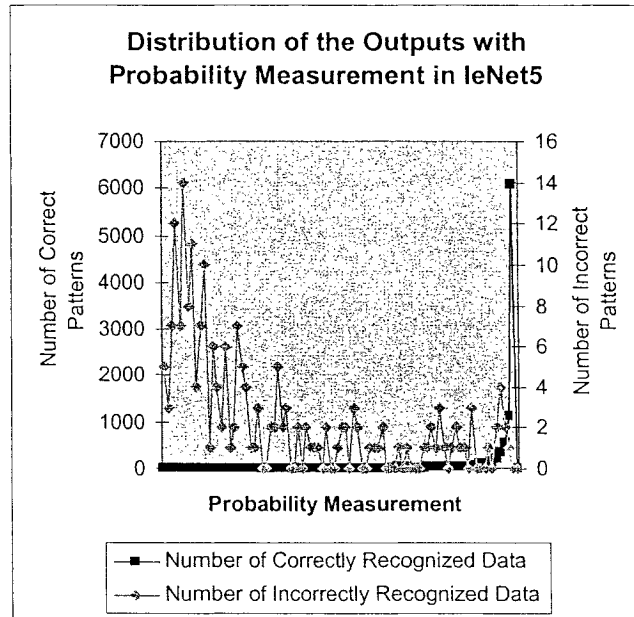


Figure 20: Distributions for LeNet-5 using PM

When using DM with a fixed threshold, SVM rejects a number of patterns. The distribution of these patterns in LeNet-5 with correct and incorrect recognition data are shown in Figure 21. Although it seems that the range of measurement of incorrect data is almost the same as the range of correct data, the amount of data with the measurement from 88 – 90 have a big difference. Thus, using PM in LeNet-5 to deal with the incorrectly recognized patterns in SVM is reasonable. We reject patterns which cannot perform well in either SVM or LeNet-5.

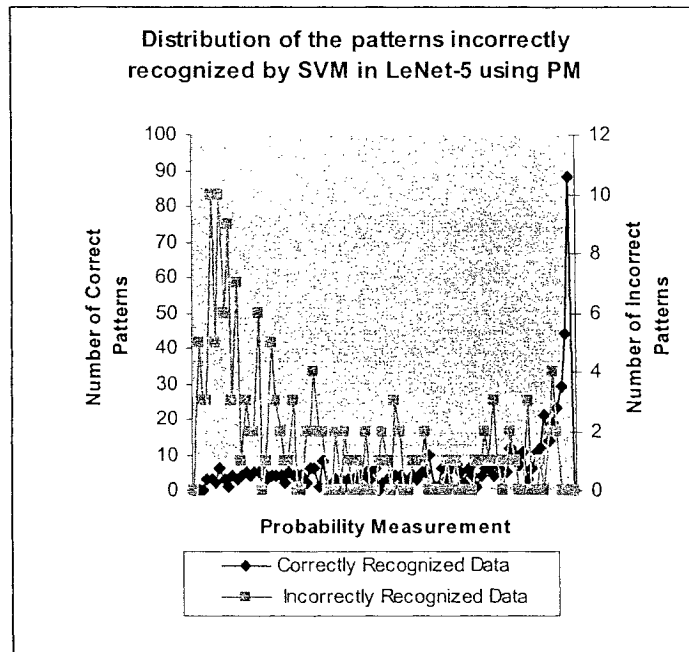


Figure 21. Distribution of patterns incorrectly recognized in SVM in LeNet-5 with PM

In addition, we can analyze the distribution of rejection from the former two classifiers in MQDF in the same way. In order to maintain the substitution rate, we control MQDF's threshold strictly.

6.2.2 Experimental Results of the Combination

In combination of this HMCS, we integrated three classifiers with a parallel topology. We compared the results of WBC, a reliable combination method at rank level introduced in Chapter 4 with the results of well known methods, such as Majority Vote and BC.

Majority Vote is used for the combination of MCS at the abstract level. If at least two of classifiers give same result, we confirm it; otherwise, we reject it. After combination occurs in this way, the recognition rate is reduced to 97.92% and the rejection rate is

0.24%. Because voting in abstract level does not include enough information from the three classifiers' outputs, the recognition rate of majority vote in abstract level is not high enough. Borda Count is used for the combination of MCS at the rank level. As a result, the recognition rate is 98.78% without rejection.

WBC is a weighted BC of MCS at the rank level. According to the experiments, the recognition rate of WBC is 99.00% without rejection, which is better than both Majority Vote and BC, as shown in Table 10.

Table 10. Recognition Results of WBC, BC, and Majority vote

	WBC	BC	Majority vote
Recognition Rate	99.00%	98.78%	97.92%
Substitution Rate	1.00%	1.22%	1.84%
Rejection Rate	0.00%	0.00%	0.24%

6.2.3 Experimental Results of the HMCS

The HMCS included in our study can be used to either give the final output or to work as an input to verifiers [54]. If the target is to give the final output with HMCS, we should focus on both the recognition rate and the reliability rate. However, if the target is the latter one, we should reduce the substitution rate as much as possible. Thus, if we only use HMCS without verification, after adjusting the thresholds at each stage on the training data, the overall recognition rate of the test set is 98.34%, while the reliability rate is 99.54%. However, if this MCS is considered as the input of verification by prototypes [54], the overall recognition rate is 95.54%, but the reliability rate is high at 99.93%, and the substitution rate decreases from 0.46% to 0.07%. A comparison of the

performance of HMCS with different classifiers and different strategies is listed in Table 11.

Table 11. Comparison of the performance of HMCS with individual classifier

Classifiers		Recognition Rate (%)	Error Rate (%)	Rejection Rate (%)	Reliability Rate (%)
SVM		99.23	0.77	0	99.23
MQDF		98.44	1.56	0	98.44
LeNet-5		98.02	1.98	0	98.02
Cooperation		94.86	0.02	5.12	99.93
Combination		99.00	1.00	0	99.00
HMCS	With verification	95.54	0.07	4.39	99.93
		97.00	0.14	2.86	99.86
		98.34	0.45	1.21	99.54
	Without verification	99.11	0.89	0	99.11

6.3 Experimental Results of Error Analysis

6.3.1 Error Rates at Various Sizes

Several steps are involved in error analysis. Firstly, we compare the recognition rates in both SVM and MQDF when we enlarge images to various sizes larger than the patterns in MNIST. The details are shown in Figure 22 and Figure 23.

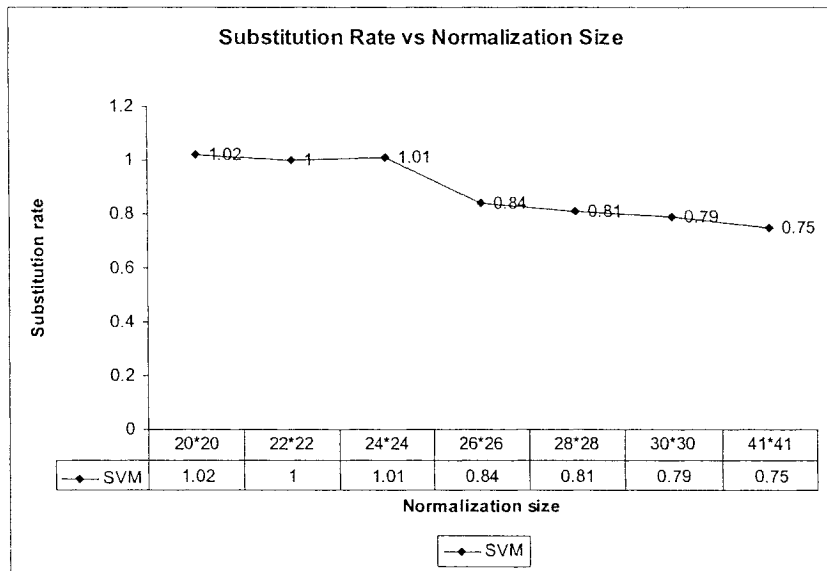


Figure 22. Substitution rates at different normalization sizes of MNIST in SVM

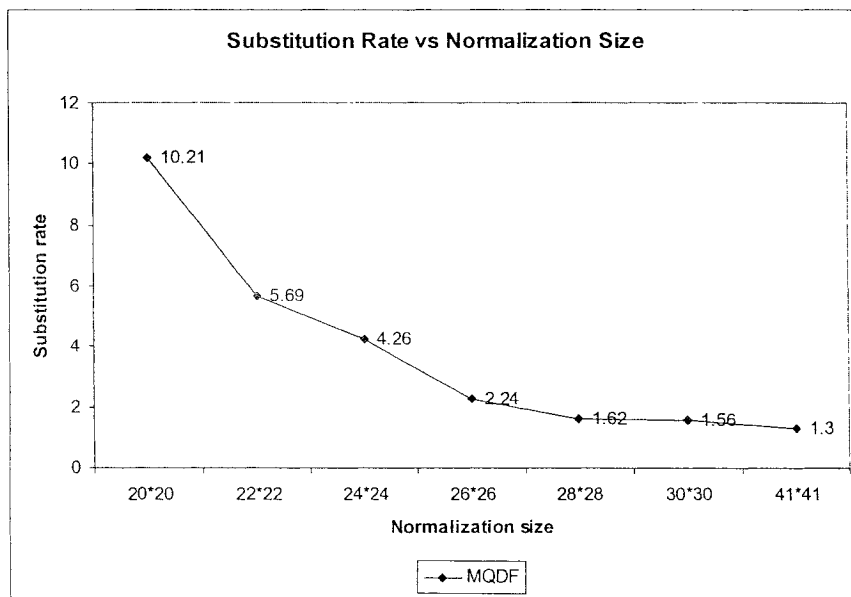


Figure 23. Substitution rates at different normalization sizes of MNIST in MQDF

As a result, we find that the recognition rates increase with both SVM and MQDF when we enlarge the images. We find that when we increase the normalization sizes from 20 * 20 to 26 * 26, the substitution rate for SVM decreases from 1.02% to 0.84% and from 10.21% to 2.24% for MQDF. When we increase the normalization sizes from 26 *

26 to $41 * 41$, the substitution rate for SVM and MQDF continues to decrease, but the differences are much smaller. We have chosen $26 * 26$ as the optimal normalization size for practical purposes in terms of performance and processing time even though the recognition rate rises when we normalize images to bigger sizes.

As images in MNIST have already been normalized, normalizing them to a bigger size leads to another source of distortion of the originals. Thus, we normalize the image to a bigger size than $20 * 20$ from the originals only once instead of twice. Hence, we constructed a small database of originals to study the real effect of size normalization.

6.3.2 Experimental Results of NIST

The MNIST database was constructed from NIST's Special Database 3 and Special Database 1, which contain binary images of handwritten digits. The NIST Special Database 19 (NIST SD 19) also includes NIST's Special Database 3 and Special Database 1. Consequently we should be able to match all the images between the normalized images from MNIST and the original images from NIST SD 19.

In order to better understand patterns in NIST SD 19, we did some statistics on NIST SD 19. According to that, the total number of handwritten labeled characters (digit and alphabetic) in NIST SD 19 is 814,255. In the training set, there are 344,307 isolated digits, and there are 58,646 isolated digits in the test set. The numbers of each class are listed in Table 12. We discovered that the images with sequence numbers ranging 0-4999 in MNIST are chosen from the test set of NIST SD19; while the images with sequence

numbers ranging from 5000-10000 in MNIST are chosen from the training set of NIST SD19.

Table 12. Numbers of each class distributed in NIST SD 19

Label	0	1	2	3	4	5	6	7	8	9
Train- ing Set	34,803	38,049	34,184	35,293	33,432	31,067	34,079	35,796	33,884	33,720
Test Set	5,561	6,655	5,888	5,819	5,721	5,539	5,858	6,097	5,695	5,813

Moreover, we found that the size of normalized samples ($20 * 20$) in MNIST is less than $\frac{1}{2}$ the average width and height of the size of the original images. According to the statistics on NIST, we found that the average size of images in the training set and that of the test set are similar. The average original width varies from 32 to 40 pixels except for class “1”, and the average original height is from 39 to 51 pixels (Table 13). Appendix III includes two tables of all the images from NIST SD 19 with maximum height, maximum width, maximum area, minimum height, minimum width, and minimum area of images in each class in both the training set and the test set.

Table 13. Summary of the width and height of the samples
in the Training and Test sets of NIST SD 19

Training set:						
Label	Number of Patterns	Range of Width	Range of Height	Mean of Width	Mean of Height	Range of Area (Width * Height)
0	34,803	13-85	14 - 83	36	39	240(15* 16)-7055 (85*83)
1	38,049	3-73	17-82	20	43	75(3*25)-5256(73*72)
2	34,184	13-123	18-80	41	42	270(15*18)-8856(123*72)
3	35,293	15-89	21-81	37	45	360(15*24)-5976(83*72)
4	33,432	16-101	9-83	40	50	336(16*21)-8181(101*81)
5	31,067	13-123	20-82	47	45	360(18*20)-8775(117*75)
6	34,079	12-85	18-81	34	47	288(16*18)-6035(85*71)
7	35,796	12-97	18-81	35	46	240(12*20)-6840(95*72)
8	33,884	15-108	21-82	37	49	418(19*22)-7560(108*70)
9	33,720	14-94	24-83	34	51	435(15*29)-6016(94*64)
Test set:						
Label	Number of Patterns	Range of Width	Range of Height	Mean of Width	Mean of Height	Range of Area (Width * Height)
0	5,561	13-79	15-79	35	41	208(13*16)-5372(68*79)
1	6,655	3-60	18-81	13	44	108(3*36)-3432(52*56)
2	5,888	14-95	15-79	38	43	210(14*15)-5775(75*77)
3	5,819	13-75	18-79	34	47	340(17*20)-5254(71*74)
4	5,721	12-76	16-81	35	48	288(18*16)-5852(76*77)
5	5,539	14-92	21-81	39	47	357(17*21)-6318(81*78)
6	5,858	12-74	17-81	32	47	216(12*18)-4891(73*67)
7	6,097	12-96	14-80	34	46	266(19*14)-6084(78*78)
8	5,695	13-78	17-82	34	48	247(13*19)-5040(70*72)
9	5,813	13-71	20-82	32	50	374(17*22)-5025(67*75)

6.3.3 Experimental Results of the Small Database

As we mentioned in Chapter 5, we constructed a small database with 181 images. Each image was mis-recognized in at least two different sizes, i.e. they are the most difficult ones.

According to the following table (Table 14), we conclude that enlarging images helps to increase the recognition rate.

Table 14. Substitution numbers of a smaller database with different normalization sources

Normalization Size	20*20	22*22	24*24	26*26	28*28	30*30
Normalized from MNIST	76	91	93	77	84	69
Normalized from originals	70	87	63	55	57	53

Normalizing images from the originals has a better performance than normalizing images from MNIST (already normalized once) when the images are normalized to the same sizes (Figure 24).

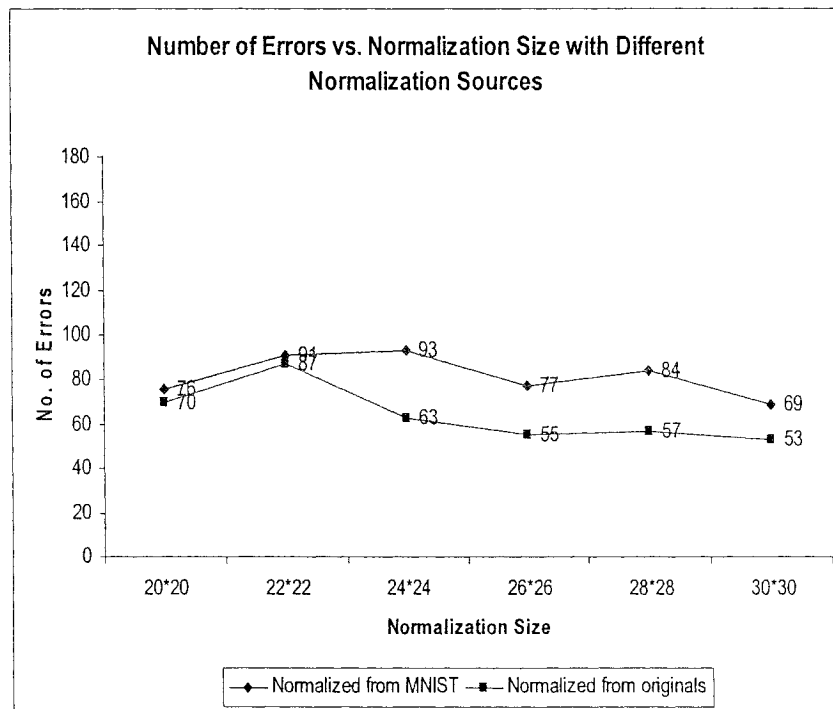


Figure 24. Number of errors in the small normalized database from different sources

Chapter 7

Conclusion

In this chapter, we summarize the contribution of this thesis and present some future work in this research direction. In this thesis, we not only proposed an effective hybrid Multiple Classifier System, but through error analysis we also investigated the relationship between the performance of handwritten numeral recognition systems and size resolution. Thus, my recommendations for future work are based on two facets.

7.1 Summary

The basic concept of this thesis is to construct an effective hybrid MCS (HMCS) of handwritten numeral recognition and to analyze factors that cause the errors in HMCS. The goal of the HMCS is to increase the reliability of the handwritten numeral recognition system while maintaining a reasonable recognition rate, which satisfies the requirement of some financial document processing systems. In error analysis of the HMCS, we focused mainly on the role of size normalization in the recognition of handwritten numerals.

The proposed HMCS integrates the cooperation (serial topology) and combination (parallel topology) of three classifiers: SVM, MQDF, and LeNet-5.

In cooperation, three measurements – First Rank Measurement (FRM), Differential Measurement (DM) and Probability Measurement (PM) are defined for their rejection options on different types of classifiers. As proven by the experiments, DM is an effective method to measure the rejected patterns when the outputs are numerical scores that contain the values of an arbitrary discriminant (MQDF) or distances to margins (SVM). However, if the outputs of classifiers are the distances to prototypes (LeNet-5), PM is a better method. As DM performs better than FRM in SVM and MQDF, FRM is not applied in cooperation of three classifiers.

In combination, Weighted Borda Count (WBC) at the rank level, which reflects *confidence* and *preference* of different ranks in different classes with different classifiers, is applied. We assigned a higher *confidence* to the classes at the first rank in each classifier, and we treated each class in each classifier differently based on their performance to show *preference*. According to the experiments, we found that WBC performs better than Majority Vote and BC.

After we integrated cooperation and combination on multiple classifiers, the final recognition rate of this hybrid system ranged from 95.54% to 99.11% with a reliability rate of 99.93% to 99.11%. Hence, we conclude that the proposed system has successfully achieved a high reliability while maintaining a reasonable recognition rate.

Although this HMCS has a high reliability rate while maintaining a reasonable recognition rate, analyzing error factors is vital to better performance. In error analysis, we focused mainly on the role of size normalization on the recognition of handwritten numerals.

The experimental results indicate that enlarging normalization size from $20 * 20$ to bigger sizes (e.g. $26*26$) is helpful in improving the recognition rate. As images in MNIST have already been normalized, normalizing them to a bigger size is the second source of distortion of the originals. Even though we distort (normalize) images twice, the recognition rates still rise. This suggests that if we normalize the image to a bigger size than $20 * 20$ from the originals, the recognition rate of the entire system will rise because we only need to normalize images from the originals once instead of twice.

Hence, we constructed a small database with originals. As it is impossible to find all the original images of the test set in MNIST, we resorted to the most difficult images, which are misrecognized in at least two different sizes, to construct the small database. Template matching with some constraints was applied to match the images in NIST and MNIST. We saw that normalizing the original data to a size larger than $20 * 20$ in MNIST increases the recognition rate. Therefore, we conclude that the performance of handwritten numeric recognition systems deteriorates dramatically due to low size resolution.

7.2 Future Research

This HMCS is not only useful for the recognition of handwritten numerals, but also for various application areas of pattern recognition (e.g. signature recognition, fingerprint recognition, face recognition, bioinformatics, etc.). Although several models and measurements have been proposed, the work is far from finished, and future research may include the following challenging problems:

1. In error analysis of this system, we know that images normalized to a larger size produced a higher recognition rate. However, enlarging images requires a higher computational cost, both in space and time. In the future, we can consider enlarging images partially instead of using the entire database as an optimal solution, e.g. mainly those images where the classifier does not have a high recognition confidence.
2. In this HMCS, although it includes cooperation and combination, its structure is not dynamic. Optimizing the model, which may integrate a selection of multiple classifiers, should be taken into consideration in the future.
3. Although DM and PM have been effectively defined for their rejection options in cooperation in this HMCS, their performance in other classifiers has not been examined. Meanwhile, other measurements may be designed for the current three classifiers. In the future, we may conduct more research on measurements of the rejection option.
4. In combination, weights, which reflect confidence and preference of each classifier in each class, are defined in this thesis. These weights are based on a statistical point of view. This means that a large enough and representative learning data set should be provided for future research. Therefore, the key issue to successfully apply this method is to construct a representative training data base, which cannot be guaranteed in this HMCS. Therefore, partial selection of training data may be considered.
5. In error analysis, we have conducted experiments to investigate its effects and have found that the performance of handwritten numeric recognition systems

deteriorates dramatically due to low size resolution. However, other factors may also reduce recognition rates. In the future, these factors (e.g. choosing different features, changing the space resolution of gradient features, etc.) may be taken into consideration.

References

1. J. X. Dong, A. Krzyzak, and C. Y. Suen, "A fast SVM training algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, No. 3, 2003, pp. 367 – 384.
2. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, No. 11, November 1998, pp. 2278 – 2324.
3. F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, No. 1, January 1987, pp. 149 – 153.
4. C. Y. Suen, C. Nadal, and R. Legault, T. A. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," *Proc. IEEE*, vol. 80, No. 7, 1992, pp. 1162 – 1180.
5. C. Y. Suen, C. Nadal, T. A. Mai, R. Legault, and L. Lam, "Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts," in C. Y. Suen (ed.), *Frontiers in Handwritten Recognition*, Montreal, Concordia University, 1990, pp. 131 – 143.
6. L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Systems, Man and Cybernetics*, vol. 22, No. 3, 1992, pp. 418 – 435.
7. F. Kimura and M. Shridhar, "Handwritten Numeral Recognition Based on Multiple Algorithm," *Pattern Recognition*, vol. 24, No. 10, 1991, pp. 969 – 983.
8. S. B. Cho and J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Trans. Systems, Man and Cybernetics*, vol. 25, No. 2, 1995, pp. 380 – 384.
9. Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for recognition of unconstrained handwritten numerals," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, No. 1, 1995, pp. 90 – 94.
10. V. Gunes, M. Menard, P. Loonis, and S. Petit-renaud, "Combination, cooperation and selection of classifiers: A state of the art," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, No. 8, 2003, pp. 1303 – 1324.
11. G. Fumera and F. Roli, "Analysis of error-reject trade-off in linearly combined multiple classifiers," *Pattern Recognition*, vol. 37, No. 6, 2004, pp. 1245– 1265.

12. J. C. Borda, "Memoire sur les Elections au Scrutin," *Histoire de l'Academie Royale des Sciences*, 1781.
13. K. Arrow, *Social Choice and Individual Values*. Wiley, New York, 1963.
14. K. Tumer and J. Ghosh, "Robust Combining of Disparate Classifiers through Order Statistics," *Pattern Analysis and Applications*, vol.3, No.4, 2000, pp. 189 – 200.
15. T. K. Ho, "A theory of multiple classifier systems and its application to visual word recognition," PhD thesis, State University of New York at Buffalo, May 1992.
16. A. Agesti, *An Introduction to Categorical Data Analysis*. Wiley, 1996.
17. A. Britto-Jr., R. Sabourin, E. Lethelier, F. Bortolozzi, and C. Y. Suen, "Improvement handwritten numeral string recognition by slant normalization and contextual information," *Proceedings of the 7th IWFHR*, Amsterdam-Netherlands, September 2000, pp. 323 – 332.
18. E. Kavallieratou, Fakotakis N. and G. Kokkinakis, "Slant estimation algorithm for OCR systems," *Pattern Recognition*, vol. 34, No. 12, 2001, pp. 2515 – 2522.
19. T. Y. Zhang, C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, No. 3, March 1984, pp. 236 – 239.
20. J. Ghosh, "Multiclassifier Systems: Back to the future," *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, vol. 2364, 2002, pp. 1 – 15.
21. M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," *Pattern Recognition*, vol. 35, No. 10, 2002, pp. 2051 – 2059.
22. L. Lam, "Classifier combinations: Implementations and theoretical issues," *Multiple Classifier Systems, First International Workshop, MCS 2000*, Cagliari, Italy, June 2000, pp. 77 – 86.
23. N. C. de Condorcet. *Essai sur l'application de l'analysis a la probabillite des decisions rendues a la pluralite des voix*. Imprimerie Royale, Paris, 1785.
24. L. G. Roberts, "Machine perception of three-dimensional solids," *Optical and Electro-Optical Information Processing*, J. T. Tippet, Ed. MIT Press, Cambridge, MA, 1965.

25. S. Lee and Y. Kim, "Multiresolution recognition of handwritten numerals with wavelet transform and multilayer cluster neural network," *Proceeding of the 3rd International Conference on Document Analysis and Recognition*, vol. 2, Montreal, Canada, 1995, pp. 1010-1014.
26. P. Zhang, D. Bui, and C. Y. Suen, "Extraction of Hybrid Complex Wavelet Features for the Verification of Handwritten Numerals", *Proceedings of the 9th International Workshop on Frontiers of Handwriting Recognition*, Tokyo, Japan, 2004, pp. 347 – 352.
27. S. Battiato, G. Gallo, and F. Stanco, "A New Edge-Adaptive Algorithm for Zooming of Digital Images", *Proceedings of IASTED Signal Processing and Communications SPC 2000*, Marbella, Spain, 2000, pp. 144-149.
28. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, Winter 1989, pp. 541 – 551.
29. Y. LeCun, I. Kanter, and S. Solla, "Eigenvalues of covariance matrices: application to neural-network learning," *Physical Review Letters*, vol. 66, no. 18, May 1991, pp.2396 – 2399.
30. R. L. Winkler, "Combining probability distributions from dependent information sources," *Management Science*, vol. 27, no. 4, April 1981, pp. 479 – 488.
31. P. A. Morris, "Decision analysis expert use," *Management Science*, vol. 20, no. 9, 1974, pp. 1233 – 1241.
32. H. V. Roberts, "Probabilistic prediction," *Journal of American Statistical Society*, vol. 60, 1965, pp. 50 – 52.
33. R. L. Winker, "Probabilistic prediction: Some experimental results," *Journal of American Statistical Society*, vol. 66, no. 336, 1971, pp.675 – 685.
34. M. H. DeGroot, "Reaching a consensus," *Journal of American Statistical Society*, vol. 69, 1974, pp. 118 – 121.
35. E. Eisenberg and D. Gale, "Consensus of subjective probabilities: the pari-mutuel method," *Annals of Mathematical Statistics*, vol. 39, 1959, pp. 165 – 168.
36. M. Stone, "The opinion pool," *Annals of Mathematical Statistics*, vol. 32, 1961, pp. 1339 – 1342.
37. S. S. Keerthi, E. G. Gilbert, "Convergence of a generalized SMO algorithm for SVM classifier design," *Machine Learning*, vol. 46, 2002, pp. 351- 360.

38. Y. S. Huang, "Combination of multiple classifier for the recognition of totally unconstrained handwritten numerals," PhD thesis, Concordia University, 1994.
39. M. A. Kraaijveld, "An experimental comparison of non-parametric classifiers for time-constrained classification tasks," *Proceedings of International Conference on Pattern Recognition*, 1998, pp. 428 – 435.
40. R. O. Duda and P. E. Hart, "Pattern classification and scene analysis," *John Wiley & Sons*, New York, 1973.
41. G. Nagy, "State of the art in pattern recognition," *Proceedings of the IEEE*, vol. 56, no. 836, 1968, pp. 62.
42. J. Schurmann, "Pattern Classification – *A unified view of statistical and neural approaches*," Wiley-Interscience, New York, 1996.
43. K. S. Fu, "Syntactic Pattern Recognition and Applications," Prentice-Hall, 1982.
44. S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on System, Man, and Cybernetics*, vol. 21, no. 3, 1991, pp. 660 – 674.
45. F. Esposito, D. Malerba, and G. Semeraro, "A comparative analysis of methods for pruning decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, May 1997, pp. 476 – 491.
46. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," Chapman & Hall, New York, 1984.
47. J. R. Quinlan, "C4.5: Programming for machine learning," Morgan Kaufmann, San Mateo, California, 1993.
48. T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, August 1998, pp. 832 – 844.
49. 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.
50. C. L. Liu, H. Sako, and H. Fujisawa, "Discriminative learning quadratic discriminant function for handwriting recognition," *IEEE Transactions on Neural Network*, vol. 15, No. 2, March 2004, pp. 430 – 444.
51. R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis," Wesley, New York, 1973, pp. 67.

52. J. Cai and Z.-Q. Liu, "Integration of structural and statistical information for unconstrained handwritten numeral recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, March 1999, pp. 263 – 270.
53. 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.
54. J. M. Tan, "Automatic verification of the outputs of multiple classifiers for unconstrained handwritten numerals," Master's thesis, Concordia University, 2004.
55. C. Y. Suen and J. M. Tan, "Analysis of errors of handwritten digits made by a multitude of classifiers," *Pattern Recognition Letters*, vol. 26, no. 3, 2005, pp. 369 – 379.
56. C. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, 2004, pp. 265 – 279.
57. J. Park, V. Govindaraju, and S. N. Srihari, "OCR in a Hierarchical Feature Space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no.4, 2000, pp. 400 – 407.

Appendices

Appendix I:

Index of images in MNIST, which are substituted in different sizes

181 images substituted in at least 2 different sizes:											
582	924	1901	3762	4176	4224	4497	4761	6571	6883	9530	9729
247	1112	1530	1871	1878	2130	2135	2182	2462	4271	4284	4360
5654	6555	6572	6625	8325	8527	9587	9664	9792	447	448	583
947	1319	1393	1681	1709	1737	2040	2053	2952	3369	3422	3902
4201	4807	5955	6559	6576	6597	9505	9634	412	659	1039	1242
2293	2326	2414	2654	2939	3021	3073	4306	4369	4924	9850	674
691	716	882	938	1226	1621	2447	2454	4063	4578	4823	5972
6558	7921	8246	8316	8408	29	320	449	495	726	846	1014
1290	1299	1790	2052	2877	2927	3060	3225	3821	3869	3941	4065
4163	4571	4575	4838	5176	5735	5937	6560	6561	7845	8059	8550
9024	9692	18	92	149	321	358	445	684	740	1033	1182
1364	1500	1549	1982	2035	2070	2105	2118	2189	2387	2405	2648
2945	3005	3100	3316	3365	3559	3604	3702	3780	3806	4078	4265
4433	4723	4740	4874	4879	4956	5835	5888	5973	6011	6755	6783
6895	7430	7853	7862	8094	9009	9015	9620	9642	9700	9839	9944
9980											
231 images substituted in 1 size:											
63	73	115	158	184	209	241	290	340	404	406	435
478	522	551	578	593	646	696	707	810	813	844	894
936	1002	1003	1022	1044	1068	1114	1212	1247	1256	1260	1270
1296	1403	1414	1444	1454	1581	1611	1634	1637	1641	1695	1701
1721	1754	1769	1915	1938	1941	1952	1984	1994	2016	2043	2044
2109	2129	2198	2276	2309	2314	2327	2380	2382	2406	2437	2488
2514	2523	2532	2671	2678	2770	2800	2850	2863	2921	2953	3030
3186	3250	3329	3330	3344	3377	3381	3558	3475	3503	3534	3629
3672	3726	3751	3756	3767	3772	3778	3785	3817	3834	3893	3964
3985	4007	4017	4018	4086	4156	4167	4289	4300	4317	4325	4419
4425	4437	4443	4451	4505	4536	4548	4551	4594	4625	4690	4696
4699	4731	4755	4759	4783	4924	4860	4880	4943	5188	5190	5261
5268	5288	5331	5450	5600	5601	5626	5634	5734	5769	5802	5866
5939	6042	6065	6081	6091	6093	6137	6157	6166	6172	6173	6370
6532	6553	6581	6592	6599	6632	6651	6662	6765	6806	6847	6988
7081	7208	7216	7259	7457	7505	7552	7774	7857	7902	7915	8062
8065	8071	8081	8092	8095	8112	8217	8254	8263	8277	8279	8295
8310	8320	8377	8382	8453	8508	8607	9019	9022	9175	9225	9227
9255	9280	9427	9698	9779	9811	9847	9849	9867	9874	9875	9892
9904	9905	9943									

Appendix II:

Index of images in NIST and MNIST, which are substituted in at least 2 different sizes













MNIST	NIST	label	MNIST	NIST	label	MNIST	NIST	label	MNIST	NIST	label
582	046256	8	3422	020336	6	495	039198	8	2105	012373	3
924	029251	2	3902	054485	5	726	020157	7	2118	051740	6
1901	021018	9	4201	053927	1	846	042415	7	2189	014970	9
3762	013627	6	4807	008282	8	1014	040296	6	2387	006254	9
4176	056443	2	5955	019873	3	1290	035601	3	2405	039487	3
4224	021744	9	6559	099864	4	1299	050578	5	2648	016183	9
4497	006575	8	6576	080528	7	1790	022650	2	2945	033751	3
4761	053812	9	6597	229350	0	2052	019093	8	3005	047162	9
6571	052859	9	9505	320036	7	2877	054565	4	3100	019049	5
6883	273430	1	9634	116274	0	2927	039371	3	3316	055336	7
9530	028766	9	412	003422	5	3060	053059	9	3365	013295	6
9729	024669	5	659	052206	2	3225	021785	7	3559	049346	8
247	055304	4	1039	006733	7	3821	047709	9	3604	056385	7
1112	055964	4	1242	057181	4	3869	007685	9	3702	006293	5
1530	015234	8	2293	041915	9	3941	023157	4	3780	022683	4
1871	048358	2	2326	006402	0	4065	026195	0	3806	053463	5
1878	027001	8	2414	034953	9	4163	037757	9	4078	000955	9
2130	035302	4	2654	022815	6	4571	033342	6	4265	030829	4
2135	002621	6	2939	036004	9	4575	003941	4	4433	028239	7
2182	033186	1	3021	048695	2	4838	040366	6	4723	048716	2
2462	003019	2	3073	057812	1	5176	089023	8	4740	034875	3
4271	001508	5	4306	037004	3	5735	252635	5	4874	006178	9
4284	035992	9	4369	014734	9	5937	128333	5	4879	004154	8
4360	035247	5	4924	015975	1	6560	070395	9	4956	013023	8
5654	321845	7	9850	054117	0	6561	169043	7	5835	061379	7
6555	103320	8	674	007587	5	7845	333833	8	5888	223432	4
6572	156873	1	691	051347	8	8059	044268	2	5973	160917	3
6625	233623	8	716	007842	1	8550	327660	2	6011	237875	3
8325	006132	0	882	050809	9	9024	008312	7	6755	229704	8
8527	305783	4	938	012575	3	9692	237236	9	6783	152681	1
9587	282157	9	1226	017226	7	18	050034	3	6895	047501	9
9664	067322	2	1621	010030	0	92	054071	9	7430	007210	5
9792	193498	4	2447	047283	4	149	036431	2	7853	106810	8
447	007832	4	2454	049033	6	321	004169	2	7862	256992	8
448	006996	9	4063	002887	6	358	014554	7	8094	162800	2
583	031661	2	4578	005391	7	445	058271	6	9009	130565	7
947	033897	8	4823	027861	9	684	032148	7	9015	246077	7
1319	049789	8	5972	037645	5	740	013822	4	9620	041283	9
1393	029768	5	6558	165654	6	1033	016353	8	9642	226879	9
1681	031153	3	7921	299972	8	1182	044865	6	9700	204103	2
1709	010039	9	8246	109948	3	1364	054498	8	9839	065932	4
1737	002324	5	8316	244183	7	1500	013856	7	9944	252626	3
2040	000271	5	8408	286072	8	1549	057788	4	9980	186239	2
2053	014586	4	29	027709	1	1982	057765	6			
2952	014950	3	320	049672	9	2035	055771	5			
3369	036042	9	449	002705	3	2070	054427	7			





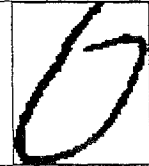





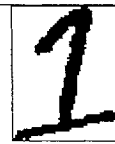





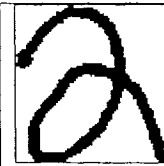

Appendix III:










Patterns with maximum or minimum width, height or area in NIST SD 19: Training Set and Test Set





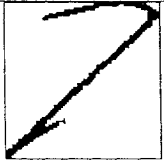













Training Set						
0						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	83	14	85	13	7055	240
Sequence No.	334375	005562	145759	001078	334375	008316
Images						
1						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	82	17	73	3	5256	75
Sequence No.	079237	033581	149129	054707	149129	054707
Images						
2						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	80	18	123	13	8856	270
Sequence No.	103383	150790	102303	189076	102303	150790
Images						
3						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	79	18	75	13	5254	340
Sequence No.	034993	053431	058285	037134	020261	051422
Images						

4						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	81	16	76	12	5852	288
Sequence No.	029576	005009	056349	032153	056349	005009
Images						
5						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	18	21	92	14	6318	357
Sequence No.	007608	035235	000546	050428	031831	050156
Images						
6						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	81	17	74	12	4891	216
Sequence No.	057417	051581	010582	010989	049163	051004
Images						
7						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	80	14	96	12	6084	266
Sequence No.	051971	049409	003088	001544	031136	049409
Images						
8						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	82	17	78	13	5040	247
Sequence No.	013015	055408	000894	043742	025053	043742

Images						
9						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	82	20	71	13	5025	374
Sequence No.	052406	036191	003100	029413	011900	029577
Images						

Test Set						
0						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	79	15	79	13	5372	208
Sequence No.	037909	000597	033649	000080	037909	027013
Images						
1						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	81	18	60	3	3432	108
Sequence No.	010767	050564	032519	003352	044100	003352
Images						
2						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	79	15	95	14	5775	210
Sequence No.	050547	036186	039341	017370	023766	036186
Images						

3						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	79	18	75	13	5254	340
Sequence No.	034993	053431	058285	037134	020261	051422
Images						
4						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	81	16	76	12	5852	288
Sequence No.	029576	005009	056349	032153	056349	005009
Images						
5						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	18	21	92	14	6318	357
Sequence No.	007608	035235	000546	050428	031831	050156
Images						
6						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	81	17	74	12	4891	216
Sequence No.	057417	051581	010582	010989	049163	051004
Images						
7						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	80	14	96	12	6084	266
Sequence No.	051971	049409	003088	001544	031136	049409

Images						
8						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	82	17	78	13	5040	247
Sequence No.	013015	055408	000894	043742	025053	043742
Images						
9						
Characters	Max Height	Min Height	Max Width	Min Width	Max Area	Min Area
Size	82	20	71	13	5025	374
Sequence No.	052406	036191	003100	029413	011900	029577
Images						

Appendix IV:

Original images of patterns in NIST, which are incorrectly recognized in MNIST with HMCS

0	0	0	0	0	0	0
1	1	1	1	1	1	1
1	0	1	2	3	3	3
3	3	3	3	3	4	4
4	4	4	4	4	5	5
5	5	5	6	6	6	6
6	6	6	8	6	7	7
7	7	7	7	7	7	7
7	7	7	8	8	8	8
8	8	8	8	8	9	8
4	9	5	3	1	9	9

9	9	9	9	9	9	9
4	9	9	9	9	9	9
8	9	9				

Appendix V:

Total Probability Theorem:

Given n mutually exclusive events A_1, \dots, A_n whose probabilities sum to unity, then

$$P(B) = P(B | A_1)P(A_1) + \dots + P(B | A_n)P(A_n)$$

where B is an arbitrary event, and $P(B | A_i)$ is the conditional probability of B assuming A_i .