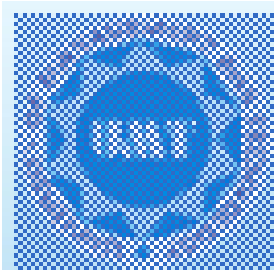
FINAL YEAR PROJECT REPORT

A COMPARISION OF DIGIT RECOGNITION TECNIQUES



Project Advisor

Mirza Mubasher Baig

Submitted By

AHMAD FAHEEM ABID	**022224-028**
SAIMA ASGHAR	**040420-066**

Department of Computer Sciences

School of Science and Technology

University of Management and Technology

A COMPARISION OF DIGIT RECOGNITION TECNIQUES

Project Report Submitted to the Department of Computer Science, University of Management and Technology

In partial fulfillment of the requirements for the degree of

Bachelor of Science

In

Computer Science

By

AHMAD FAHEEM ABID **022224028 SAIMA ASGHAR** **040420-066**

---------- **----------**

Advisor **Head of Department**

A COMPARISION OF DIGIT RECOGNITION TECNIQUES

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction to Handwritten Digit Recognition

Optical Character Recognition (OCR)

Optical Character recognition (OCR) has been one of the most active areas of research during the last three decades [1-4]. While automatic recognition of computer printed characters is extremely successful, and many commercial products are available for machine printed character recognition. Human are very good at recognizing the written characters and can attain 100% accuracy on neatly written characters. But the automatic recognition of human written characters is still a challenging task and we are far from perfect automatic recognition of human written characters. The task is difficult because of the unlimited variation in the writing style of people and because of the noise introduced by the digitalizing process.

The automatic recognition of handwritten characters has substantial benefits including automated processing of handwritten documents, automatic data entry, searching within images, man-machine interface etc [5]. The automated processing of handwritten documents can make this immense source of information useful while machine interface open up a whole new way of communication between machine and the humans.

The recognition of handwritten character is broadly categorized into **online** and **offline** character recognition [6 PAMI]. **Online** character recognition refers to the recognition problem when complete pen stroke information along with the digital image of the character is available to the recognition routines. This additional information of pen movement greatly simplifies the task and the recognition problem might be posed as a sequence recognition problem. In such a case the Hidden Markov Model (HMM) is successfully applied for recognizing the characters[7-9].

Offline recognition of handwritten characters only uses pre-captured images of the character and no information about the pen stroke is available. Methods based on computing similarity between different characters are used to classify a given character. Many different methods

including neural networks, support vector machine, self organizing maps etc have been successfully applied to classify a given character. Researchers have suggested various representations of characters and similarity computing methods to classify characters[10-15].

Handwritten Digit recognition

Handwritten Digit recognition is a subtask of handwritten character recognition with many useful applications including automatic sorting of mail using postal code [16], automatic processing of bank checks [17] etc.

Handwritten Digit recognition Dataset

We performed experiments of the MINIST [18] database consisting of 60000 training digit images and 10000 test digit images. The digit images have been normalized and centered in 28x28 image therefore reducing significant preprocessing efforts.

Previous results on the Digit Recognition Dataset

Table 1 [19] shows the results of various classification algorithms performed on this dataset. It is interesting to note that very high recognition rates have been attained on this dataset by using complex features.

In this project we compare two approaches for the recognition of offline handwritten digits. The methods include the HMM and a new way of using template images for digit recognition. Both the approaches do not yield optimal results and many methods proposed by researchers give significantly better results. The results obtained by using template images, chapter 3, are significantly better than the use of HMM as described in chapter 2. The use of templates yield extremely efficient way as no features are computed and only few basic arithmetic operations are used to classify the digits. This approach uses pixel based matching and hence it is suggested (without any experiments reported in this report) that matching incorporating localized and global features might significantly improve the accuracy of this method without degrading it efficiency. Another important property of the new way of template matching is that it is very stable in the sense that very large reduction in the amount of training data does not have any significant impact on its accuracy.

The remaining report is divided into two chapters 2 and 3. Chapter 2 presents the basic theory of HMM and results obtained when HMM are used to classify handwritten digits. In chapter 3 the method of generating the template images and their use and results for handwritten digit recognition are presented.

Table 1: Recognition rate of different classifiers

METHODS	% OF ERROR NUMBER
Linear classifier	12.0

Linear classifier (nearest neighbor-NN)	8.4
Pairwise linear classifier	7.6
K-NN, Euclidean	5.0
2-layer NN, 300 hidden units (HU) (28x28-300-10)	4.7
2-layer NN, 1000 HU (28x28-1000-10)	4.5
2-layer NN, 1000 HU, [distortions] (28x28-1000-10)	3.8
2-layer NN, 300 HU, [distortions] (28x28-300-10)	3.6
1000 RBF (Radial Basis Function) + linear classifier	3.6
40 PCA (Principal Component Analysis) + quadratic classifier	3.3
3-layer NN, 300+100 HU (28x28-300-100-10)	3.05
3-layer NN, 500+150 HU (28x28-500-150-10)	2.95
3-layer NN, 300+100 HU, [distortions] (28x28-300-100-10)	2.5
3-layer NN, 500+150 HU, [distortions] (28x28-500-150-10)	2.45
K-NN Euclidean, deslant	2.4
LeNet-1 [16x16]	1.7
2-layer NN, 300 HU, [deslant] (20x20-300-10)	1.6
K-NN, Tangent Distance, [16x16]	1.1
SVM (Support Vector Machine) poly 4	1.1
LeNet-4	1.1
LeNet-4 / K-NN	1.1
LeNet-4 / Local	1.1
Reduced Set SVM poly 5	1.0
LeNet-5	0.95
Virtual SVM poly 9 [distortions]	0.8
LeNet-5 [distortions]	0.8
Boosted LeNet-4 [distortions]	0.7