Generalized Solution to Fetch Information from the Cheques of Different Banks

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology
In
Computer Science and Applications

Submitted By
Nikita Jain
(Roll No. 601003030)

Under the supervision of
Dr. Rajiv Kumar
Assistant Professor

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS
THAPAR UNIVERSITY
PATIALA – 147004

June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Generalized solution to fetch information from the cheques of different banks*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in *School of Mathematics and Computer Applications* Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Rajiv Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature:
(Nikita Jain)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Rajiv Kumar)
Assistant Professor,
SMCA

Countersigned by

(Dr. S. S. Bhatia)
Professor and Head
School of Mathematics and Computer Applications
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to **Dr. Rajiv Kumar, Assistant Professor,** School of Mathematics and Computer Applications, Thapar University, Patiala for introducing me to document image processing and for all his guidance and support. This thesis work was enabled and sustained by his vision and ideas. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. His patience and support helped me overcome many crisis situations and finish this dissertation.

I am also thankful to our **Head of the Department, Dr. S. S. Bhatia** as well as **PG Coordinator, Mr. Singara Singh, Assistant Professor,** School of Mathematics and Computer Applications, entire faculty and staff of School of Mathematics and Computer Applications and then friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I thank all those who have contributed directly or indirectly to this work.

Lastly, I would also like to thank my parents for their years of unyielding love and encourage. They have always wanted the best for me and I admire their determination and sacrifice.

Nikita Jain
(601003030)

ABSTRACT

As the population is growing, bank transactions which involves the use of bank cheques is increasing rapidly. A great deal of work has been done on automatic processing of bank cheques like automatic extraction of items from cheque, signature verification, recognition of date from the cheque, amount filled by the user etc. Till now, the MICR code processing is performed using special type of scanner which reads the magnetic ink particles. These machines are very precisely designed which make them very expensive and requires constant maintenance to maintain their reliability. Moreover the MICR code is written with the special type of magnetic ink which is very costly. Any damage to MICR code characters will hamper its performance. It has been analyzed that no other approach has been developed for the processing of MICR code, so an effort has been made by the authors to develop a novel approach for processing the MICR code. The concept of OCR process serves as the basis for this approach.

This approach automatically segments and recognizes the MICR code present at the bottom of the cheque. After analyzing number of cheques, it was observed that dimensions of digits were varying across the cheques. This made it difficult to find the pattern which helps to recognize the digit. In order to overcome this problem an approach was followed to manipulate the size of each digit to a specific dimension using the Resizing concept. MICR band which is present at the bottom of the cheque contains information. One of the information is MICR code. MICR code gives the important information about the name of the city, bank name and branch name of the bank who has issued that particular cheque. The uniqueness of our approach lies in the fact that it doesn’t necessitate any prior information and requires minimum human intervention. The system performance is quite promising on a large dataset of real cheque images.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The storage of scanned documents have to be bulky in size and many processing applications as searching for a content, editing, maintenance are either hard or impossible. Such documents require human beings to process them manually, for example, postman's manual processing for recognition and sorting of postal addresses and zip code. Optical Character Recognition (OCR) is the mechanical or electronic translation / reading of images of handwritten, typewritten or printed text (usually captured via scanner) into machine editable text. OCR is a field of research in pattern recognition, artificial intelligence and machine vision. An OCR system enables you to take a book or a magazine article, feed it directly into an electronic computer file, and then edit the file using a word processor. All OCR systems include an optical scanner for reading text, and sophisticated software for analyzing images. Most OCR systems use a combination of hardware (specialized circuit boards) and software to recognize characters, although some inexpensive systems do it entirely through software. Advanced roman OCR systems can read text in large variety of fonts, but they still have difficulty with handwritten text.

1.1 HISTORY OF OPTICAL CHARACTER RECOGNITION (OCR)

To understand the phenomena described in the above section, one is to look at the history of OCR and its development, recognition methods, computer technologies, and the differences between humans and machines. It is always fascinating to be able to find ways of enabling a computer to mimic human functions, like the ability to read, to write, to see things, and so on. According to Bunke and Wang [1997], OCR research and development can be traced back to the early 1950s, when scientists tried to capture the images of characters and texts, first by mechanical and optical means of rotating disks and photomultiplier, flying spot scanner with a cathode ray tube lens, followed by photocells and arrays of them. At first, the scanning operation was slow and one line of characters could be digitized at a time by moving the scanner on the paper medium. Subsequently, the inventions of drum and flatbed scanners arrived, which extended scanning to the full page. Then, advances in digital integrated circuits brought photo arrays with higher density, faster transports for documents and higher speed in scanning and digital conversions.

According to Cheriet *et al*. [1986] these important improvements greatly accelerated the speed of character recognition and reduced the cost, and opened up the possibilities of processing a great variety of forms and documents. Throughout the 1960s and 1970s, new OCR applications sprang up in retail businesses, banks, hospitals, post offices, insurance, railroad, aircraft companies, newspaper publishers and many other industries. In parallel with these advances in hardware development, intensive research on character recognition was taking place in the research laboratories of both academic and industrial sectors.

According to Nagy *et al*. [1999] both recognition techniques and computers were not that powerful in the early days (1960s), OCR machines tended to make lots of errors when the print quality was poor, caused either by wide variations in type fonts and roughness of the surface of the paper or by the cotton ribbons of the typewriters. To make OCR work efficiently and economically, there was a big push from OCR manufacturers and suppliers toward the standardization of print fonts, paper and ink qualities for OCR applications. New fonts such as OCRA and OCRB were designed in the1970s by the American National Standards Institute (ANSI) and the European Computer Manufacturers Association (ECMA), respectively. These special fonts were quickly adopted by the International Organization for Standardization (ISO) to facilitate the recognition process. As a result, very high recognition rates became achievable at high speed and at reasonable costs. Such accomplishments also brought better printing qualities of data and paper for practical applications. Actually, they completely revolutionized the data input industry and eliminated the jobs of thousands of keypunch operators who were doing the really mundane work of keying data into the computer. Character recognition is used as an umbrella term, which covers all types of machine recognition of characters in various application domains. It is processing by machine of text based input patterns to produce meaningful outputs. The input may come from online devices like tablets, stylus based devices or offline devices like scanners. Output may be a sequence of symbols like ‘Y’, ‘E’ ,‘S’ or a date on cheque like ‘Nov 14, 2011’ or validation result of a signature. As the input may come from different sources, they may be categorized broadly as offline and online and are explained in the next section.

1.2 TYPES OF CHARACTER RECOGNITION

Character recognition can be categorized into following two parts:-

- Online Character Recognition
- Offline Character Recognition

Offline character recognition refers to the process of recognizing words that have been scanned from a surface (such as a sheet of paper) and are stored digitally in gray scale format. After being stored, it is conventional to perform further processing to allow superior recognition.

In case of online character recognition, the handwriting is captured and stored in digital form via different means. Online handwritten character recognition deals with automatic conversion of characters, which are written on a special digitizer, tablet personal computer (PC) or personal digital assistant (PDA) where a sensor picks up the pen tip movements as well as pen up or pen down switching. Usually, a special pen is used in conjunction with an electronic surface. As the pen moves across the surface, the two dimensional coordinates of successive points are represented as a function of time and are stored in order.

It is generally accepted that the online method of recognizing handwritten text has achieved better results than its offline counterpart. This may be attributed to the fact that more information may be captured in the online case such as the direction, speed and the order of strokes of the handwriting. The major difference between online and offline character recognition is that online character recognition has real time contextual information but offline data does not. This difference generates a significant divergence in processing architectures and methods.

Handwritten Character Recognition is more difficult to implement than printed character recognition due to diverse human handwriting styles and customs.

Now, since the document image has been acquired with the help of a scanner as this work deals with the offline character recognition, it is required to go through a lot of processes to get recognized. Such processes can be grouped into stages which are discussed in the next section.

1.3 STAGES OF OPTICAL CHARACTER RECOGNITION

The Optical Character Recognition process includes hierarchical tasks which are grouped in the stages of the character recognition as preprocessing, segmentation, recognition and post processing. In some methods, some of the stages are merged or omitted; in others a feedback mechanism is used to update the output of each stage depending on the field of application. The stages are depicted in the figure below:-

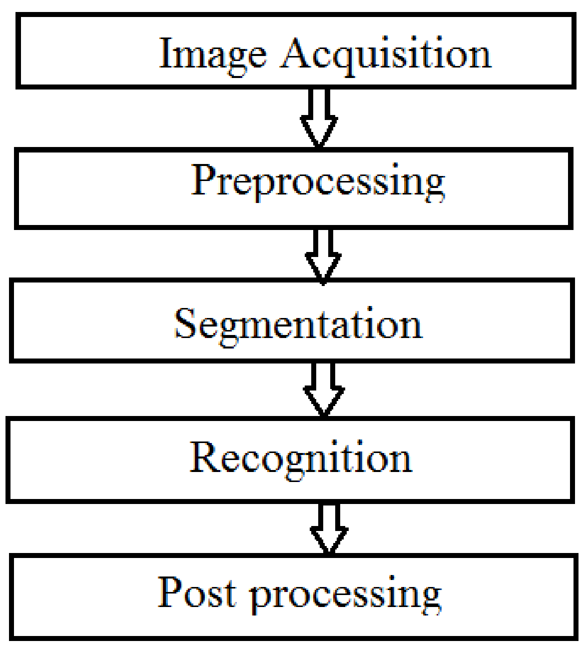


Figure 1.1 Stages in Optical Character Recognition

The various stages can be summarized as:-

- **PREPROCESSING**

  The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that is easy for the character recognition systems to operate accurately. It covers all functions to produce 'cleaned up' version of original image.

- **SEGMENTATION**

  The preprocessing stage yields a clean document in the sense that a sufficient amount of shape information, high compression, and low noise on a normalized image is obtained. The next stage is segmenting the document into its sub components. Segmentation is an important stage because the extent one can reach in separation of words, lines or characters directly affects the recognition rate of the script. There are two types of segmentation, external segmentation, which is the isolation of various writing units, such as paragraphs, sentences, or words and internal segmentation, which is the isolation of letters, especially in cursively written words.

- **RECOGNITION**

  The OCR process assigns a character image to a class by using a classification algorithm based on the features extracted and the relationships among the features. Since members of a character class are equivalent or similar in as much as they share defining attributes, the measurement of similarity, either explicitly or implicitly, is central to any classifier.

  Feature extraction is concerned with recovering the defining attributes obscured by imperfect measurements. To represent a character class, either a prototype or a set of samples must be known. The feature selection process attempts to recover the pattern attributes characteristic of each class. Global features, such as the number of holes in the character, the number of concavities in its outer contour, and the relative protrusion of character extremities, and local features, such as the relative positions of line endings, line crossovers, and corners are commonly used. The classification stage identifies each input character image by considering the detected features.

In the statistical classification approaches, character image patterns are represented by points in a multidimensional feature space. Each component of the feature space is a measurement or feature value, which is a random variable reflecting the inherent variability within and between classes. A classifier partitions the feature space into regions associated with each class, labeling an observed pattern according to the class region into which it falls.

- **POSTPROCESSING**

  The incorporation of context and shape information in all the stages of character recognition systems is necessary for meaningful improvements in recognition rates. This is done in the post processing stage with a feedback to the early stages of character recognition. The simplest way of incorporating the context information is the utilization of a dictionary for correcting the minor mistakes of the character Recognition systems. The basic idea is to spell check the character recognition output and provide some alternatives for the outputs of the recognizer that do not take place in the dictionary.

Presented work is related with preprocessing and segmentation. It is necessary to discuss these techniques in detail which are explained in the next section.

1.4 OBJECTIVES OF PREPROCESSING

The main objectives of preprocessing techniques to achieve them, as stated by Arica and Vural [2001], are explained here:-

- **NOISE REDUCTION**

  The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops, etc. The distortion, including local variations, rounding of corners, dilation, and erosion, is also a problem. The major objective of noise removal is to remove any unwanted bit patterns, which do not have any significance in the output. The most important reason to reduce noise is that extraneous features will otherwise cause subsequent errors in recognition. Another reason is that noise reduction reduces the size of the image file, and this in turn reduces the time required for subsequent processing and storage.

- **NORMALIZATION OF THE DATA**

  Normalization methods aim to remove the variations of the writing and obtain standardized data.

- **REDUCTION IN THE AMOUNT OF INFORMATION TO BE RETAINED**

  It is well known that classical image compression techniques transform the image from the space domain to domains, which are not suitable for recognition. Compression for character recognition requires space domain techniques for preserving the shape information.

The basic techniques used for size reduction are thresholding and thinning, can be explained as under:-

1.4.1 THRESHOLDING

In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray scale or color images as binary images by picking a threshold value.

The information is binary, the data, in the form of pixels with intensity values, are not likely to have only two levels, but instead a range of intensities. This may be due to non uniform printing or non uniform reflectance from the page, or a result of intensity transitions at the region edges that are located between foreground and background regions.

The objective in thresholding is to mark pixels that belong to true foreground regions with a single intensity (ON) and background regions with a different intensity (OFF).

However, for the many documents that have a wide range of background and object intensities, the fixed threshold level often does not yield images with clear separation between the foreground components and background. For instance, when a document is printed on differently colored paper or when the foreground components are faded due to photocopying, or when different scanners have different light levels, the best threshold value will also be different.

The techniques for thresholding can be categorized as global and local thresholding, as per Senior and Robinson [1998], are explained here:-

- **GLOBAL THRESHOLDING**

  If the pixel values of the components and those of the background are fairly consistent in their respective values over the entire image, then a single threshold value can be found for the image. This use of a single threshold for all image pixels is called global thresholding.

  It picks one threshold value for the entire document image which is often based on an estimation of the background level from the intensity histogram of the image. For an image with well differentiated foreground and background intensities, the histogram will have two distinct peaks. The valley between these peaks can be found as the minimum between two maxima and the intensity value there is chosen as the threshold that best separates the two peaks.

  The thresholding techniques are shown in the figure 1.2 with various choices of the threshold values for the technique. The low threshold gives an unclear image while the high threshold removes some of the important details from the image. Hence, the selection of good threshold is necessary for thresholding technique to achieve its objective.

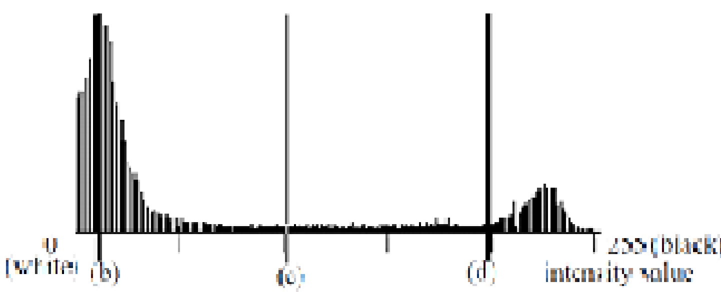


Figure 1.2 (a) Histogram of Original Gray-Scale Image

Figure 1.2 (b) Image Thresholded with Too Low Threshold Value

Figure 1.2 (c) Image Thresholded with Good Threshold Value

Figure 1.2 (d) Image Thresholded with Too High Threshold Value

- **LOCAL THRESHOLDING**

The images do not always contain well differentiated foreground and background intensities due to poor contrast and noise. Local thresholding is done by analyzing gray level intensities within local windows across the image to determine local thresholds. It uses different values for each pixel according to the local area information.

The main problem with this technique is the choice of window size. The chosen window size should be large enough to guarantee that a large enough number of background pixels are included to obtain a good estimate of average value, but not so large as to average over non uniform background intensities.

1.4.2 THINNING

The purpose of thinning is to reduce the image components to their essential information so that further analysis and recognition can be facilitated. For instance, the same words can be handwritten with different pens giving different stroke thicknesses, but the literal information of the words is the same. It provides a tremendous reduction in data size, thinning extracts the shape and context information of the characters. It refers to the process of reducing the width of a line like object from many pixels wide to just single pixel. This process can remove irregularities in letters and in turn, makes the recognition algorithm simpler because they only have to operate on a character stroke, which is one pixel wide.

The basic iterative thinning operation is to examine each pixel in an image within the context of its neighborhood region of at least 3×3 pixels and to peel the region boundaries, one pixel layer at a time, until the regions have been reduced to thin lines.

1.5 SEGMENTATION

Segmentation is the most challenging part of developing recognition software. Some systems force segmentation by providing individual boxes to write in instead of one flowing input section.

A process that seeks to decompose an image of a sequence of characters into sub images of individual symbols by segmenting lines and words. There are two types of segmentation, namely, external and internal segmentation, as per Arica and Vural [2001], are explained in the next section.

1.5.1 EXTERNAL SEGMENTATION

It is the isolation of various writing units, such as paragraphs, sentences, or words, and internal segmentation, which is the isolation of letters, especially in cursively written words. It is the most critical part of the document analysis, which is a necessary step prior to the offline character recognition. Although document analysis is a relatively different research area with its own methodologies and techniques, segmenting the document image into text and non text regions is an integral part of the OCR software.

1.5.2 INTERNAL SEGMENTATION

It is the isolation of letters, especially in cursively written words. Although the methods have developed remarkably in the last decade and a variety of techniques have emerged, segmentation of cursive script into letters is still an unsolved problem. The character segmentation strategies namely, explicit segmentation, implicit segmentation and mixed strategies, are explained here:-

- **EXPLICIT SEGMENTATION**

  In this strategy, the segments are identified based on "character like" properties. The process of cutting up the image into meaningful components is given a special name, dissection. Dissection is a process that analyzes an image without using a specific class of shape information. The criterion for good segmentation is the agreement of general properties of the segments with those expected for valid characters.

- **IMPLICIT SEGMENTATION**

  This segmentation strategy is based on recognition. It searches the image for components that match predefined classes. Segmentation is performed by the use of recognition confidence, including syntactic or semantic correctness of the overall result. In this approach, two classes of methods can be employed, which are methods that make some search process and methods that segment a feature representation of the image.

- **MIXED STRATEGIES**

  They combine explicit and implicit segmentation in a hybrid way. A dissection algorithm is applied to the image, but the intent is to over segment, *i.e.*, to cut the image in sufficiently many places that the correct segmentation boundaries are included among the cuts made. Once this is assured, the optimal segmentation is sought by evaluation of subsets of the cuts made. Each subset implies a segmentation hypothesis, and classification is brought to bear to evaluate the different hypothesis and choose the most promising segmentation.

The next section elaborates upon the hierarchy of the segmentation of the document image.

1.5.3 HIERARCHY OF SEGMENTATION

The segmentation hierarchy of the document can be described on the basis of the structure of the document. A document generally consists of several pages. A page usually contains several lines of text, which are, in turn made up of words. The words can be further seen as a grouping of characters. Based on this structure, the segmentation hierarchy can be defined as shown here:-

- **PAGE SEGMENTATION**

  Page segmentation can be defined as the process of extracting pages from a document.

- **LINE SEGMENTATION**

  Line segmentation can be defined as the process of extracting lines from a page.

- **WORD SEGMENTATION**

  Word segmentation can be defined as the process of extracting words from a line.

- **CHARACTER SEGMENTATION**

  Character segmentation can be defined as the process of extracting characters from a word.

1.5.4 SIGNIFICANCE OF SEGMENTATION

The extent one can reach in separation of lines, words or characters in segmentation process directly affects the recognition rate of the script.

Segmentation of a given document into lines, words and characters give help in the recognition stage. Segmentation helps in increasing the recognition rate. If the segmentation of any script is done in an accurate manner then it directly affects its recognition rate *i.e*. Accurate segmentation will increase the recognition rate.

There are various strategies which can be applied to perform the segmentation of the document according to the above explained hierarchy. The strategies for segmentation to deal with are in the next section.

1.6 STRATEGIES IN SEGMENTATION

According to Casey and Lecolinet [1996], the segmentation strategies which can be categorized as classical approach, recognition based segmentation approach and holistic approach are discussed here:-

- Classical approach
- Recognition-based segmentation
- Holistic method

1.6.1 CLASSICAL APPROACH

In classical approach segments are identified based on "character-like" properties such as height, width, separation from neighboring components, disposition along a baseline, etc. This technique cuts image into a sequence of sub-images, which are called dissections.

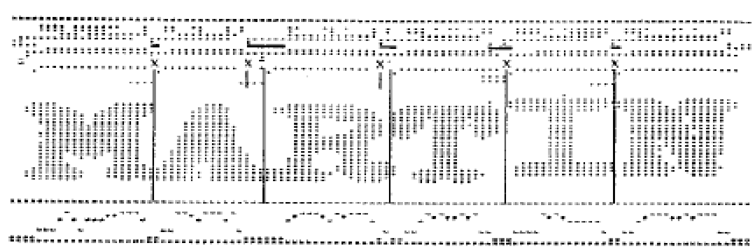
The criterion for good segmentation is the agreement of general properties of the segments obtained with those expected for valid characters. Examples of such properties are height, width, separation from neighboring components, disposition along a baseline etc. This process of cutting up the image into meaningful components is given a special name, "dissection". Under dissection there are various techniques like White Space and Pitch, Projection Analysis, Bounding Box analysis.

- **WHITE SPACE AND PITCH**

  In machine printing, vertical whitespace often serves to separate successive characters. This property can be extended to handprint by providing separated boxes in which to print individual symbols. In applications such as billing, where document layout is specifically designed for OCR, additional spacing is built into the fonts used. The notion of detecting the vertical white space between successive characters has naturally been an important concept in dissecting images of machine print or handprint. In many machine print applications involving limited font sets, each character occupies a block of fixed width. The pitch, or number of characters per unit of horizontal distance, provides a basis for estimating segmentation points. The sequence of segmentation points obtained for a given line of print should be approximately equally spaced at the distance corresponding to the pitch. This provides a global basis for segmentation, since separation points are not independent. Hoffman and McCullough [1971] generalized this process and gave it a more formal framework. In their formulation, the segmentation stage consisted of three steps:-

- Detection of the start of a character.
- A decision to begin testing for the end of a character.
- Detection of end-of-character (called sectioning).

  Sectioning, step 2, was the critical step. It was based on a weighted analysis of horizontal black runs completed, versus runs still incomplete as the print line was traversed column by column.



The dissection method of Hoffman and McCullough. An evaluation function based on a running count of horizontal black-white and white-black transitions is plotted below the image. The horizontal black bars above the image indicate the activation region for the function. Vertical lines indicate human estimates of optimal segmentation

Figure 1.3 White Space and Pitch Method

- **PROJECTION ANALYSIS**

The vertical projection also called the "vertical histogram" of a print line consists of a simple running count of the black pixels in each column. It can serve for detection of white space between successive letters it can indicate locations of vertical strokes in machine print, or regions of multiple lines in handprint. Thus, analysis of the projection of a line of print has been used as a basis for segmentation of non cursive writing. When printed characters touch, or overlap horizontally, the projection often contains a minimum at the proper segmentation column.

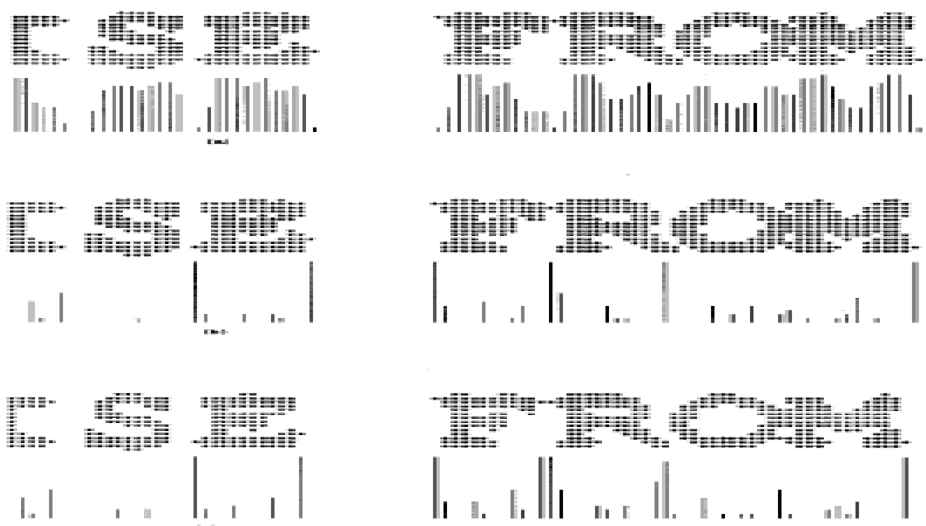


Figure 1.4 Projection Analysis

In the first part of the above image, we are calculating the number of black pixels corresponding to each column. It is an easy matter to detect white columns between characters, and regions of light contact. The function fails, however, to make clear the O-M separation.

In the second part of the above image, we are calculating the difference between the total number of black pixels and the number of black pixels we get from first part of figure. This gives a clear peak at most separation columns, but may still fail for the O-M case.

In the third part of the above image, we are calculating Differencing measure after column ANDing. The image transformed by an AND of adjacent columns, with the difference function of second part of the image applied to the transformed image. The AND operation tends to increase separation, leading to a better defined peak at the O-M boundary.

When printed characters touch, or overlap horizontally, the projection often contains a minimum at the proper segmentation column. In first part of the image, the projection was first obtained, and then the ratio of second derivative of this curve to its height was used as a criterion for choosing separating columns. This ratio tends to peak at minima of the projection, and avoids the problem of splitting at points along thin horizontal lines.

A peak-to-valley function was designed to improve on this method. A minimum of the projection is located and the projection value noted. The sum of the differences between this minimum value and the peaks on each side is calculated. The ratio of the sum to the minimum value itself is the discriminator used to select segmentation boundaries.

- **BOUNDING BOX ANALYSIS**

  The distribution of bounding boxes tells a great deal about the proper segmentation of an image consisting of non cursive characters. By testing their adjacency relationships to perform merging, or their size and aspect ratios to trigger splitting mechanisms, much of the segmentation task can be accurately performed at a low cost in computation.

  This approach has been applied, for example, in segmenting handwritten postcodes using knowledge of the number of symbols in the code.

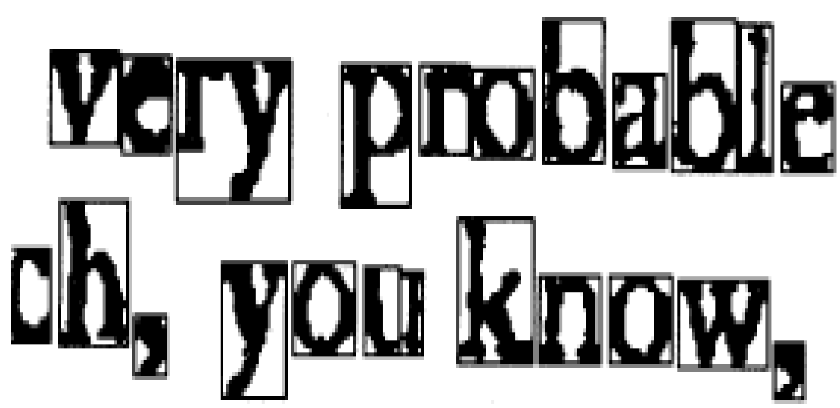


Figure 1.5 Bounding Box Analysis

In the figure 1.5 characters that consist of two components (*e.g.*, the "U" in "you"), as well as components consisting of more than one character (*e.g.*, the "ry" in "very"). The bounding box of each component is also shown.

1.6.2 RECOGNITION BASED SEGMENTATION

After an image has been segmented into regions, it is ready to enter the next level that is, the feature extraction stage. The end result of the image acquisition, preprocessing, segmentation and character fragmentation is a matrix of numbers that represents a character fragment in some way. In the general case, however, the matching of these numbers to a template may be too time consuming and not flexible enough. Therefore, feature extraction is needed. It assigns an input character to one of many pre specified classes which are based on the extracted features and their analysis.

Feature extraction is concerned with recovering the defining attributes obscured by imperfect measurements. To represent a character class, either a prototype or a set of samples must be known. The feature selection process attempts to recover the pattern attributes characteristic of each class. Global features, such as the number of holes in the character, the number of concavities in its outer contour, and the relative protrusion of character extremities, and local features, such as the relative positions of line endings, line crossovers, and corners are commonly used.

Recognition-based segmentation, in which the system searches the image for components that match classes in its alphabet. Many people think that the existence of reliable features to distinguish boundaries in all fonts from interior regions is arguable, open-loop approaches, segmentation-to-recognition, and render errors irrecoverable. Therefore character segmentation should be closely coupled with character recognition.

Structural features of each character fragment are extracted in this method.

The recognition based segmentation techniques are:-

- Sliding window method
- Closed-loop segmentation and recognition
- Multiple hypothesis scheme

1.6.3 HOLISTIC METHODS

There are many applications that require the recognition of unconstrained handwritten words. A word can be either purely numeric as in the case of a ZIP Code, or purely alphabetic as in the case of US state abbreviations or mixed as in the number of an apartment for *e.g.,* 1A. In general, a character string recognizer has many applications. The applications include, but are not limited to, reading bank checks, reading tax forms and interpretation of postal addresses. The task becomes particularly challenging when adjacent characters in a character string are touching. Unlike purely alphabetic strings where joining of the characters is natural and takes place by means of ligatures, the joining of numerals in a numeric word and the uppercase characters in an abbreviation are accidental. The various ways in which two digits can touch are categorized. Some of the categories lend themselves to natural segmentation, whereas for some the holistic approach is the only option available.

In holistic method the system seeks to recognize words as a whole, thus, avoiding the need to segment into characters. Holistic methods in essence revert to the classical approach with words as the alphabet to be read.

This method performs feature extraction in the first step, then global recognition by comparing the representation of the unknown word with those of the references stored in the lexicon. Consequently, this method uses the "classical approach".

One of the advantages of the holistic approach is that no distinction needs to be made among touching and non-touching pairs. A segmentation based method must necessarily make such a determination. While the holistic method applies equally to both touching and non touching characters, its advantage over traditional segmentation based methods is more pronounced among the touching character pairs. Such pairs will be referred to as TCP and TDP for touching character pairs and touching digit pairs, respectively.

Input to the TDP problem is a pair of digits extracted from numeric strings in an address and the expected output is its classification as one of possible classes. Input to the TCP problem is a pair of characters that represent the US state abbreviation as it appears on an address. The expected output is the identity of the state name. A small subset of the 676 (26*26) possible abbreviations are valid.

1.7 APPLICATIONS OF OPTICAL CHARACTER RECOGNITION

Some of the more important applications of offline optical character recognition are discussed in the following section:-

CHEQUE READING: Offline character recognition is basically used for cheque reading in banks. Cheque reading is the very important commercial application of offline handwritten recognition. Handwritten recognition system plays very important role in banks for signature verification and for recognition of amount filled by user. The Cheque OCR is a unique Optical Character Recognition based Cheque reader which is a useful and critical application for intelligent archiving solution in bank, investment or any institute worked with the cheques and needs to have an intelligent archiving solution. Cheque OCR solution includes software for scan, read and recognize number of cheque with its intelligent algorithms and to store front and back sides of each cheque with a high quality and compressed size in database and to prepare an acquire facilities for users to search and find specific cheque by its customer, date or number information from database. Generally, MICR band consists of four parts, *i.e.*, Cheque number, MICR code, Bank account number and Transaction id.

Cheque Number: - The first set of numbers represents the cheque number. It is a six digit number.

MICR Code: - It stands for Magnetic Ink Character Recognition. This number helps a bank to recognize the bank and branch that issued the cheque. Banks have to process hundreds of cheques daily. Going through each and every cheque is a cumbersome process. Instead, the cheques are sorted through a cheque reading machine which uses this number to identify the bank and branch a cheque belongs to. This makes the process faster.

The MICR number is a nine digit number, which consists of three parts:-

City Code: - The first three digits represent the city code and are same as the first three digit of the PIN code of that city. For *e.g.*, a bank in Hyderabad will have first three digits of MICR code as 500 (since PIN code for Hyderabad starts with 500)

Bank Code: - The next three digits represent the bank code. Every bank has a unique code assigned to it. For *e.g*., ICICI bank's code is 229, for HDFC it is 240 and so on.

Branch Code: -The last three digits represent the branch code.

Bank account Number: - The third set of six digit numbers represents account number (It consists of a few digits of account number).

Transaction ID: - The last two digits tell whether a cheque is a local cheque or payable at par cheque. 29, 30 and 31 represents payable at par cheque, while 09, 10 and 11 represents local cheque.

MICR characters are printed in special typefaces with a magnetic ink or toner, usually containing iron oxide. As a machine decodes the MICR text, it first magnetizes the characters in the plane of the paper. Then the characters are passed over a MICR read head, a device similar to the playback head of a tape recorder. As each character passes over the head it produces a unique waveform that can be easily identified by the system.

POSTCODE RECOGNITION: Handwritten recognition system can be used for reading the handwritten postal address on letters. Offline handwritten recognition system used for recognition handwritten digits of postcode. Handwritten character Recognition can be read this code and can sort mail automatically.

FORM PROCESSING: Handwritten character Recognition can be also used for form processing. Forms are normally used for collecting the public information. Replies of public information can be handwritten in the space provided.

SIGNATURE VERIFICATION: Handwritten character Recognition can be also used for identify the person by signature verification. Signature identification is the specific field of handwritten identification in which the writer is verified by some specific handwritten text. Handwritten recognition system can be used for identify the person by handwriting, because handwriting may be vary from person to person.

CHAPTER 2

LITERATURE SURVEY

Optical character recognition acts as a challenging field for the researchers. The inexplicable research work in this area has led to the development of innumerable approaches to deal with the various aspects of the character recognition. This technology has been widely implemented in different walks of business; the major area of OCR is banks. The use of OCR in banks has revolutionized the industry, making life easier for all. In order to understand the current state of art in this area, a survey of work done related to the character recognition process and bank cheque processing has been presented in this chapter.

Lu [1995] presented an overview of character segmentation techniques in machine printed documents and hand printed text and covered techniques for segmenting uniformed or proportional fonts, broken and touching characters. The segmentation of fixed pitch text, broken characters, and kerned characters has also been summed up. The various steps for textual processing like determining skew, finding paragraphs and columns, performing optical character recognition have been explained. The segmentation techniques based on character size, contour analysis and vertical projections have been discussed. Paper also summarized techniques based on text image features and techniques based on recognition results.

Zhao and Srihari [1995] presented an algorithm for recognition of machine printed word images of multiple font types and varying qualities. The information provided by both the entire shape of the word and its component letters was utilized in recognition. The algorithm used the general shape of a word as the primary cue in recognition, with a few letters, which are highly identifiable when compared with the others, as the supplementary cues. In determining word shape, an ideal word pattern approach was applied to deal with the difficulty introduced by large variety of font types and image qualities. They reduced the size of given lexicon by recognizing the first and last characters.

Casey and Lecolinet [1996] provided a review of various techniques and methodologies in character segmentation. They described segmentation methods under four main headings. First was classical approach which consisted of methods that partition the input image into sub images, which were then classified. The operation of attempting to decompose the image into classifiable units was called "dissection." The second class of methods avoided dissection, and segments the image either explicitly, by classification of pre specified windows, or implicitly by classification of sub sets of spatial features collected from the image as a whole. The third strategy was a hybrid of the first two, employed dissection together with recombination rules to define potential segments, but was using classification to select from the range of admissible segmentation possibilities offered by these sub images. Fourth was holistic approach that avoided segmentation by recognizing entire character strings as units are described.

Trier *et al*. [1996] presented an overview of feature extraction methods for offline recognition of segmented (isolated) characters. Different feature extraction methods were designed for different representations of the characters, such as solid binary characters, character contours, skeletons (thinned characters) or gray level sub images of each individual character. The feature extraction methods were discussed in terms of invariance properties; reconstruct ability and expected distortions and variability of the characters. The problem of choosing the appropriate feature extraction method for a given application was also discussed.

Liu *et al*. [1996] proposed a novel approach for the extraction of legal and courtesy amounts and date from cheque images based on the structural description of cheques. A method for the representation of cheques has been presented. Several image processing techniques and algorithms have been discussed.

Suen *et al*. [1996] interpreted various items of information contained in the image of a cheque. To isolate and identify each category of handwritten information, implementation of image processing has been done. They used relevant recognizers to read the numeric and legal amounts, as well as the date on the cheque.

Wu and Manmatha [1997] developed a simple and effective algorithm for document image clean up and binarization. The need for binarization, as well as the issues in binarization has also been discussed. The proposed algorithm smoothen the input image using Gaussian filter and then thresholds it based on the intensity histogram of the image. The global and local thresholding techniques with the related issues in them have been discussed. A comparison of the proposed algorithm with other popular algorithms like Otsu's Algorithm, Tsai's method has also been made.

Ha and Bunke [1997] presented a new approach to offline, handwritten numeral recognition. From the concept of perturbation due to writing habits and instruments, they proposed a recognition method which is able to account for a variety of distortions due to eccentric handwriting. The key idea of the perturbation approach, which lies in the process of reversing an input image back to one of its standard forms, has also been stated. The methodology for the parameterization process based on four geometric transformations, namely, rotation, slant, perspective view and shrink has also been explained.

Mo and Mathews [1998] presented an adaptive algorithm for preprocessing document images prior to binarization in character recognition problems. The adaptive filter utilized a quadratic system model to provide edge enhancement for input images that have been corrupted by noise and other types of distortions during the scanning process. Experimental results demonstrated significant improvement in the quality of the binarized images over both direct binarization and a previously available preprocessing technique has also been included in this paper.

Senior and Robinson [1998] described a complete system for the recognition of offline handwriting. Preprocessing techniques have been described, including segmentation and normalization of word images to give invariance to scale, slant, slope and stroke thickness. Issues of vocabulary choice and rejection have also been discussed. The techniques for histogram production, baseline estimation, slant correction, smoothing, thresholding, thinning and parameterization have also been explained. Representation of the image has been discussed and the skeleton and stroke features used have been described.

Guillevic and Suen [1998] described the recognition of legal amounts of a bank cheque processing system. They described overall system as a combination of a global feature scheme with a hidden markov model. The global features consisted of encoding of relative position of ascenders, descenders and loops within a word. Hidden markov model used one feature set based on the orientation of contour points as well as their distance to the baselines. The system presented is modular and independent of specific languages.

Cai and Liu [1999] proposed an approach that integrates the statistical and structural information for unconstrained handwritten numeral recognition. This approach used state duration adapted transition probability to improve the modeling of state duration in conventional hidden markov model and uses macro states to overcome the difficulty in modeling pattern structures by hidden markov model. The proposed method is superior to conventional approaches in many aspects. In the statistical and structural models, the orientations were encoded into discrete codebooks and the distributions of locations were modeled by joint Gaussian distribution functions.

Blumenstein and Verma [1999] proposed an algorithm for segmenting unconstrained printed and cursive words. The algorithm had initially over segments handwritten word images (for training and testing) that used heuristics and feature detection. An artificial neural network was then trained with global features extracted from segmentation points found in words designated for training. Segmentation points located in test word images were subsequently extracted and verified using the trained artificial neural network.

Ye *et al*. [1999] presented a technique for extracting the user entered information from bank cheque images based on a layout driven item extraction method. The baselines of cheques has been detected and eliminated by using gray level mathematical morphology. A priori information about the positions of data has been integrated into a combination of top down and bottom up analyses of cheque images. The handwritten information has been extracted by a local thresholding technique and the information lost during baseline elimination has been restored by mathematical morphology with dynamic kernels. A goal directed evaluation of the extraction approaches has been proposed.

Plamondon and Srihari [2000] rolled out a comprehensive survey for the online and offline handwriting recognition. The paper described the features of the handwritten language, their translation into the electronic form and the underlying concepts behind the handwritten character recognition algorithms. The various algorithms for preprocessing, character and word recognition and their performance have also been described. The classification methods like the structural and rule based methods and the statistical methods have been discussed. The applications of offline handwritten character recognition like handwritten address interpretation, bank cheque recognition, signature verification has also been described. The language tools for language processing and analysis have been stated. The various application areas and recent tools available in these areas have also been presented.

Jain *et al*. [2000] summarized and compared some of the well known methods like template matching, statistical, syntactic or structural, neural networks used in various stages of a pattern recognition system and identified research topics and applications which were at the forefront of this challenging field.

Arica and Vural [2001] presented the historical evolution of CR systems. Then, the available CR techniques with their superiorities and weaknesses have been reviewed. Finally, the current status of CR has been discussed, and directions for future research have been suggested. Special attention has been given to the offline handwriting recognition.

Morita *et al*. [2001] described an offline system under development to process unconstrained dates on Brazilian bank cheques in an Omni-writer context. After preprocessing, a word image has been explicitly segmented into characters or pseudo-characters and represented by two feature sequences of equal length, which have been combined using hidden markov model. The word models have been generated from the concatenation of appropriate character models. In addition to the small date database, they used some of the legal amount database to increase the frequency of characters in the training and the validation sets.

Bansal and Sinha [2002] presented a two pass algorithm for the segmentation and decomposition of Devanagari composite characters or symbols into their constituent symbols. The proposed algorithm extensively used structural properties of the script. In the first pass, words have been segmented into easily separable characters or composite characters. Statistical information about the height and width of each separated box has been used to hypothesize whether a character box is composite. In the second pass, the hypothesized composite characters have been further segmented. The algorithm has been designed to segment a pair of touching characters.

Timar *et al.* [2002] described the various algorithms used in the preprocessing and segmentation phase of offline handwritten character recognition process. Various tasks of preprocessing phase such as thinning, localization, de skewing have also been discussed. The strategies for line localisation and word localisation have also been discussed. The segmentation technique for words using the skeleton endpoints has been explained. A novel trigger wave-based word segmentation algorithm has been presented in the paper, which operates on the skeletons of words. The technique for localisation of lower and upper baselines, producing the skeleton of the background of the image, producing skeletons of the foreground of the image and their usage in segmentation of words into characters has also been described.

Kasturi *et al*. [2002] gave a detailed description of the document image analysis process. The sequence of steps started from data capture, pixel level processing, feature level analysis till text recognition and analysis has elaborated upon. A brief analysis of graphical documents has been presented. The techniques for noise reduction and binarization have been discussed. The techniques for thinning and region detection, chain coding and vectorization have also been explained. The techniques for line and curve fitting, critical point detection, skew estimation, layout analysis have also been discussed. The strategy for feature extraction and classification based on template matching and contextual processing has also been explained. The various OCR's for Indian languages and document analysis in multilingual context has also been stated.

Pal *et al*. [2003] presented an automatic scheme to identify text lines of different Indian scripts from a document. For the separation task at first the scripts have been grouped into a few classes according to script characteristics. Next feature based on water reservoir principle, contour tracing, profile have been employed to identify them without any expensive OCR like algorithms. The technique for the identification of the script based on the various features has also been discussed. The approach has insensitive to font, style and case variation has been stated.

Allier and Emptoz [2003] presented a particular application of Gabor filtering for machine printed document image. They assumed that the text can be seen as texture, characters being the smallest texture elements, and verified this hypothesis by a series of experiments over different sets of character images. They applied a bank of 24 Gabor filters (4 frequencies and 6 orientations) on each set, and then extracted texture features used to classify character images without a priori knowledge using a Bayesian classifier. Results were shown for different characters written in a same font, and for different font types.

Siromoney *et al*. [2003] described computer recognition of machine printed letters of the Tamil alphabet. Each character has been represented as a binary matrix and encoded into a string using two different methods. The encoded strings form a dictionary. A given text has been presented symbol by symbol and information from each symbol has been extracted in the form of a string and compared with the strings in the dictionary. The letters have recognized and printed out in roman letters following a special method of transliteration.

Ohali *et al*. [2003] described an effort towards the development of Arabic cheque databases for research in the recognition of handwritten Arabic cheques. Databases of real life Arabic legal amounts, Arabic sub words, courtesy amounts, Indian digits, and Arabic cheques have been described. This paper highlighted some characteristics of the Arabic language and presented the various steps that have been completed to build these databases including segmentation, binarization and data tagging. It also described a solid validation procedure including grammars and algorithms used to verify the correctness of the tagging process. Detailed descriptions of the database organization and class distribution have been included.

Pal and chaudhuri [2004] presented a review of the OCR work done on Indian language scripts. The character recognition process, its applications, its history has been explained. The properties of the Indian scripts like basic characters, modified characters, headline and zones have been stated. They stated that the character recognition for the poor quality documents, multi font documents, multi script documents, handwritten documents needs to be developed. The character recognition systems with font and structure information, benchmarking and ground truth generation and post recognition error correction are also required to be designed.

Zheng *et al*. [2004] presented a new machine printed Arabic character segmentation algorithm, which is based on the vertical histogram and some rules. The rules were based on the structural characteristics between background regions, character components and characteristics of isolated Arabic characters. These rules were used to check whether the sub word includes only one character. They used vertical histogram and some other rules to find real segmentation points.

Farooq *et al*. [2005] described the techniques used for slant and slope correction and line and word segmentation in the handwritten Arabic documents. A linear regression technique has been used for baseline detection and connected components have been identified for line and word segmentation. An orientation independent technique for baseline detection of Arabic words has been used. The procedure for using the horizontal projection profile for the line segmentation and the vertical projection profile for the word segmentation has also been discussed. The skew correction algorithm has been used to improvise the results for the line segmentation and word segmentation in the handwritten Arabic documents.

Seethalakshmi *et al*. [2005] referred optical character recognition as a process of converting printed text documents into software translated Unicode Tamil text. In a preprocessing phase the image file has been checked for skewing. If skewed, corrected by simple rotation technique. This image passed through a noise elimination phase and has been binarized. The segmentation algorithm converted words into character image glyphs using horizontal histograms. Image glyphs have been considered for recognition using Unicode mapping. The various features that have been considered for classification are character width, character height, number of horizontal lines, number of vertical lines number of slope lines and so on. The extracted features have been passed to a support vector machine where the characters have been classified by supervised learning algorithm.

Li *et al*. [2006] modeled text line detection as an image segmentation problem by enhancing text line structures using a Gaussian window and adopted the level set method to evolve text line boundaries. Level set based detection techniques like text line structure enhancement, text line boundary evolution, and text line refinement has been discussed.

Majumdar and Chaudhuri [2006] presented automatic recognition of both printed and handwritten Bangla numerals. Pixel based and shape based features has been chosen for the purpose of recognition. The pixel based features have been normalized pixel density over 4 × 4 blocks in which the numeral bounding box has been partitioned. The shape based features have been normalized position of holes, end points, intersections and radius of curvature of strokes found in each block. Multi layer neural network architecture has been chosen as classifier of the mixed class of handwritten and printed numerals.

Fujisawa [2007] presented an overview on the technical advances in the field of character and document recognition, decade by decade. Also key technical developments have been highlighted especially for Kanji (Chinese character) recognition in Japan. Technical issues around post address recognition have been discussed. Advanced techniques including information integration have been promoted. Robustness design principles have been introduced.

Shaaban [2008] presented a new approach to tackle the problem of recognizing machine printed Arabic texts. Because of the difficulty of recognizing cursive Arabic words, the text has been normalized and segmented to be ready for the recognition stage. The new scheme for recognizing Arabic characters depended on multiple parallel neural networks classifier. The classifier has two phases. The first phase categorized the input character into one of eight groups. The second phase classified the character into one of the Arabic character classes in the group.

Sharma and Singh [2008] presented an algorithm for the segmentation of the handwritten text in Gurmukhi script. They stated that the technique to segment the characters is to use inter character gap as a segmentation point. The various characteristics of the Gurmukhi script have been explained. The method for the preprocessing of the document to remove the noise has also been stated. The algorithm for line detection, word detection and character detection has been discussed.

Kannan and Prabhakar [2009] gave an overview of the ongoing research in optical character recognition (OCR) systems for Tamil scripts. The paper aimed to provide a starting point for the researchers entering into this field. Peculiarities in Tamil scripts, present status of the OCRs for Tamil scripts, techniques used in them, recognition accuracies, and the resources available, have been discussed in detail.

Prasad *et al.* [2009] presented a system for the offline handwritten character recognition of Gujrati script using pattern matching. The methods for image acquisition, median filtering, image inversion, image thinning, segmentation and recognition have been explained in this paper.

Kumar and Singh [2010] formulated an approach to segment the scanned document image. This approach initially considered the whole image as one large window and then broke the large window into less large windows giving the lines has been explained. The character recognition process and the segmentation strategies have also been elaborated upon. The method that used the window consisting of identified line to find the words present in that line has been discussed. The algorithm that used the window defined by identified word to find the characters present in that word has also been explained. They stated that a concept of variable sized window, that is, the window whose size can be adjusted according to needs, has been employed for this approach.

Sankari *et al*. [2010] discussed verification of bank cheques by using account number and account holder's signature presents on the cheque image. Problem of exact localization of active regions among non active contours in the image has been discussed. They located the regions based on the prior knowledge of Cartesian coordinate space which further involved various steps such as gray scale conversion, segmenting contour, inner localization of account number, feature extraction and verification. These steps have been performed on cheque images for reducing dimensionality of cheque size. They employed segmentation and localization together to extract active regions of interests such as account number and signature. Furthermore, segmented account numbers have been obtained into individual digits using inner localization. Feature extraction has been implemented on both account number and signature with trained images using hamming distance measures. Finally to identify the matching verification is processed.

Jayadevan *et al*. [2011] presented the state of the art in automatic processing of handwritten cheque images. They discussed the important results reported so far in preprocessing, extraction recognition and verification of handwritten fields on bank cheques and highlighted the positive directions of research till date. The paper has a comprehensive bibliography of many references as a support for researchers working in the field of automatic bank cheque processing. The paper also contained some information about the products available in the market for automatic cheque processing.

Wen *et al*. [2011] proposed an algorithm for license plate recognition applied to the intelligent transportation system on the basis of a novel shadow removal technique and character recognition algorithms. This paper consisted of two major contributions. One contribution has been a new binary method, *i.e.*, the shadow removal method, which is based on the improved Bernsen algorithm combined with the Gaussian filter. Second contribution is a character recognition algorithm known as support vector machine integration. This paper also presented improved techniques for image tilt correction and image gray enhancement.

Alginahi [2012] presented the description of the Arabic script characteristics with an overview on OCR systems and a comprehensive review mainly on offline printed Arabic character segmentation techniques. Segmentation techniques like segmentation based on histogram and baseline, segmentation based on contour tracing, segmentation based on thinning, segmentation based on neural network, segmentation based on morphological operators, segmentation based on hidden markov model, segmentation based on template matching, segmentation based on transforms, segmentation based on strokes, segments and tokens, holistic approach in Arabic character segmentation have been discussed.

Patil and Begum [2012] presented a method for discriminating handwritten and printed text from document images based on shape features. They stated that the separation of handwritten and printed text from document image has been essential to optimize the OCR accuracy and to activate an appropriate OCR engine. It leads to reduce the search space of the OCR and it facilitated the retrieval of handwritten and printed text from document images.

CHAPTER 3

PROBLEM DEFINITION AND ITS SOLUTIONS

3.1 PROBLEM DEFINITION

As the population is growing, bank transactions which involves use of bank cheques is increasing rapidly. Although there is overall rapid emergence of electronic payments by credit cards but the bank transactions involving cheques are still increasing throughout the world. After doing the literature survey, it has been observed that a great deal of work has been done on bank cheques like detection and recognition of date, amount filled by the user, signature verification etc., but no work has been done related to the MICR band.

MICR band which is present at the bottom of the cheque gives information about the cheque no, MICR code, account no and transaction id. MICR code is a 9 digit code. It constitutes a code to represent city, bank and branch (3 digits each). This nine digit code is unique for any bank-branch combination. MICR stands for Magnetic Ink Recognition. For reading this code banks are using special type of scanner which has certain disadvantages like MICR scanners are very precisely designed machines that are expensive and need constant maintenance to maintain their reliability. In writing this code in cheques, special type of magnetic ink is needed which is very costly. As the MICR scanners read the magnetic ink particles, any damage to MICR character, will hamper its performance. As the bank transactions involving cheques are increasing rapidly, more number of scanners is needed, this becomes very costly.

Taking these disadvantages in mind, authors are motivated to develop an algorithm for the detection and recognition of the MICR code so that it will help to read the code without using magnetic ink on cheques, and thus no need of special type of scanners. All the work can be done through simple scanner and software, which will help in reducing cost.

3.2 PROPOSED SOLUTION

The authors proposed a new idea, to use the concept of OCR in processing of MICR code in bank cheques, which includes a number of stages as discussed in chapter 1. In order to provide a solution to the problem explained in the previous section, a number of techniques are used, which are discussed in this section.

In the proposed solution, image of cheque can be taken with the help of scanner. As the image will be colored, so binarization technique will be needed to change the image in black and white. Now as the work to be done is on the MICR code, which is present at the bottom of the cheque, the line can be extracted using the Line_segmentation module. After this Word_segmentation algorithm can be used to extract three words of three digits each. After this, Digit_segmentation module can be used to extract each digit from each word. As each digit is of different dimension, resizing approach can be used to make each digit of every cheque to same dimension. Further patterns can be recognized using Pattern_recognition algorithm. If this process will not be able to recognize any digit, then that digit will be send to thinning module, and again the pattern will be recognized. Once able to recognize digits, information regarding city, bank and branch name can be fetched through database.

The cheque image is initially acquired through scanner say, Cheque_Image, which is referred as figure 4.1 in the next chapter.

ALGO 1: MICR_CODE_PROCESSING (Cheque_Image)

This is the main algorithm which is calling number of modules. First module is binarization, which converts the colored image into black and white image. In second module, Line_segmentation, line containing the MICR band is extracted. Third module, Word_segmentation, will extract the word, of three digits each from the extracted line. In module Digit_segmentation will extract each digit from the extracted word. Resizing module will make each extracted digit of same dimension so that patterns can be easily recognized. In grid_making module, a matrix of 6×4 is created for each resized digit and corresponding to each grid, number of black pixels is counted. In pattern recognition, after studying so many cheques, a common pattern has been found to recognize each digit. Finally after recognizing each digit, it is matched from the database and corresponding to that city, bank and branch name of the bank who has issued that cheque is displayed on the screen.

The algorithms that are discussed under proposed solution have been implemented in C# using .NET framework.

The Drawing namespace and Bitmap class present in the inbuilt libraries of C# is utilized to process the bitmap image. The file header as well as Info header, used to read the BMP file, is given below:-

```
Private Type
BITMAPFILEHEADER
bfType As Integer
bfSize As Long
bfReserved1 As Integer
bfReserved2 As Integer
bfOhFileBits As Long
End Type
```

```
Private Type
BITMAPINFOHEADER
biSize As Long
biWidth As Long
biHeight As Long
biplanes As Integer
biBitCount As Integer
biCompression As Long
biSizeImage As Long
biXPelsPerMeter As Long
biYPelsPerMeter As Long
biClrUsed As Long
biClrImportant As Long
End Type '40 bytes
```

Input Parameters:

A bitmap image say, Cheque_Image, is given as input to the MICR_CODE_PROCESSING algorithm.

Output Parameters:

A file say, city_bank_branch, is produced as output from this algorithm.

The pseudo code for MICR_CODE_PROCESSING is shown below.

```
MICR_CODE_PROCESSING (Cheque_Image)
    Call Binarization (Cheque_Image)
    //The output from binarization is saved as Binarization_output.
    Call Line_segmentation (Binarization_output)
    //The output from Line_segmentation is saved as Line_segmentation_output
    Call Word_segmentation (Line_segmentation_output, line_points)
    //The output from Word_segmentation is saved as Word_segmentation_output
    Call Digit_segmentation (Word_segmentation_output, word_points)
    //The output from Digit_segmentation is saved as Digit_segmentation_output
    For each digit
        Call Resizing (Digit_segmentation_output, digit_points)
        //The output from Resizing is saved as Resized_image
        Call Grid_making (Resized_image, 0, 0, Resized_image_width-1,
        Resized_image_height-1)
        //The output from Grid_making is saved as grid_file.
        Call Pattern_recognition(grid_file)
        //The output from Pattern_recognition is saved as numbers.
    Cheque_details(numbers)
    //The output from Cheque_details is saved as  city_bank_branch
End of MICR_CODE_PROCESSING
```

The MICR_CODE_PROCESSING algorithm calls several other modules, which are discussed in the following section.

MODULE 1.1: Binarization (Cheque_Image)

The gray scale or color images are converted to the binary images by picking a threshold value as it reduces the memory requirements and increases the processing speed. The current module uses a global threshold value, as stated in Kumar and Singh [2011].

Input Parameters:

A bitmap image, Cheque_Image, is given as input to the Binarization module.

Output Parameters:

A bitmap image say, Binarization_output, is produced as output from this module.

The pseudo code for Binarization is shown below.

```
Binarization (Cheque_Image)
For X=Min_X to Max_X
    For Y=Min_Y to Max_Y
        Set Pixel_Intensity_Sum=Pixel_Intensity_Sum+Pixel_Intensity(X, Y)
        Set Pixel_Count=Pixel_Count+1
Set Average_Intensity= Pixel_Intensity_Sum/ Pixel_Count
For X=Min_X to Max_X
    For Y=Min_Y to Max_Y
        If (Pixel_Intensity(X, Y) >=Average_Intensity)
            Set Pixel_Intensity(X, Y) =WHITE
        Else
            Set Pixel_Intensity(X, Y) =BLACK
Return Binarization_Output
End of Binarization.
```

Intermediate Parameters:

Min_X - Defines the minimum x coordinate value for Cheque_Image.

Max_X - Defines the maximum x coordinate value for Cheque_Image.

Min_Y - Defines the minimum y coordinate value for Cheque_Image.

Max_Y - Defines the maximum y coordinate value for Cheque_Image.

Pixel_Intensity(X, Y) - Returns or Sets the pixel intensity at coordinates X and Y.

The binarized image which is obtained as output after application of the Binarization module is Binarization_output which is referred as figure 4.2 in the next chapter.

MODULE 1.2: Line_segmentation (Binarization_output)

The line containing the MICR code is present at the bottom of the cheque, which has been extracted using the Line_segmentation module. This will be look like as shown in the figure 3.1 below. This module is counting the number of black pixels for the complete height corresponding to each line, and on the basis of the difference between the two adjacent black pixel counts, a line is extracted. The current module uses a technique, as stated in Kumar and Singh [2011].

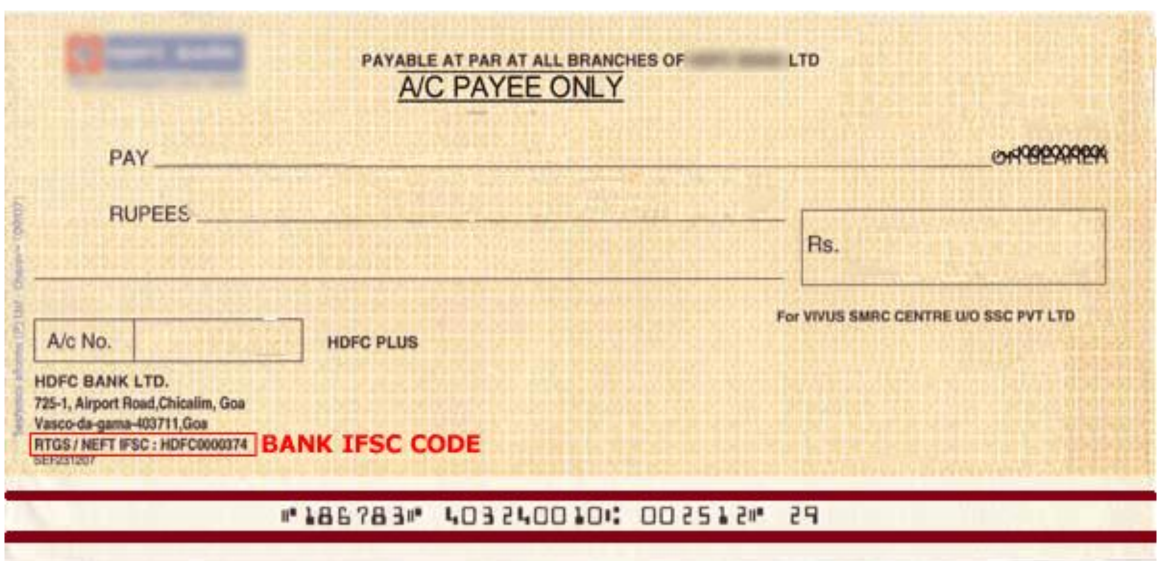


Figure 3.1 Line_segmentation

Input Parameters:

A bitmap image, Binarization_output, is given as input to the Line_segmentation module.

Output Parameters:

A bitmap image say, Line_segmentation_output, and a text file say, line_points, is produced as output from this module.

The pseudo code for Line_segmentation is shown below.

```
Line_segmentation (Binarization_output)
        For Y=0 to Height
        {
                Set Pixelcount = 0
                For     X=0 to Width
                        {
                                Set Color C = Get pixel value of (X, Y)
                                If black then
                                Set Pixelcount = Pixelcount + 1
                        }
                Add Pixelcount values to a file namely, Pixelvalues.
         }
        For P= 0 to Pixelvalues.count
        {
          Set Start_line = P
                While ((P < Pixelvalues.count) and (Pixelcount[P]>0))
                        Set P=P+1
                        Set End_line = P
                        If (Start_line! = End_line)
                                Write Start_line and End_line points in the file,
                        namely line_points.
        }
Return Line_segmentation_output
End of Line_segmentation.
```

Intermediate Parameters:

Height - Height of input cheque image.

Width - Width of input cheque image.

Pixelcount - Number of black pixels count.

The image which is obtained as output after application of the Line_segmentation module is Line_segmentation_output which is referred as figure 4.3 in the next chapter.

MODULE 1.3: Word_segmentation (Line_segmentation_output, line_points)

Our application requires only MICR code, which is a 9 digit code, so to extract this 9 digit code from the extracted line using the module 1.2, it will be look like as shown in figure 3.2 below, by using the Word_segmentation module. The current module uses a technique, as stated in Kumar and Singh [2011].

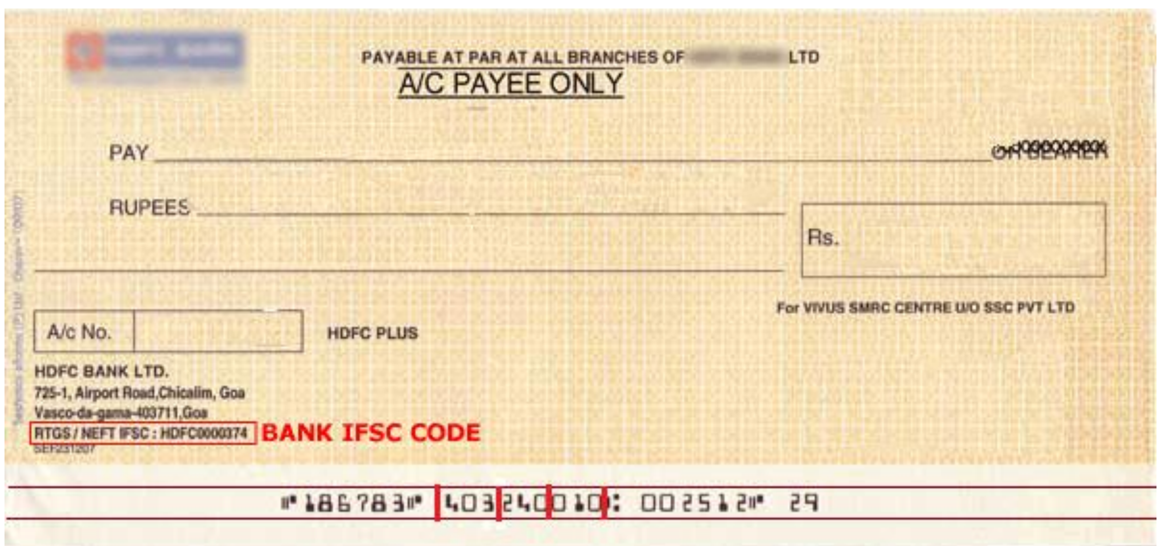


Figure 3.2 Word_segmentation

This module is counting the number of black pixels for the complete width corresponding to the height of the extracted line, and on the basis of the difference between the two adjacent black pixel counts, each word of three digits, is extracted.

Input Parameters:

A bitmap image, Line_segmentation_output, and a text file, line_points is given as input to the Word_segmentation module.

Output Parameters:

A bitmap image say, Word_segmentation_output, and a text file say, word_points, is produced as output from this module.

The pseudo code for Word_segmentation is shown below.

```
Word_segmentation (Line_segmentation_output, line_points)
 For P = 0 to Width
 {
        Set Counter = 0
        Read the second last point say point1 and last point say point2 from the file
        namely, line_points.
        For Z = Point1.Y to Point2.Y
               {
                      Set Color C = Get pixel value of (P,Z)
                      If C is black
                             Set Counter = Counter + 1
               }
        Add Counter values to a file namely, Counter_values.
}
For I = 0 to Counter_values.count
{
        Set d = value at (I+1) – value at I
        Set counter =0
        If ((d > Pk) and counter < 2)
               If (counter = 0)
                      Set Starting_point = value at I+1
                      Set counter = counter +1
               Else
                      Set Ending_point = value at I+1
                      Set counter = counter +1
                      Add values at Starting_point and Ending_point to a file
                      namely, word_points.
 }
 Return Word_segmentation_output
 End of Word_segmentation.
```

Intermediate Parameters:

Width - Width of input cheque image.

Counter- Number of black pixels count.

Starting_point- Start point coordinates of extracted word.

Ending_point- End point coordinates of extracted word.

We have taken value of Pk as 9 after studying number of cheques.

The image which is obtained as output after application of the Word_segmentation module is Word_segmentation_output which is referred as figure 4.4 in the next chapter.

MODULE 1.4: Digit_segmentation (Word_segmentation_output, word_points)

To extract each digit from the extracted word, Digit_segmentation module has been used. This module is counting the number of black pixels for the complete word corresponding to the height of the extracted word, and on the basis of the difference between the two adjacent black pixel counts, each digit from a word, is extracted. After extraction it will look like as shown in figure 3.3 below. The current module uses a technique, as stated in Kumar and Singh [2011].

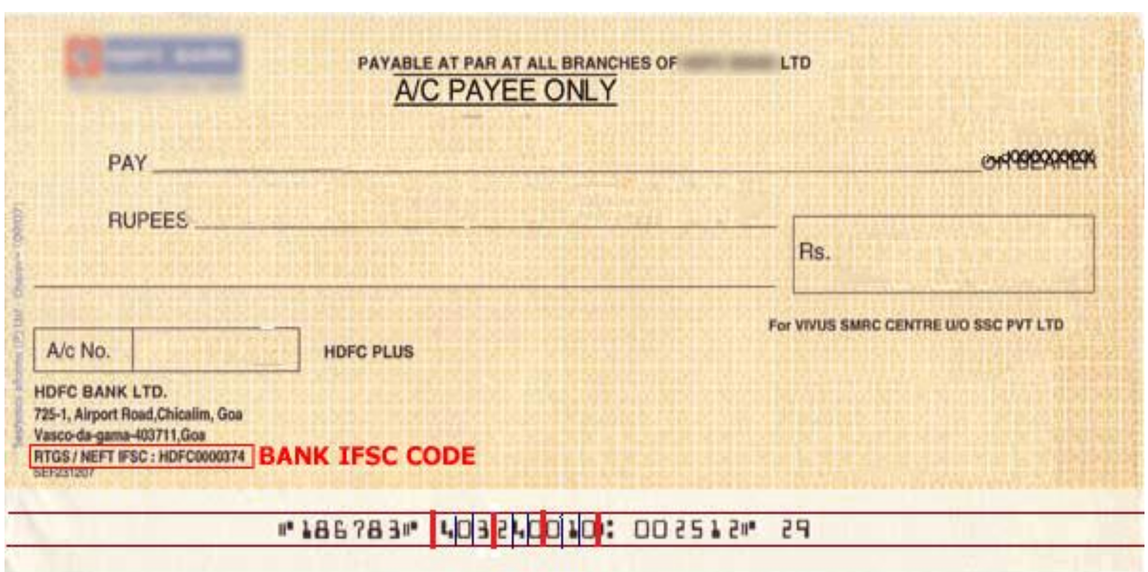


Figure 3.3 Digit_segmentation

Input Parameters:

A bitmap image, Word_segmentation_output, and a text file word_points, is given as input to the Digit_segmentation module.

Output Parameters:

A bitmap image say, Digit_segmentation_output, and a text file say, digit_points, is produced as output from this module.

The pseudo code for Digit_segmentation is shown below.

```
Digit_segmentation (Word_segmentation_output, word_points)
Read the values of Starting_point and Ending_point from a file namely, word_points
For I = Starting_point.X to Ending_point.X
{
        Set D = difference between two adjacent values from the file, word_points.
        If (D > Pr)
                Write d_start, d_end points in a file namely, digit_points.
        Using d_start and d_end points, draw a line.
}
Return Digit_segmentation_output
End of Digit_segmentation.
```

Intermediate Parameters:

Starting_point- Start point coordinates of extracted word.

Ending_point- End point coordinates of extracted word.

d_start- Start point coordinates of extracted digit.

d_end- End point coordinates of extracted digit.

We have taken value of Pr as 2 after studying number of cheques.

The image which is obtained as output after application of the Digit_segmentation module is Digit_segmentation_output which is referred as figure 4.5 in the next chapter.

MODULE 1.5: Resizing (Digit_segmentation_output, digit_points)

After extracting each digit, authors found that dimension of each digit vary with the cheques of different banks, due to which there is no pattern common among the digits through which they can be recognized.

The matrix before applying resizing module for digit 3 and digit 4 is shown below:-

1	1	2
0	0	1
1	1	1

4	4	7
5	4	9
4	4	9

0	1	3
0	1	3
0	1	3

1	2	1
1	0	1
0	1	1

3	2	2
0	0	1
3	2	0

Figure 3.4 (a) Matrices for digit 3

2	1	0
4	0	0
4	0	2

3	0	0
4	0	0
4	0	2

5	0	0
6	0	1
2	2	6

12	4	0
13	13	12
0	2	7

3	2	3
3	0	3
3	2	3

Figure 3.4 (b) Matrices for digit 4

As the above matrices shows that there is no common pattern, thus, authors developed an algorithm, Resizing, based on the use of graphics in OCR.

Resizing is the process of expanding or compressing the dimensions of an object. Positive resizing constants R_x and R_y are used to describe changes in length with respect to x and y direction, respectively. To make all extracted digits of same dimension, Resizing module is used.

Input Parameters:

A bitmap image, Digit_segmentation_output, and a text file word_points, is given as input to the resizing module.

Output Parameters:

A bitmap image say, Resized_image, corresponding to each digit is produced as output from this module.

The pseudo code for Resizing is shown below.

```
Resizing (Digit_segmentation_output, digit_points)
Find out Resizing factor using the formula:
Set Ry = 12/Image Height
Set Rx = 8/ Image width
For X = 0  to Height
{
        For Y = 0 to  Width
                {
                        Set Color C = Get Pixel value of (X,Y)
                        Set New_x = Rx * d_start
                        Set New_y = Ry*d_end
                        If C is black then
                                Set Pixel value at New_x and New_y as Black
                        Else
                                Set Pixel value at New_x and New_y as White.
                }
}
Return Resized_image
End of Resizing.
```

Intermediate Parameters:

Height - Height of extracted resized image representing single digit.

Width - Width of extracted resized image representing single digit.

R_x and R_y denotes resizing factor.

The image which is obtained as output after application of the Resizing module is Resized image which is referred as figure 4.6 in the next chapter.

MODULE 1.6: Grid_making (Resized_image, 0, 0, Resized_image_width-1, Resized_image_height-1)

For each resized image, a matrix of 6 × 4 is created and for each grid, number of black pixels are counted through Grid_making module.

Input Parameters:

A bitmap image, Resized_image, is given as input to the Grid_making module.

Output Parameters:

A text file say, grid_file, corresponding to each digit is produced as output from this module.

The pseudo code for Grid_making is shown below.

```
Grid_making (Resized_image, 0, 0, Resized_image_width-1, Resized_image_height-1)
Set srt_w = minx
Set srt_h = miny
Set increment_width = width / 4
Set increment_height = height / 6
For Z = 1 to 6
{
        Set grid_width = Image width/ 4
        For K = 1 to 4
        {
                Set Pixelcount = 0
                For Y = miny to (grid_height + srt_h)
```

```
                {
                        For  X = minx to (grid_width + srt_w)
                        {
                                Set Color colr = Get pixel value of (X, Y);
                                If  black then
                                Set Pixelcount = Pixelcount +1
                        }
                }
                Add Pixelcount values to a file say grid_file.
                Set minx = grid_width + srt_w
                Set grid_width = increment_width * (k + 1);
        }
        Set miny = grid_height + srt_h;
        Set grid_height = increment_height * (z + 1);
        Set minx = srt_w;
}
End of Grid_making.
```

Intermediate Parameters:

Minx - denotes the minimum x coordinate value of the image.

Maxx - denotes the maximum x coordinate value of the image.

Miny - denotes the minimum y coordinate value of the image.

Maxy - denotes the maximum y coordinate value of the image.

Pixelcount - Number of black pixels count.

The file which is obtained as output after application of the Grid_making module is grid_file which is referred as figure 4.7 in the next chapter.

MODULE 1.7: Pattern_recognition (grid_file)

For each resized image, a matrix of 6×4 is created and for each grid, number of black pixels are counted through Grid_making module. After applying the above algorithms on a number of cheques, matrices are studied and tried to find out common pattern for all the digits from 0 to 9. This module will look for the white pixels, and on the basis of that each digit can be recognized.

Input Parameters:

A text file, grid_file, is given as input to the Pattern_recognition module.

Output Parameters:

A text file say, numbers, is produced as output from this module.

The pseudo code for Pattern_recognition is shown below.

```
Pattern_recognition (grid_file)
If(C [13] == 0 and C [14] ==0 and C [23] ] == 0 and C [24] ] == 0 and C [61] ] == 0 and C [62] )
Numeral = 4
Else if ((C [21] == 0 and C [22] ==0) and ((C [43] == 0 and C [44] ==0) or(C [53] == 0 and C [54] ==0))
Numeral = 2
Else if (C [21] == 0 and C [22] ==0 and C [51] == 0 and C [52] == 0)
Numeral = 3
Else if (C [31] == 0 and C [41] ==0 and C [51] == 0 and C [61] == 0)
Numeral = 7
Else if (C [22] ==0 and C [51] == 0 and C [52] ==0 and C [61] == 0 and C [62] == 0)
Numeral = 9
Else if (((C [23] == 0 and C [24] ==0) or (C [33] == 0 and C [34] == 0)) and ( (C [41] == 0 and C [42] ==0)
or (C [51] == 0 and C [52] == 0)))
Numeral = 5
Else if (C [23] == 0 and C [24] ==0 and C [52] == 0 and C [53] == 0)
Numeral = 6
Else if (C [13] == 0 and C [14] ==0 and C [23] == 0 and C [24] == 0)
Numeral = 1
Else if (C [22] == 0 and C [52] ==0)
Numeral = 8
Else if ((C [32] == 0 and C [33] ==0 and C [42] == 0 and C [43] == 0) or (C [22] == 0 and C [32] ==0 and
C [42] == 0)
Numeral = 0
```

```
If numeral is not from 0 to 9 then
        Call Thinning (Resized_image)
        // The output from Thinning is saved as Thinning_output.
        Call Grid_making (Thinning_output, 0, 0, Thinning_output_width-1, Thinning_output_height-1)
        Call Pattern_recognition(grid_file).
Write numeral on a file namely, numbers.
End of Pattern_recognition.
```

Intermediate Parameters:

From grid_making module a matrix of 6×4 is obtained. Each grid of a matrix is numbered from C11 to C64, figure 3.5, where C stands for cell number.

C11	C12	C13	C14
C21	C22	C23	C24
C31	C32	C33	C34
C41	C42	C43	C44
C51	C52	C53	C54
C61	C62	C63	C64

Figure 3.5 Matrix of 6×4 representing each cell

The file which is obtained as output after application of the Pattern_recognition module is numbers which is referred as figure 4.10 in the next chapter.

MODULE 1.7.1: Thinning (Resized_image)

The text in the document image may have the thickness of several pixels depending on the type of the writing instrument. The reduction of thickness of text will reduce the amount of data and helps to extract the shape information of the character. The module consists of two sub iterations – one deletes the north west boundary points and other deletes the south east corner points, as stated in Zhang and Suen [1984].

Input Parameters:

A bitmap image, Resized_image, is given as input to the Thinning module.

Output Parameters:

A bitmap image say, Thinning_Output, is produced as output from this module.

The pseudo code for Thinning is shown below.

```
Thinning(Resized_Image)
Set counter=0
While(counter=0)
    For X=Min_X to Max_X
        For Y=Min_Y to Max_Y
            If(C1 is true)
                Set Pixel_Intensity(X,Y)=WHITE
                Set counter=counter+1
If(counter=0)
    Goto label A
Set counter=0
    For X=Min_X to Max_X
        For Y=Min_Y to Max_Y
            If(C1 is true)
                Set Pixel_Intensity(X,Y)=WHITE
                Set counter=counter+1
If(counter=0)
    Goto label A
A: return Thinning_Output
```

Intermediate Parameters:

C1 and C2 define the set of conditions as described here:-

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure 3.6 Neighborhood Matrix of pixel (P1)

C1 has the following set of conditions:-

1. 2 <= B (P1) <= 6
2. A (P1) =1
3. P2*P4*P6=0
4. P4*P6*P8=0

C2 has the following set of conditions:-

1. 2 <= B (P1) <= 6
2. A (P1) =1
3. P2*P4*P8=0
4. P2*P6*P8=0

A (P1) defines the number of 01 transition in the ordered set P2, P3, P4…..P9.

B (P1) defines the number of non zero neighbors of P1.

B (P1) = P2+P3+P4……+P9.

The thinned image which is obtained as output after application of the Thinning module is Thinning_output which is referred as figure 5.8 in the next chapter.

MODULE 1.8: Cheque_details (numbers)

A database is created, consisting of three tables *i.e.*, citydetails, bankdetails, and branchdetails. Citydetails consist of two attributes *i.e.*, city_code and city_name, bankdetails consist of two attributes *i.e.*, bank_code and bank_name and branchdetails consist of two attributes *i.e.*, branch_code and branch_name.

City_name is extracted with the help of city_code, bank_name is extracted with the help of both city_code and bank_code and branch_name is extracted with the help city_code, bank_code and branch_code.

Each word is matched from the database digit by digit, and corresponding to that city name, bank name and branch name is extracted.

Input Parameters:

A text file, numbers, is given as input to the Cheque_details module.

Output Parameters:

A text file say, city_bank_branch, is produced as output from this module.

```
Cheque_details (numbers)
Set city_code = Read the first three digits from the file namely, numbers.
Set bank_code = Read the next three digits from the file namely, numbers.
Set branch_code = Read the next  three digits from the file namely, numbers.
Set city_name = Extract city name from the database based on the city_code and write in
file namely, city_bank_branch.
Set bank_name = Extract bank name from the database based on the bank_code and
city_code and write in file city_bank_branch.
Set branch_name = Extract area name from the database based on the city_code,
bank_code and branch_code and write in file city_bank_branch.
End of Cheque_details.
```

The file which is obtained as output after application of the Cheque_details module is city_bank_branch which is referred as figure 4.11 in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Implementation

The authors provide a generalized approach for the MICR code processing. In it number of modules is described for extracting and recognizing the digits in the MICR code. Started with one of the preprocessing technique, binarization, changes the scanned image of cheque into black and white mode. After extracting the line in which the MICR code is present, each digit is extracted. After analyzing it was found that dimension of each digit varies across the cheques. Thus the approach failed to recognize the digits. Then an approach, Resizing is proposed which convert the dimension of each digit to a specific dimension. This novel approach helps in removing the bottleneck in the MICR code processing and helps in recognizing the MICR code digits with a satisfactory performance.

The results obtained after implementation of all the algorithms are discussed in the next section.

4.2 Output Images

The image acquired through scanner is shown in Figure 4.1.

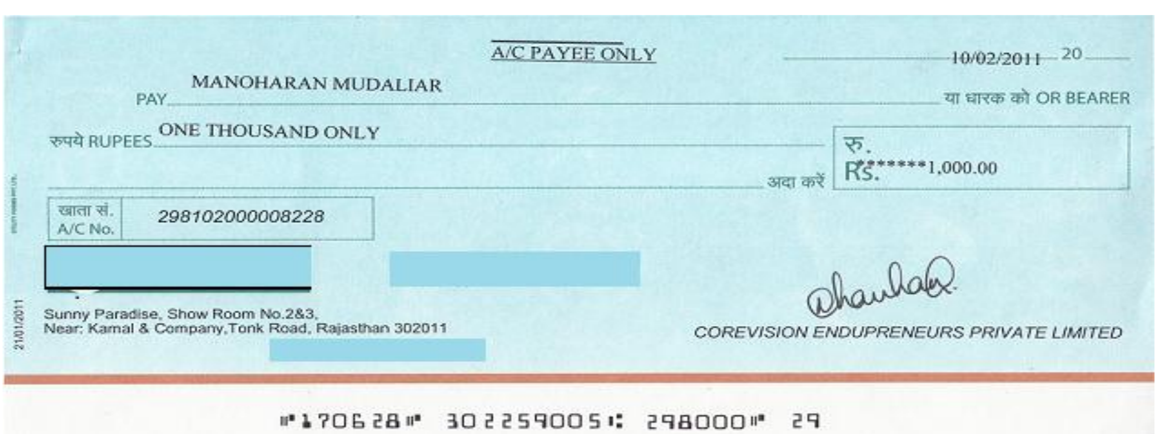


Figure 4.1 Cheque_Image

It can be seen in figure 4.1 that the background color is present in the scanned image and thus, it requires preprocessing technique *i.e.*, binarization. Binarization is applied to the Cheque_image. It can be observed that the textured and colored background present in the figure 4.2 is now removed.

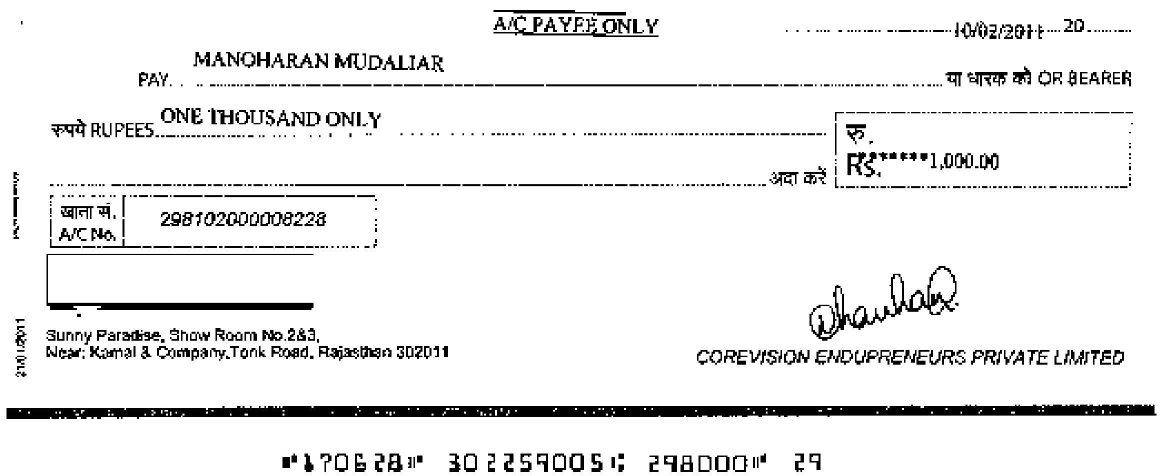


Figure 4.2 Binarization_output

The next step is to extract the line from the cheque which is present at the bottom of the cheque and thus Line_segmentation is applied to the Binarization_output image. Desired line is extracted as shown in figure 4.3.

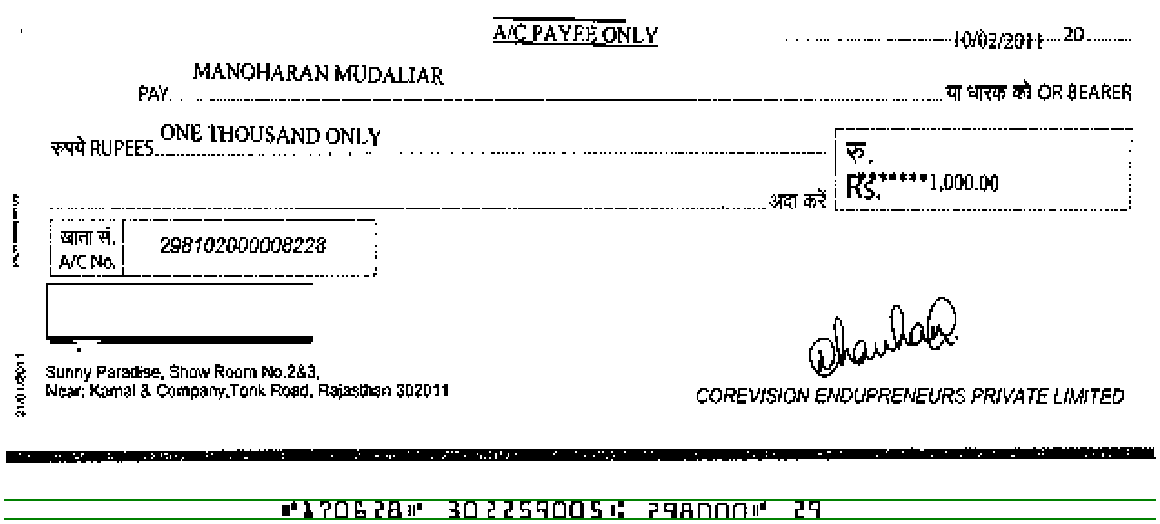


Figure 4.3 Line_segmentation_output

The three words (consist of three digit each) from the MICR code are now extracted from the line extracted in the previous step, as shown in figure 4.4.

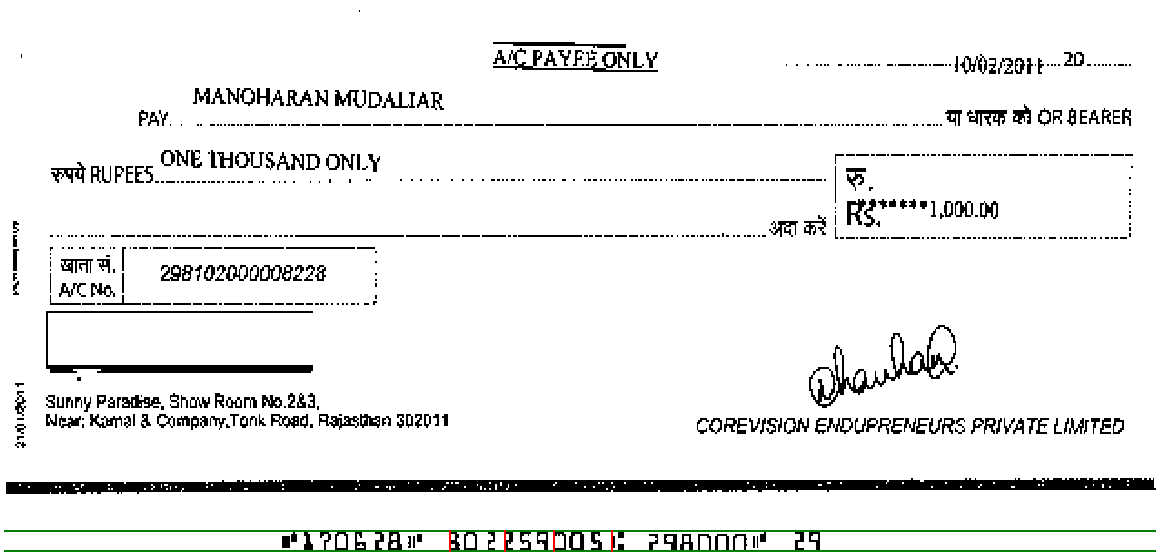


Figure 4.4 Word_segmentation_output

The three digits (for each word) from the MICR code are now extracted from the word extracted in the previous step, as shown in figure 4.5.

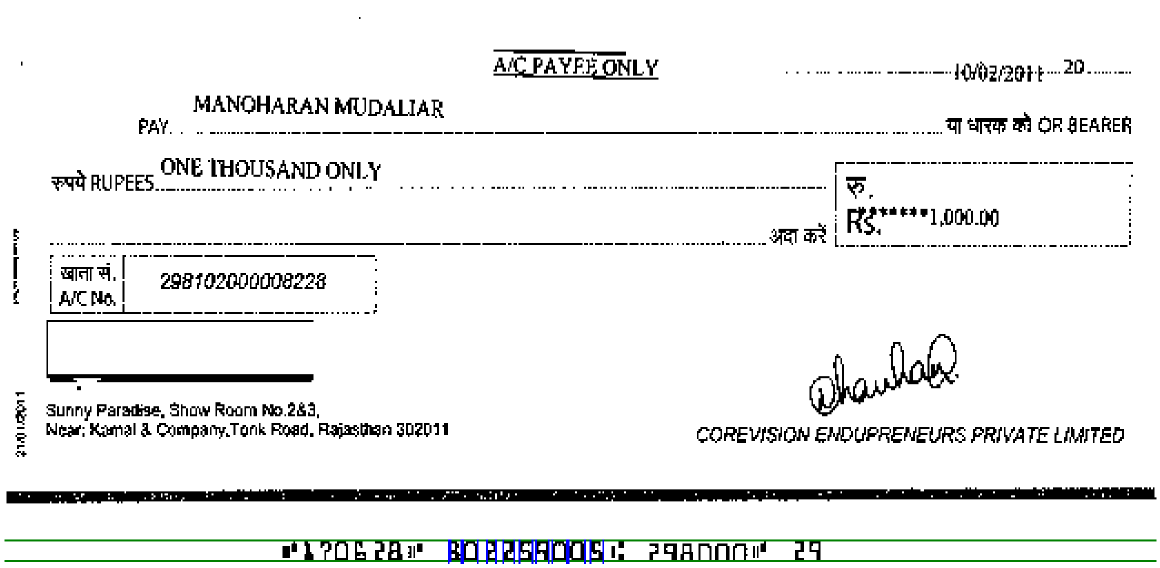


Figure 4.5 Digit_segmentation_output

Original dimension of each extracted digit was:-

DIGIT	DIMENSION(L×B)
3	6×10
0	8×10
2	5×10
2	6×10
5	7×10
9	7×10
0	8×10
0	9×10
5	7×10

Table 4.1 Original dimension of each digit

Now each digit in the previous step is resized to same dimension *i.e.* 8×12 by applying Resizing as shown in figure 4.6.

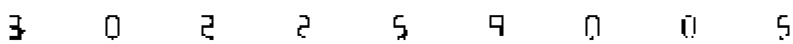


Figure 4.6 Resized_image

For each digit, a matrix having the total number of black pixel count is shown in Figure 4.7.

2	2	3	0
0	0	2	0
2	2	2	0
1	2	4	2
0	1	4	1
4	4	4	0

Figure 4.7 (a) Matrix for 3

3	2	3	1
2	0	0	2
2	0	0	2
2	0	0	2
2	0	0	2
1	3	2	0

Figure 4.7 (b) Matrix for 0

2	2	4	0
0	0	2	0
2	2	4	0
2	0	0	0
3	0	0	0
0	2	4	0

Figure 4.7 (c) Matrix for 2

1	2	3	0
0	0	2	0
1	2	3	0
3	0	0	0
3	0	0	0
0	2	0	0

Figure 4.7 (d) Matrix for 2

3	2	1	0
2	0	0	0
3	2	2	0
0	0	2	1
0	0	3	2
0	3	3	0

Figure 4.7 (e) Matrix for 5

3	2	2	2
2	0	0	2
3	2	2	2
0	0	2	2
0	0	2	2
0	0	0	0

Figure 4.7 (f) Matrix for 9

3	2	3	1
2	0	0	2
2	0	0	2
2	0	0	2
2	0	0	2
0	3	0	0

Figure 4.7 (g) Matrix for 0

0	2	1	1
4	0	0	2
4	0	0	2
4	0	0	2
3	0	0	1
0	1	1	0

Figure 4.7 (h) Matrix for 0

3	2	1	0
2	0	0	0
3	2	2	0
0	0	2	0
0	0	3	0
2	1	0	0

Figure 4.7 (i) Matrix for 5

In these matrices, it is not able to recognize digit 3, as matrix of 3 is not matched with the found pattern. Thus, it will call thinning program. After applying thinning on digit 3, its image is shown in figure 4.8.

Figure 4.8 Thinning_output

Now after thinning, again grid_making module is called for thinning_output, its grid is shown in figure 4.9.

1	2	2	0
0	0	2	0
2	2	2	0
0	0	4	0
0	0	4	0
2	4	0	0

Figure 4.9 Matrix for 3 after thinning

After applying Pattern_recognition algorithm, digits recognized by this algorithm based on the matrices above is shown in Figure 4.10.

3 0 2 2 5 9 0 0 5

Figure 4.10 Numbers

Now after recognizing digits, they are matched with the database by calling cheque_details and fetch the city_bank_branch, as shown in figure 4.11.

JAIPUR IDBI GAUTAM MARG

Figure 4.11 city_bank_branch

4.3 Results

The algorithm for the detection and the recognition of MICR code has been applied on a number of bank cheques. The results showing the number of digits recognized accurately has been tabulated in table 4.2.

Cheques	Digits Recognized	Accuracy (%)
Cheque1	9	100
Cheque2	9	100
Cheque3	8	88.88
Cheque4	9	100
Cheque5	9	100

Table 4.2 Recognition Results for MICR code

The comparison between the accurately recognized and inaccurately recognized digits of MICR code in bank cheques has been made through graph shown in figure 4.12.

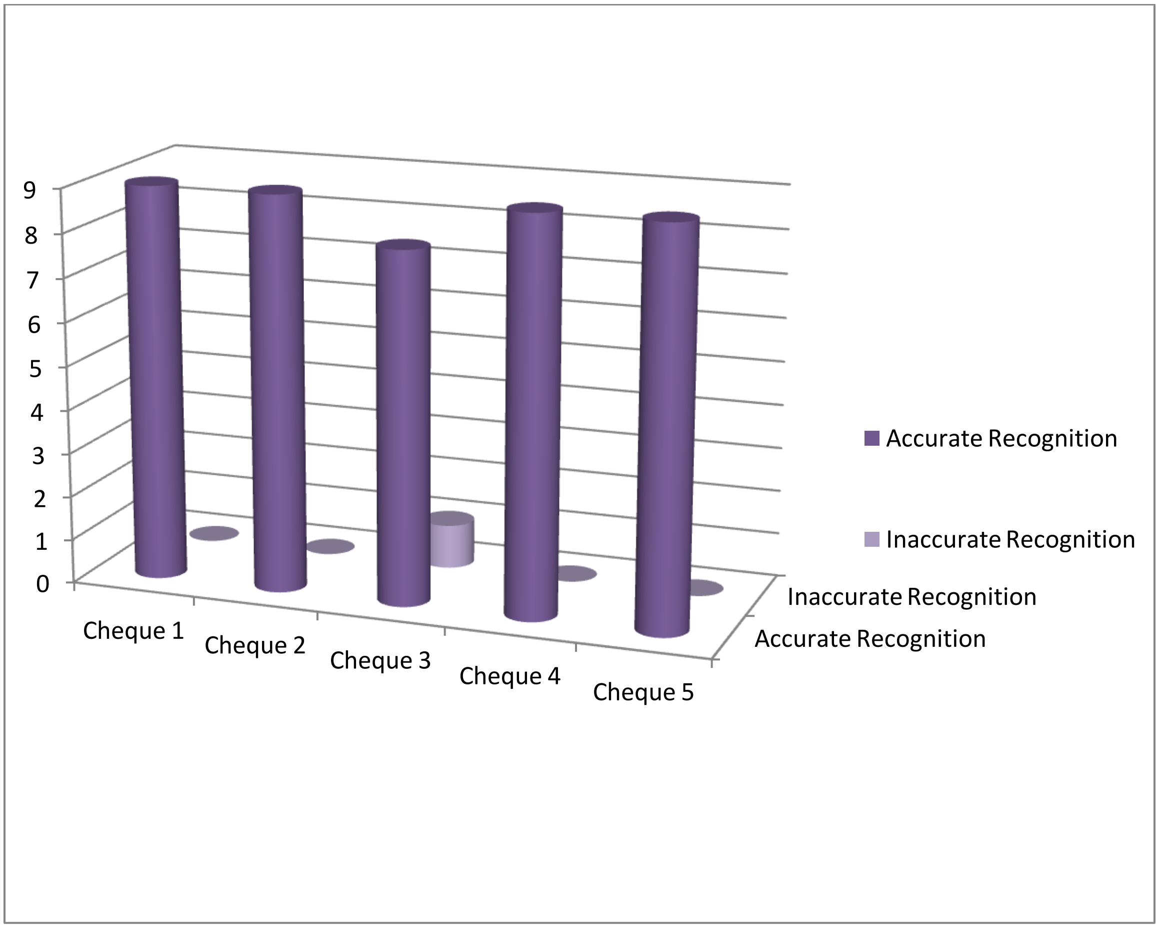


Figure 4.12 Comparison of accurately recognized versus inaccurately recognized digits in MICR code.

It can be observed from figure 4.12 that there is some inaccurate recognition. The reason for this could be identified that there is noise present on the MICR band in the cheque which, leads to distortion of the image when the binarization process is applied.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

It has been observed that a bank transaction which involves a use of bank cheques is increasing rapidly as the population is growing. MICR code which is present at the bottom of the cheque gives important information about the name of the city, bank name and the name of the branch, which has issued that cheque. This code is unique for any bank-branch transaction. For reading the MICR code banks use special type of scanner which has certain disadvantages like MICR scanners are very precisely designed machines that are expensive and need constant maintenance to maintain their reliability. In writing this code in cheques, special type of magnetic ink is needed which is very costly. MICR scanners read the magnetic ink particles, any damage to MICR character, will hamper its performance.

The initial phase of discovering a new approach for MICR code processing was quite tough. Through the literature survey, the authors found that no approach other than the usage of special MICR scanners is present for the processing of the MICR code. It was challenging for the authors to give a new dimension to the processing of MICR code. The concept of OCR was thus used to develop the new approach for detecting and recognizing the MICR code. Initially matrices were created for each digit in the MICR code to find the common pattern for recognition but, this approach failed due to the varying dimension of each digit of MICR code across the cheques. To solve this problem, an effort was made to convert the digits of the MICR code to a specific dimension. Resizing algorithm calculates the new coordinates of each digit, and multiplies them with the original coordinates, thus changing the dimensions of each digit. The approach for recognition without Resizing became a bottleneck in the MICR code recognition process. Thus, Resizing plays an important role in MICR code recognition and its performance greatly affects the recognition result. Our approach is giving the satisfactory recognition rate because of proposing Resizing approach.

5.2 FUTURE SCOPE

So far an effective, efficient and generalized approach for detecting and recognising MICR code has been proposed, we need to further enhance the study in order to improve the MICR code recognition performance. The various dimensions for the future work are discussed here.

- All other information present in the MICR band i.e., cheque number, few digits of account number and transaction id can be extracted and recognized.
- MICR code present on different types of forms can be recognized.
- From various barcode, product code on products this code can be extracted and recognized.
- The proposed approach can also be extended for demand draft, traveler cheques.

REFERENCES

1. Alginahi, Y. M., A survey on Arabic character segmentation, *International Journal on Document Analysis and Recognition*. Retrieved June 2, 2012, http://www.springerlink.com/content/t276077j5kj1u516 .

2. Allier, B. and Emptoz, H., Character Prototyping in Document Images Using Gabor Filters. in *Proceedings of the 9th International Conference on Image Processing*, (2003), 537-540.

3. Arica, N. and Vural, F. Y. 2001. An Overview of Character Recognition focused on Off-line Handwriting. *IEEE Transactions on Systems, Man, and* Cybernetics, 31(2), 216-233.

4. Bansal, V. And Sinha, R. M. K. 2002. Segmentation of Touching and Fused Devanagari Characters. *Pattern Recognition*, 35(4), 875-893.

5. Blumenstein, M. And Verma, N., A New Segmentation Algorithm for Handwritten Word Recognition. in *Proceedings of International Joint Conference on Neural Networks*, (1999), 2893-2898.

6. Bremananth, R., Sankari, M. and Benazir, M., Verification of Bank Cheque Images using Hamming Measures. in *proceedings of the 11th International Conference on Control, Automation, Robotics and Vision*, (2010), 2531-2536.

7. Bunke, H. and Wang, P. S. P. *Handbook of Character Recognition and Document Image Analysis*. World Scientific Publishing, Singapore, 1997.

8. Cai, J. And Liu, Z. 1999. Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3), 263-270.

9. Casey, R. and Lecolinet, E. 1996. A Survey of Methods and Strategies in Character Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(7), 690-706.

10. Cheriet, M., Kharma, N., Liu, C. and Suen, C. Y. *Character recognition by computer and applications*. A John Wiley & Sons, New Jersey, 1986.

11. Farooq, F., Govindaraju, V. and Perrone, M. P., Pre-processing Methods for Handwritten Arabic Documents. in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, (2005), 267-271.

12. Fujisawa, H., A View on the Past and Future of Character and Document Recognition. in *Proceedings of the 9th International Conference on Document Analysis and Recognition*, (2007), 3-7.

13. Guillevic, D. and Suen, C. Y. 1998. Recognition of Legal Amounts on Bank cheques. *Pattern Analysis and Applications*, 1(1), 28-41.

14. Ha, T. M. and Bunke, H. 1997. Off-Line, Handwritten Numeral Recognition by Perturbation Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), 535-539.

15. Hoffman, R. L. and McCullough, J. W. 1971. Segmentation Methods for Recognition of Machine-Printed Characters. *IBM Journal of Research and* Development, 15(2), 153-165.

16. Jain, A. K., Duin, R. P. W., and Mao, J. 2000. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37.

17. Jayadevan, R., Kolhe, S. R., Patil, P. M. and Pal, U. 2011. Automatic processing of handwritten bank cheque images: a survey, *International Journal of Document Analysis and Recognition*. Retrieved June 2, 2012, http://www.springerlink.com/content/r27330445207r163.

18. Kannan, R. J. and Prabhakar, R. 2009. A Comparative Study of Optical Character Recognition for Tamil Script. *European Journal of Scientific Research*, 35(4), 570-582.

19. Kasturi, R., Gorman, L. and Govindaraju, V. 2002. Document Image Analysis: A Primer. *S¯adhan¯a* , 27(1), 3–22.

20. Kumar, R. and Singh, A. 2010. Detection and Segmentation of Lines and Words in Gurmukhi Handwritten Text using Flexible Windowing. *International Journal of Computer Theory and Engineering*, 2(3), 329 - 332.

21. Kumar, R. and Singh, A. 2008. Segmentation of Handwritten Text in Gurmukhi Script. *International Journal of Computer Science and Security*, 2(3), 12-17.

22. Li, Y., Zheng, Y. and Doermann, D., Detecting Text Lines in Handwritten Documents. in *Proceedings of the 18th International Conference on Pattern Recognition*, (2006), 1030-1033.

23. Liu, K., Suen, C. Y. and Nadal, C., Automatic Extraction of Items from Cheque Images for Payment Recognition. in *Proceedings of the 13th International Conference on Pattern* Recognition, (1996), 798-802.

24. Lu, Y. 1995. Machine Printed Character Segmentation-An Overview. *Pattern Recognition*, 28(1), 67-80.

25. Majumdar, A. and Chaudhuri, B. B., A MLP Classifier for both Printed and Handwritten Bangla Numeral Recognition. in *proceeding of the 5th Indian Conference proceeding on Computer vision*, *Graphics and Image Processing*, (2006), 796-804.

26. Mo, S. and Mathews, J. 1998. Adaptive, Quadratic Preprocessing of Document Images for Binarization. *IEEE Transactions on Image Processing*, 7(7), 992-999.

27. Morita, M., Yacoubi, A. E., Sabourin, R., Bortolozzi, F. and Suen, C. Y., Handwritten Month Word Recognition on Brazilian Bank Cheques. in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, (2001), 972-976.

28. Nagy, G., Nartker, T. A. and Rice, S. V., Optical Character Recognition: An Illustrated Guide to the Frontier. in *proceedings of Document Recognition and Retrieval*, (1999), 58-69.

29. Ohali, Y. A., Cheriet, M. and Suen, C. 2003. Databases for recognition of handwritten Arabic cheques. *Pattern Recognition*, 36(01), 111-121.

30. Pal, U. and Chaudhuri, B. B. 2004. Indian Script Character Recognition: A Survey. *Pattern Recognition*, 37(9), 1887-1899.

31. Pal, U., Sinha, S. and Chaudhuri, B. B., Multi-Script Line Identification from Indian Documents. in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, (2003), 880-884.

32. Patil, U. and Begum, M. 2012. Word level handwritten and Printed Text separation based on shape features. *International Journal of Emerging Technology and Advanced Engineering*, 2(4), 590-594.

33. Plamondon, R. and Srihari, S. N. 2000. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 63-84.

34. Prasad, J. R., Kulkarni, U. V. And Prasad, R. S., Offline Handwritten Character Recognition for Gujrati Script using Pattern Matching. in *Proceedings of the 3rd international conference on Anti-Counterfeiting, security, and identification in communication*, (2009), 611-615.

35. Seethalakshmi, R., Sreeranjani, T. R., Balachandar, T., Singh, A., Singh, M., Ratan, R. and Kumar, S. 2005. Optical Character Recognition for printed Tamil text using Unicode. *Journal of Zhejiang University Science*, 6(11), 1297-1305.

36. Senior, A. and Robinson, A. 1998. An Off-line Cursive Handwriting Recognition System. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 309-312.

37. Siromoney, G., Chandrasekaran, R. and Chandrasekaran, M. 2003. Computer Recognition of Printed Tamil Characters. *Pattern Recognition*, 10(4), 243-247.

38. Shaaban, Z., A New Recognition Scheme for Machine- Printed Arabic Texts based on Neural Networks. in *proceeding of World Academy of Science*, (2008), 707-710.

39. Timar, G., Karacs, K. and Rekeczky, C., Analogic Preprocessing and Segmentation Algorithms for Off-line Handwriting Recognition. in *Proceedings*

of the 7th IEEE International Workshop on Cellular Neural Networks and their Applications, (2002), 407- 414.

40. Trier, O. D., Jain, A. K. and Taxt, T. 1996. Feature Extraction Methods for Character Recognition - A Survey. *Pattern Recognition*, 29(4), 641-662.

41. Wen, Y., Lu, Y., Yan, J., Zhou, Z., Deneen, K. M. V. and Shi, P. 2011. An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems*, 12(3), 830-845.

42. Wu, V. And Manmatha, R., Document Image Clean-Up and Binarization. in *Proceedings of Society for PhotoOptical Instrumentation Engineers (SPIE)*, (1997), 263-273.

43. Ye, X., Cheriet, M., Suen, C. Y. and Liu, K. 1999. Extraction of bank check items by mathematical morphology. *International Journal on Document Analysis and Recognition*, 2(2-3), 53-66.

44. Zhang, T. Y. and Suen, C. Y. 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. *Communications of the ACM*, 27(3), 236-239.

45. Zhao, S. X. and Srihari, S. N., A Word Recognition Algorithm for Machine-printed Word Images. in *Proceedings of the third International Conference on Document Analysis and Recognition*, (1995), 351-354.

46. Zheng, L., Hassin, A. H. and Tang, X. 2004. A new algorithm for machine printed Arabic character segmentation, *Pattern Recognition Letters*, 25(15), 1723-1729.