Tribhuvan University
Institute of Science and Technology

An Image Processing Oriented Optical Mark Reader

Project Report
Submitted to

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

Submitted By
Er. Ramesh Neupane
Roll No. 23

Tribhuvan University
Institute of Science and Technology

An Image Processing Oriented Optical Mark Reader

Project Report
Submitted to

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

Submitted By
Er. Ramesh Neupane
Roll No. 23

Under the Supervision of
Asst. Professor Dr. Sanjeeb Prasad Panday

Supervisor’s Recommendation

I hereby recommend that this project prepared under my supervision by **Er. Ramesh Neupane** entitled **“An Image Processing Oriented Optical Mark Reader”** in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

… … … … … … … … … … … … … … … …

Asst. Prof. Dr. Sanjeeb Prasad Panday
Department of Computer & Electronics Engineering,
Institute of Engineering,
Pulchowk, Nepal

Date: 20th Dec, 2012

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science & Information Technology

Letter of Approval

We certify that we have read this project report and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

… … …… … … … … … …

Prof. Dr. Tanka Nath Dhamala
Central Department of Computer Science
& Information Technology,
Tribhuvan University,
Kathmandu, Nepal
(Head)

… … …… … … … … … …

Asst. Prof. Dr. Sanjeeb Prasad Panday
Department of Computer
& Electronics Engineering,
Institute of Engineering,
Pulchowk, Kathmandu, Nepal
(Supervisor)

. … … … … … … … … …

(External Examiner)

… … … … … … … …

(Internal Examiner)

Date: 20th Dec, 2012

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science & Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than he listed here have been used in this work.

… … … … … … … …
Er. Ramesh Neupane
Date: 20th Dec, 2012

ACKNOWLEDGEMENTS

First and above all, I thank God for his love and guidance of all the time. I submit my gratitude to the Almighty who gave ability, strength and confidence in me to complete this work. Without his mercy this would never come to be today’s reality. My life is so blessed because of him.

With deep sense of gratefulness I express my genuine thanks to my respected and worthy supervisor **Asst. Prof. Dr. Sanjeeb Prasad Panday**, Department of Electronics & Computer Engineering (Kathmandu, Nepal) for his valuable guidance in carrying out this work under his effective supervision and enlightenment.

I would like to thank my respected promoter **Prof. Dr. Tank Nath Dhamala**, Head of Computer Science & IT Department, TU (Kathmandu, Nepal) for his encouragement and guidance.

Thanks to all my close friends for their supports and all kind of help. I also want to convey my thanks to MATLAB developer team and special thanks to Google Search Engine developer team.

Finally, I thank my family for their love, support and encouragement.

ABSTRACT

This project is a development of an optical mark reader that can be used for counting the examination score from the multiple-choice answer sheet. The system is developed based on PC-type microcomputer connecting to an image scanner or web cam. The image of sheet taken from webcam or scanner is converted into binary image using *adaptive binarization method*; skew correction is done by detecting line using *hough transform*. Binary image is cut into Roll number region and Answer region using reference square fixed on answer sheet. The system operations can be distinguished into two models: *learning mode* and *operation mode*. In the l*earning mode*, system separates each static block in a sheet using horizontal and vertical projection after doing morphological operation. Each block is divided into ten equal parts and mark area for each question and mark cells for each option of each question. Operational mode is used to check and decide the answer. For answer checking purpose, the number of black pixels in each answer block is counted and decision is created from comparison of number of black pixel between different cells of each question. Similar process is used to extract Roll number. Score is calculated by comparing correct answer and marked answer. Finally, the list of Roll number and scores can be exported into data sheet.

Keywords: *optical mark reader, edge detection, Hough Transform, Skew detection, Adaptive binarization, Morphological Filtering, vertical and Horizontal Projection.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

BCBP - Bar Code on Boarding Passes

BTC - Batch Transaction Capture

CCD – Charge Coupled Device

DN - Different No of question of images

EAN - European Article Number

EM - External Object Mixed Sheets

FPGA - Field Programmable Gate Array

GUI - Graphical User Interface

IATA - International Air Transport Association

IBM - International Business Machines Corporation

IM - Incorrectly Marked Sheets

IPT - Image Processing Toolbox

IR - Incorrectly Roll No marked Sheets

LCOMR- Low-Cost Optical Mark Reader

OCR - Character Recognition

OMR - Optical Mark Recognition

PC - Personal Computer

QR - Quick Response

SDM - Soft Decision Method

TBM - Text Binarization Method

UPC - Universal Product Code

1 INTRODUCTION

Optical mark recognition (OMR) is the automated process of capturing the data which is in form of some marks like bubbles, squares & horizontal or vertical tick. This is done by contrasting reflectivity at predetermined positions on the sheet. When we shine a beam of light onto the paper, the scanner is able to detect the marked region as it reflects less light than the unmarked area on the paper. In order to be detected by the scanner, the mark should be significantly darker than the surrounding area on paper and it should be properly placed.

1.1 OMR Background

Many traditional OMR devices work with a dedicated scanner device that shines a beam of light onto the form paper. The contrasting reflectivity at predetermined positions on a page is then used to detect the marked areas because they reflect less light than the blank areas of the paper. Some OMR devices use forms which are preprinted onto 'transoptic' paper and measure the amount of light which passes through the paper, thus a mark on either side of the paper will reduce the amount of light passing through the paper. In contrast to the dedicated OMR device, desktop OMR software allows a user to create their own forms in a word processor and print them on a laser printer. The OMR software then works with a common desktop image scanner with a document feeder to process the forms once filled out.

OMR is generally distinguished from optical character recognition (OCR) by the fact that a complicated pattern recognition engine is not required. That is, the marks are constructed in such a way that there is little chance of not reading the marks correctly. This does require the image to have high contrast and an easily-recognizable or irrelevant shape. A related field to OMR and OCR is the recognition of barcodes such as the UPC bar code found on product packaging. One of the most familiar applications of optical mark recognition is the use of #2 pencil (HB in Europe) bubble optical answer sheets in multiple choice question examinations. Students mark their answers, or other personal information, by darkening circles marked on a pre-printed sheet. Afterwards the sheet is automatically graded by a scanning machine. In most European countries, a horizontal or vertical 'tick' in a rectangular 'lozenge' is the most commonly used type of OMR form, the most familiar application being the UK National lottery form. Lozenge marks are a later technology and have the advantage of being easier to mark and easier to erase.

The large ‘bubble’ marks are legacy technology from the very early OMR machines that were so insensitive a large mark was required for reliability. In most Asian countries, a special marker is used to fill in an optical answer sheet. Students likewise mark answers or other information via darkening circles marked on a pre-printed sheet. Then the sheet is automatically graded by a scanning machine.

Many of today’s OMR applications involve people filling in specialized forms. These forms are optimized for computer scanning, with careful registration in the printing, and careful design so that ambiguity is reduced to the minimum possible. Due to its extremely low error rate, low cost and ease-of-use, OMR is a popular method of tallying votes.

1.2 OMR software

OMR Software is a computer software application that makes OMR possible on a desktop computer by using an Image scanner to process surveys, tests, attendance sheets, checklists, and other plain-paper forms printed on a laser printer. The need for OMR software originated because early optical mark recognition systems were designed to use dedicated scanners and special pre-printed forms with drop-out colors and registration marks. OMR Software users design their own mark-sense forms with a word processor or built-in form editor and print them locally on a laser printer saving large amount of money on large numbers of forms.

Identifying optical marks within a form such as for processing census forms, has been offered by many forms processing (Batch Transaction Capture) companies since the late 1980s. Mostly this is based on a bitonal image and pixel count with minimum and maximum pixel counts to eliminate extraneous marks such as those erased with a dirty eraser that when converted into a black and white image (bitonal) can look like a legitimate mark. So this method can cause problems when a user changes his mind, and so some products started to use grayscale to better identify the intent of the marker-internally scantron and NCS scanners used grayscale. One of the first OMR software packages that used bitonal images from image scanners was Remark Office OMR, made by Gravic, Inc. (originally named Principia Products, Inc.). Remark Office OMR 1.0 was released in 1991.

1.3 History

1.3.1 IBM 805 Test Scoring Machine

The concept of evaluating the exams/tests using machines dates back to early 1930s. A high school science teacher in Michigan, named R. B. Johnson, devised a machine for his own use to record student test answers and to compare them to an answer key set up on his machine. International Business Machines Corporation (IBM) learned of Johnson's device and hired him as an engineer to work in their Endicott, New York laboratory. IBM bought the rights to his invention and the machine was soon in market by the name of "IBM 805 Test Scoring Machine".

Tests to be scored by the machine were answered by marking spaces on separate answer sheets which had a capacity of 750 response positions, or 150 five-choice or 300 true-false questions. The answer sheets were dropped into the 805 for processing as shown in Figure 1.1.

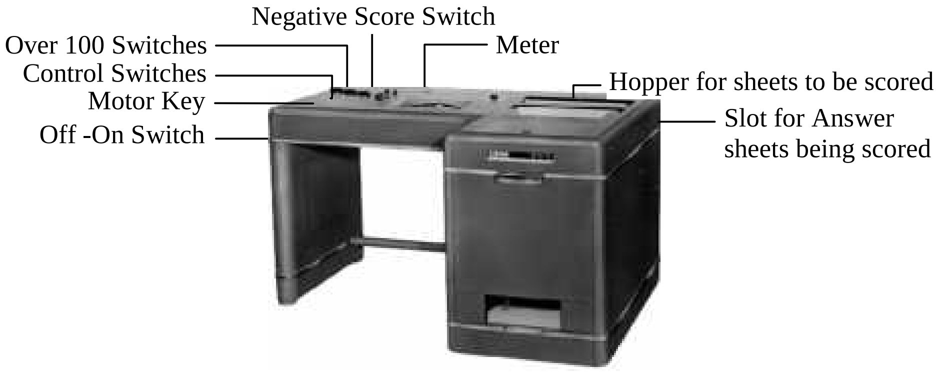


Figure 1.1 IBM 805 Test Scoring Machine

Inside the 805 was a contact plate with 750 contacts (electric fingers) corresponding to the 750 answer positions. Each contact allowed one unit of current to flow when a pencil mark was pressed against it. A scoring key separated the contacts into two groups, the "rights" and the "wrongs." When an answer sheet was inserted, an amount of current equal to the total rights and total wrongs was allowed to flow. When the operator manipulated the controls, the 805 indicated the scores. The IBM 805's speed was limited only by the operator's ability to insert sheets in the machine and record the scores. An experienced operator could record scores on answer sheets at the rate of about 800 sheets per hour. The IBM 805 was withdrawn from marketing in January 1963.

1.3.2 Optical Mark recognition

In 1930's Richard Warren, who was then working at IBM, experimented to replace conductivity method of IBM 805 by optical mark sense system. But the first successful OMR machine was developed by Everett Franklin Lindquist. Lindquist's first optical mark recognition scanner used a mimeograph paper-transport mechanism directly coupled to a magnetic drum memory. Although it was not a general purpose computer, it made extensive use of computer technology. During the same period, IBM also developed a successful optical mark-sense test scoring machine, which was commercialized as the IBM 1230 Optical mark scoring reader. This and a variety of related machines allowed IBM to migrate a wide variety of applications developed for its mark sense machines to the new optical technology. These applications included a variety of inventory management and trouble reporting forms, most of which had the dimensions of a standard punched card.OMR has been used in many situations as mentioned below. The use of OMR in inventory systems was a transition between punch cards and bar codes and is not used as much for this purpose [1]. OMR is still used extensively for surveys and testing though.

1.4 Mechanism of OMR

A typical OMR machine consists of three main units as shown in Figure 1.2.

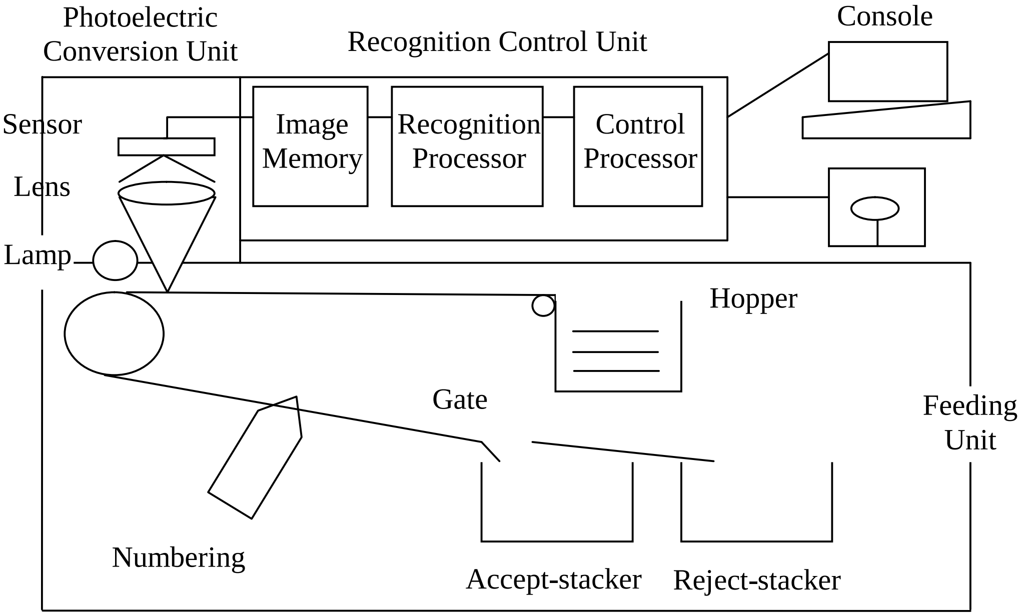


Figure 1.2 Three Units of OMR system

Feeding Unit – The use of feeding unit is to pick up sheets one by one that are piled in the hopper and lets the sheet go through photoelectric conversion unit at a fixed speed and regular interval. Then it carries the sheets to the accept stacker if the sheet has been read properly without any discrepancies or it is carried to reject stacker otherwise.

Photoelectric Conversion Unit- The photoelectric conversion unit irradiates light to the surface of the sheet by some light source like lamp, and then converts the intensity of reflection of light to an electric signal by lens and sensor and inputs the signal to the image memory. The electric signal is recognized as ‘0’ for the bright white light and ‘1’ for the dark light according to the strength of the reflected light. There are two processors in this unit: recognition processor and control processor. The recognition processor reads the mark from the image, recognizes it and send the corresponding signal to control processor. The control processor produces data, and at the same time controls all units of OMR system

Recognition Control Unit - Mark recognition is a kind of pattern recognition technique. This technique has repeatedly been improved and has steadily brought about good results. The recognition process was based on a hardwired logic in the early days. The process is now conducted by software with a recognition-specialized processor or a microcomputer. Recognition by software has brought about more flexibility in the recognition process, increase in the reading methods, advancement in accuracy of reading, and simultaneous input of different types of sheets. The two recognition modes namely Alternative Mode and Bit Mode are employed during recognition process.

Alternative Mode is used when only answer is expected for a question. In this mode, the OMR recognizes the only one entered mark in the block of mark positions on the sheet, converts the mark to a code, and produces it. In case, two marks are found in one block, depth comparison among the marks is conducted, and the deepest mark is chosen. If no difference in depth is detected among the marks, a read-error is to be reported.

Bit mode is used when there are plural answers for one question. All of the information in the block of mark positions on the sheet are recorded and coded as a series of bits.

1.5 Applications

Apart from the OMR being used in examination evaluation, it has many other applications.

- Automated Attendance marking
- Lotteries
- Voting
- Product evaluation
- Community Surveys
- Consumer Surveys
- Geo-coding
- Time sheets/Inventory Counts

1.6 Other forms of OMR

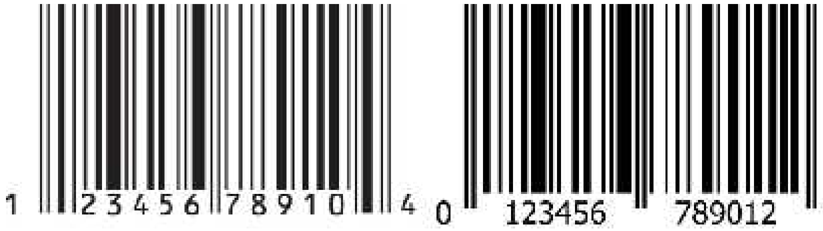
1.6.1 Optical Character Recognition

The technique of converting handwritten or printed text into computer usable form is termed as optical character recognition or OCR. The beginnings of OCR can be traced back to 1809 when the first patents for devices to aid the blind were awarded. In 1912 Emmanuel Goldberg patented a machine that read characters, converted them into standard telegraph code, and then transmitted telegraphic messages over wires without human intervention. In 1914 Fournier D'Albe invented an OCR device called the opto-phone that produced sounds. Each sound corresponded to a specific letter or character. After learning the character equivalent for various sounds, visually impaired people were able to "read" the printed material. Developments in OCR continued throughout the 1930s, becoming more important with the beginnings of the computer industry in the 1940s. OCR development in the 1950s attempted to address the needs of the business world [2].

Recognition of cursive text is an active area of research, with recognition rates even lower than that of hand-printed text. Higher rates of recognition of general cursive script will likely not be possible without the use of contextual or grammatical information. For example, recognizing entire words from a dictionary is easier than trying to parse individual characters from script. Reading the Amount line of a cheque (which is always a written-out number) is an example where using a smaller dictionary can increase recognition rates greatly. Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy. The shapes of individual cursive characters themselves simply do not contain enough information to accurately (greater than 98%) recognize all handwritten cursive script.

1.6.2 Bar Codes

Bar Codes are the zebra-striped marks of various widths and spacing which is optically readable by machines. Two examples of linear barcode of different symbology are shown in Figure 1.3. These can be read by either barcode readers or scanned from an image by special software.



(a) (b)

Figure 1.3 (a) Universal Product Code (UPC) - A barcode symbol. (b) European Article Number (EAN) - 13 barcode symbol

The first use of barcodes was to label railroad cars, but it gained popularity only after it was implemented to automate supermarket checkout systems. The simplicity, universality and low cost of barcodes has widened the application area of barcodes. Since 20th century, barcodes - especially UPC - have slowly become an essential part of modern civilization. Some common applications include-

- The tracking of item movement, including rental cars, airline luggage, nuclear waste, mail and parcels.
- Used on automobiles and its spare parts.
- Document Management – Using barcode it is easier to separate and index the documents that have been imaged in batch scanning applications.
- Many tickets now have barcodes that need to be validated before allowing the holder to enter sports arenas, cinemas, theatres, fairgrounds, transportation etc
- Recently, researchers have placed tiny barcodes on individual bees to track the insects' mating habits.

- Since 2005, airlines use an International Air Transport Association (IATA) - standard bar code on boarding passes (BCBP), and since 2008 bar codes sent to mobile phones enable electronic boarding passes.

1.6.3 Matrix Codes

The data matrix code is a 2-D barcode as shown in Figure 2.4 which contains information in black and white cells arranged either in a square or rectangle pattern. A matrix code can store as much as 2335 alphanumeric characters. The size of data encoded can ranges from few bytes to 2 kilobytes, but the length of encoded data depends on the dimension of symbol used.

Data Matrix symbols are square in shape and they are made of cells: each cell representing a bit. According to the notations used, a “0” is a “light” module and “1” is a dark module or vice versa. Every Data Matrix is composed of two solid adjacent borders in an “L” shape which is also called finder pattern. And the two other borders consists of alternating dark and light cells which are called timing pattern. The information is encoded in rows and columns of cells enclosed in these borders. The finder pattern is used to locate and orient the symbol while the timing pattern provides a count of the number of rows and columns in the symbol. Symbol sizes vary from 8×8 to 144×144 depending on the amount of information to be stored.

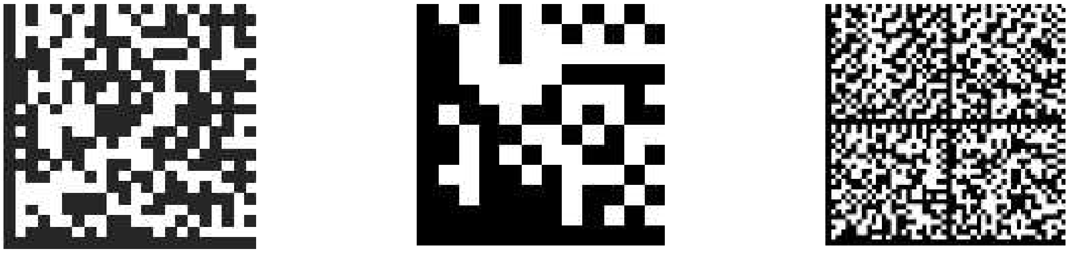


Figure 1.4 Examples of data matrix code

There are various other versions of matrix codes which are used some specific domains for example Aztec code is used in land transport and airlines, MaxiCode for tracking and managing the shipment of packages, Quick Response code (QR code) is used in Japan for mobile tagging, SemaCode stores URLs. The high capacity color barcodes is a new system where colored triangles are used instead of black and white lines or cells. This has enhanced the amount of information that can be stored by the symbols.

1.7 Objective

The process of examination evaluation requires a very high degree of accuracy which may not be possible during manual evaluation as a human being tends to get fatigued due to monotonous nature of the job. Main objectives of this research are:

- To extract date from dynamic sheet of each quality (low to high) taken from different intensity, different position, different degree of rotation with maximum degree of accuracy.
- To develop the efficient system which extracting data and publish the result for objective examination.

1.8 Literature Review

Chinnasarn et al, in his paper ***“An image-processing oriented optical mark reader”*** presented a system which was based on Personal Computer-type microcontroller and image scanner. The system operations can be distinguished in two modes: learning mode and operation mode. In the learning mode, the model corresponding to each type of answer sheet is constructed by extracting all significant horizontal and vertical lines in the blank-sheet image. Then, every possibly cross-line will be located to form rectangular area. In the operation mode, each sheet fed into the system has to be identified by matching the horizontal lines detected with every model. The data extraction from each area can be performed based on the horizontal and vertical projections of the histogram. For the answer checking purpose, the number of black pixels in each answer block is counted, and the difference of those numbers between the input and its corresponding model is used as decision criterion. Finally, the database containing a list of subjects, students, and scores can be created. This is the first image-based OMR technique [3].

Hussmann S. etal, presented his paper ***“Low Cost and high speed Optical mark reader based on Intelligent line Camera”*** in 2003. This paper describes the design and implementation of an OMR prototype system for marking multiple-choice tests automatically. Parameter testing is carried out before the platform and the multiple-choice answer sheet has been designed. Position recognition and position verification methods have been developed and implemented in an intelligent line scan camera. The position recognition process is implemented into a Field Programmable Gate Array (FPGA), whereas the verification process is implemented into a micro-controller. The verified results are then sent to the Graphical User Interface (GUI) for answers checking and statistical analysis [4]. The developed prototype system could read 720 forms/hour and the overall system cost was much lower than 14 commercially available OMR products. However, the resolution and overall system design was not satisfying and lead to further investigation [5].

Hussmann S. etal, presented his paper "***A high speed optical mark reader hardware implementation at low cost using programmable logic"*** in 2005. It has described the development of a low-cost and high speed OMR system prototype for marking multiple choice questions. The novelty of this approach is the implementation of the complete system into a single low-cost Field Programmable Gate Array (FPGA) to achieve the high processing speed. Effective mark detection and verification algorithms have been developed and implemented to achieve real-time performance at low computational cost. The OMR is capable of processing a high-resolution CCD linear sensor with 3456 pixels at 5000 frame/s at the effective maximum clock rate of the sensor of 20 MHz (4×5 MHz). The performance of the prototype system is tested for different marker colours and marking methods [5].

Deng H. etal, presented his paper ***"A low-cost OMR solution for educational applications"*** in 2008. It aims at eliminating the drawbacks of current OMR technique and designing the low-cost OMR (termed as LCOMR). The new technique is capable of processing thin papers and low printing precision answer sheets. The system key techniques and relevant implementations, which include the image scan, tilt correction, scanning error correction, regional deformation correction and mark recognition, are presented. This new technique is proved robust and effective by the processing results of large amount of questionnaires. More than 15 schools in china have adopted this new technique. The school teachers are capable of designing questionnaires by themselves and the supervisor of schools can adopt this technique to investigate the effect of learning and teaching easily and quickly. LCOMR partially resolves the drawbacks of traditional OMR techniques and improves it usability [6].

1.9 Problem Definition

Optical Mark Reader: OMR is an automatic machine for multiple-choice checking system. It is very popular in used. Not only high accuracy but also it gives high speed for check. But there will some inflexibility of OMR.

First, in each time, it can check only single answer sheet form and single subject. The OMR will check between an input answer sheet and the template answer sheet. It will reject an input sheet if differ from the template. In other word, a template form contained only one subject answer model. So if we input answer sheet with different subject from template, it will give a wrong score, some OMR machine will not check subject. Second, OMR can check only a full-fill mark in static grid position, circle or square, rectangle filled by pen or pencil. Third, for low quality images and internally skewed images it will difficult to extract data.

1.10 Proposed Solution

The image of sheet taken from webcam or scanner is converted into binary image using adaptive binarization method; skew correction is done by detecting line using hough transform. Binary image is cut into Roll number region and Answer region using reference square fixed on answer sheet. The system operations can be distinguished into two models: learning mode and operation mode. In the learning mode, system separate each static block in a sheet using horizontal and vertical projection after doing morphological operation. Each block is divided into ten equal parts and mark area for each question and mark cells for each option of each question. Operational mode is used to check and decide the answer. For the answer checking purpose, the number of black pixels in each answer block is counted, and decision is created from comparison of number of black pixel between different cells of each question. Similar process is used to extract Roll number. Score is calculated by comparing correct answer and marked answer. Finally, the list of Roll number and scores can be exported into data sheet.

1.11 Outline of the Thesis

The remaining part of the document is organized as follows,

Chapter 2 describes the system overview along with visual stapes.

Chapter 3 describes research methodologies used in the research. All the preprocessing algorithms are described in this part of the document.

Chapter 4 describes the implementation of the system. It includes the development of algorithms given in chapter 3.

Chapter 5 describes the experimentation results of the systems. It includes the Comparison of algorithms, Performance and efficiency of the systems.

Chapter 6 contains the summary, future scope and further studies of the research work.

2 SYSTEM OVERVIEW

2.1 System Overview

The overall organization of the system is shown in figure. It contains 2 models, one for learning mode. It prepares for, how to computer understand information in each answer form. This mode compound with a form modeling for understands form type. The other is operation mode that deals with how to retrieve score from the input sheet.

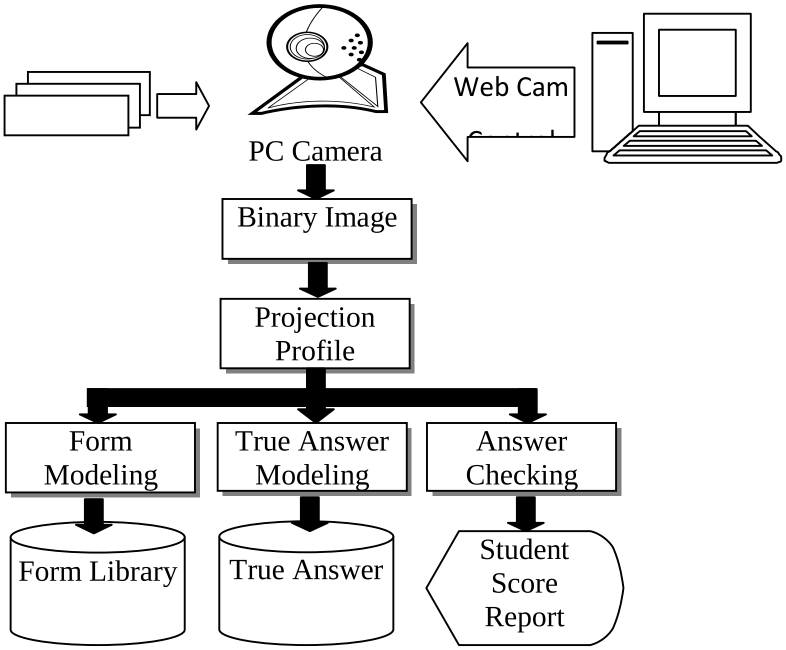


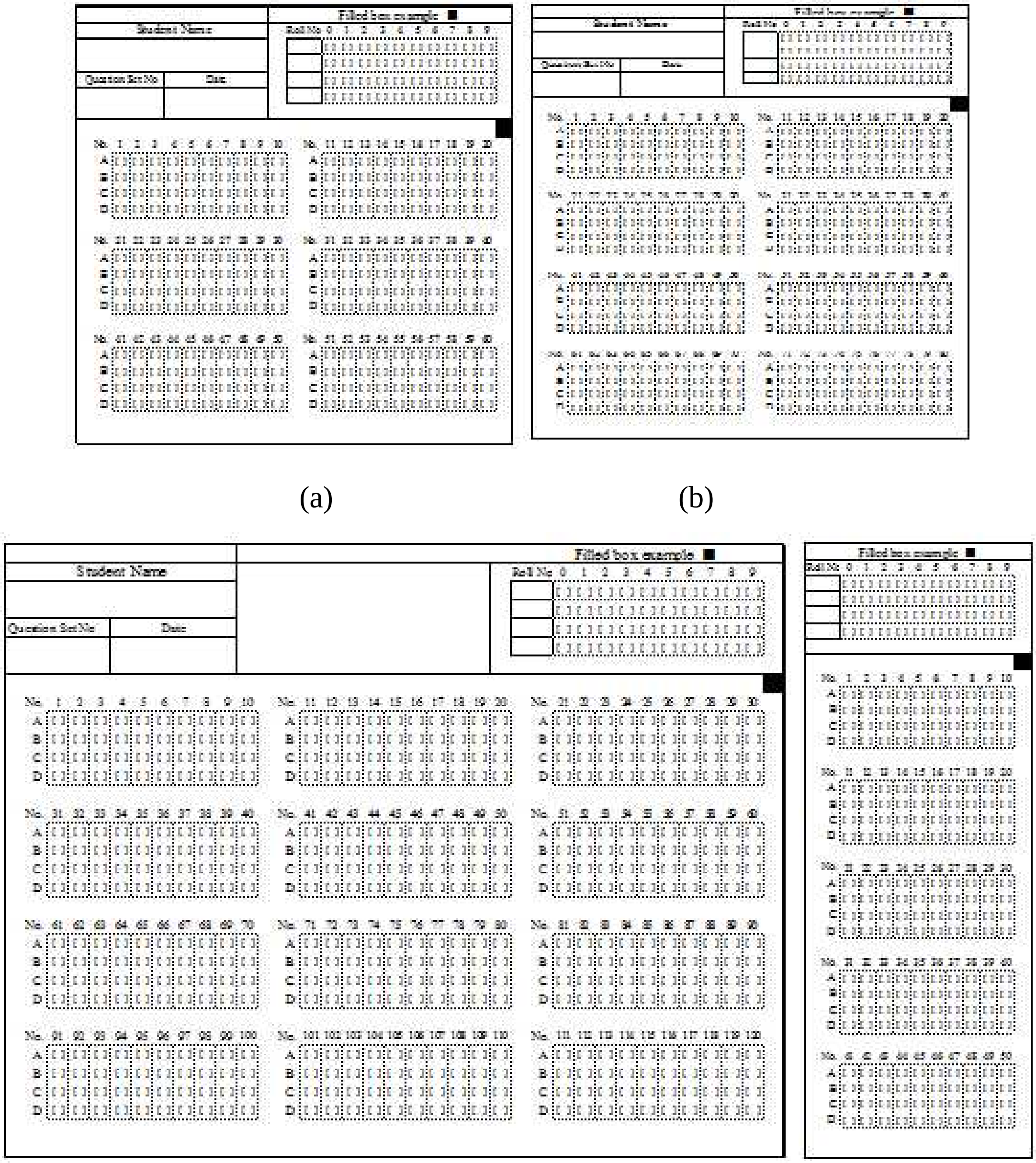
Figure 2.1 System Overview

2.2 Prototypes of Answer Sheet

For this system, I have design answer sheet by taking reference of answer sheet used by PCOMR software which is developed by SiQiSoft in 2009. PC OMR V9.0 is leading Windows based forms-processing shareware software for tests and surveys. It is an intelligent testing tool to collect data from optical marks on plain paper forms. Figure 2.2 shows the prototype of answer sheet designed for this system. There are different answer sheet with different number of question. Figure 2.2 (a), (b), (c) and (d) shows answer sheet with 60, 80, 120 and 50 number of question respectively. Each block of ten questions is static in this system. By combining this block in vertical of horizontally we can make dynamic answer sheet with different number of question.

2.3 Learning Mode

An answer sheet form is entered to the system as a digital image by optical scanner or web cam. Again, the image was preprocessed with noise reduction and skew detection. An algorithm will detect number of standard blocks of question and number of questions in sheet. In this mode I scope to model only a necessary rectangular box, Student Roll No, Answer area fields and number of questions in a sheet.



(a)

(b)

(c)

(d)

Figure 2.2 Prototype of Answer sheets, (a) with 60 Questions (b) with 100 Questions (c) with 120 Questions and (d) with 50 Questions.

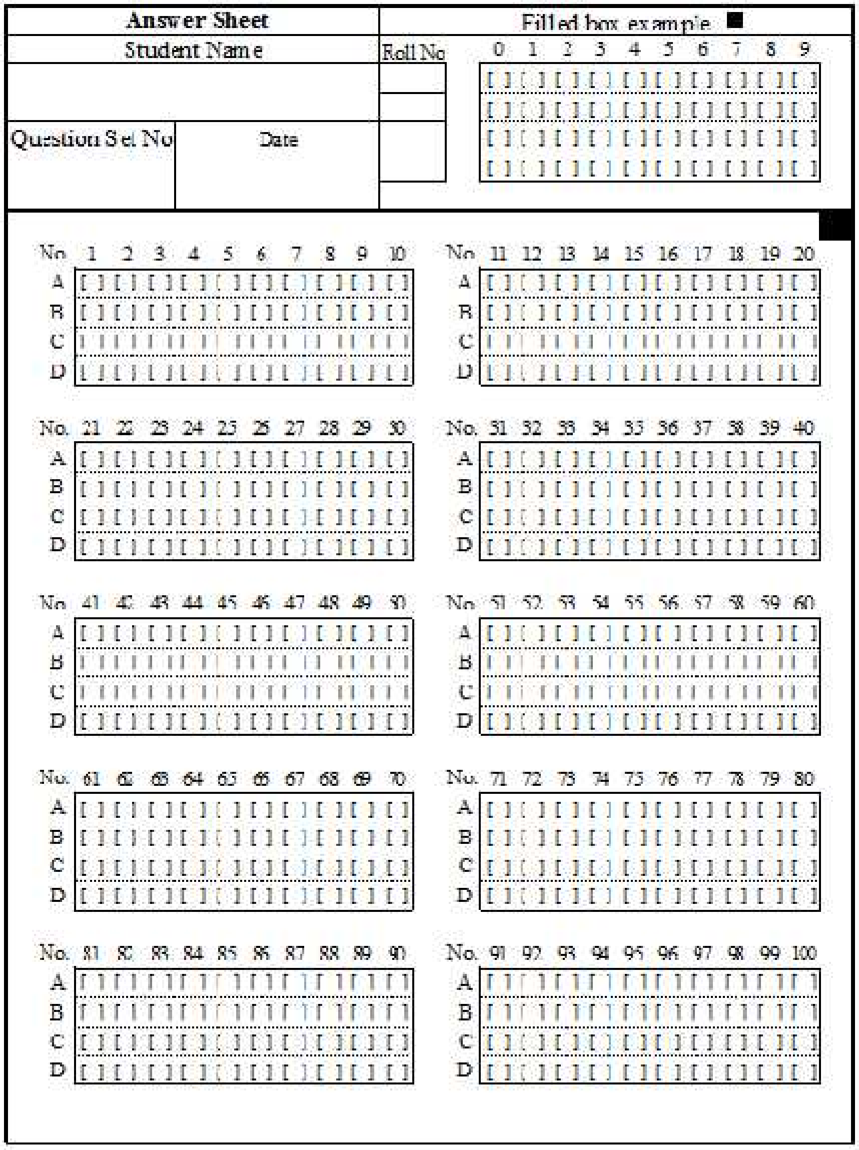

Answer Sheet
Filled box example ■
Student Name
Roll No
0 1 2 3 4 5 6 7 8 9
Question Set No
Date
No 1 2 3 4 5 6 7 8 9 10
No 11 12 13 14 15 16 17 18 19 20
No. 21 22 23 24 25 26 27 28 29 30
No. 31 32 33 34 35 36 37 38 39 40
No 41 42 43 44 45 46 47 48 49 50
No 51 52 53 54 55 56 57 58 59 60
No. 61 62 63 64 65 66 67 68 69 70
No. 71 72 73 74 75 76 77 78 79 80
No. 81 82 83 84 85 86 87 88 89 90
No. 91 92 93 94 95 96 97 98 99 100
A
B
C
D

Figure 2.3 Prototype of standard answer sheet

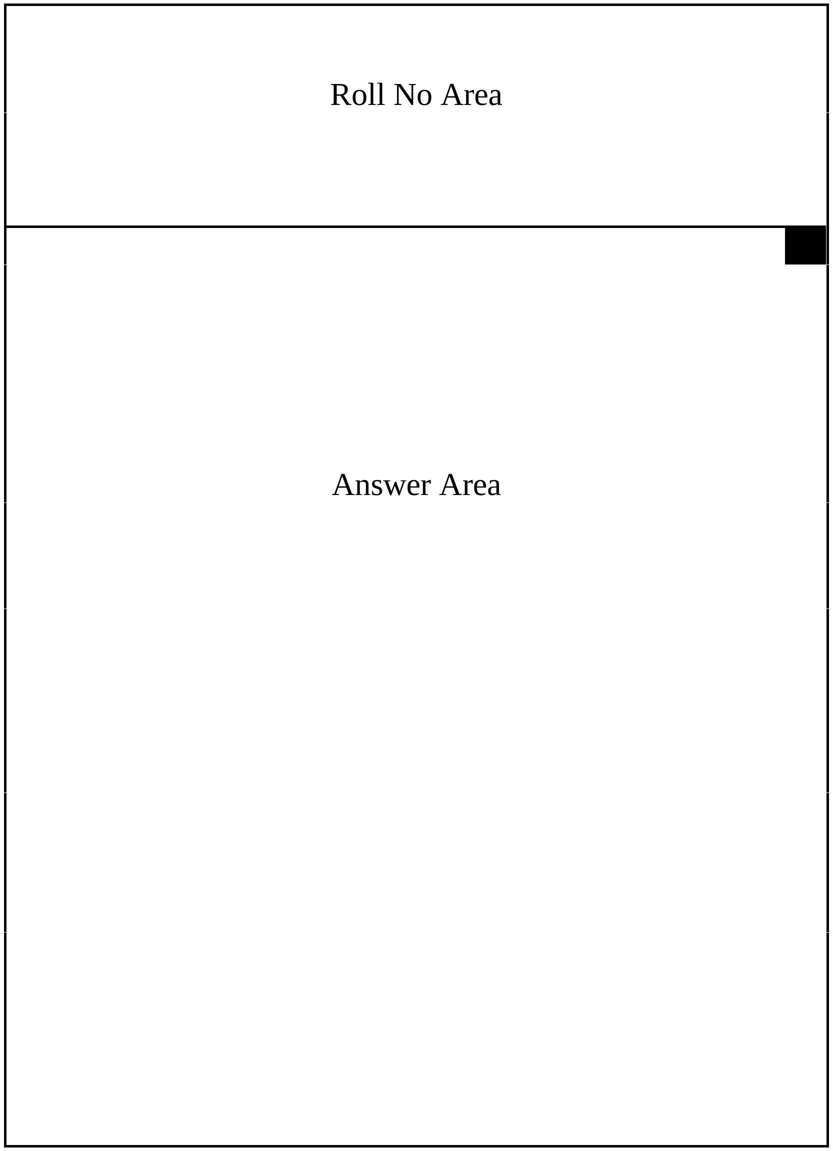


Figure 2.4 Separation of Roll No and Answer area

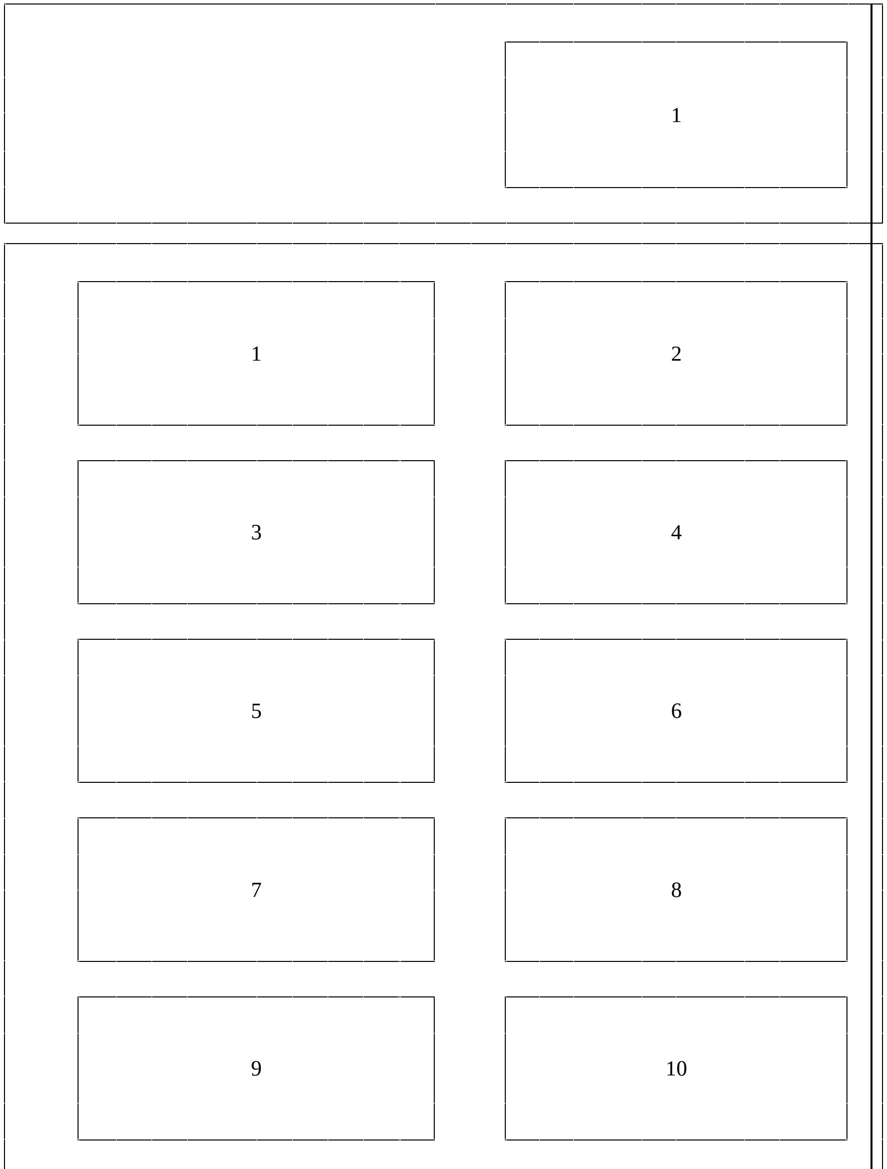


Figure 2.5 Sub Answer Area

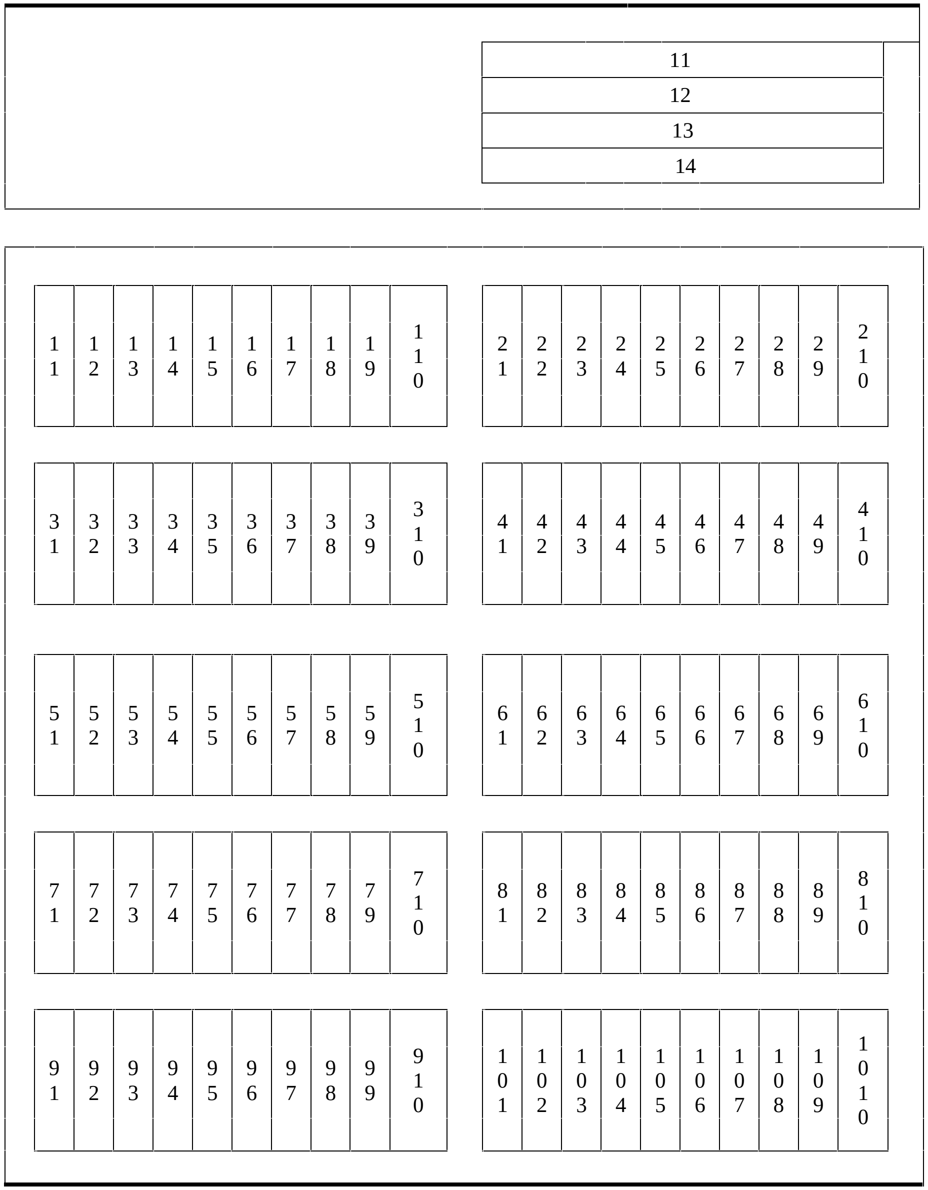
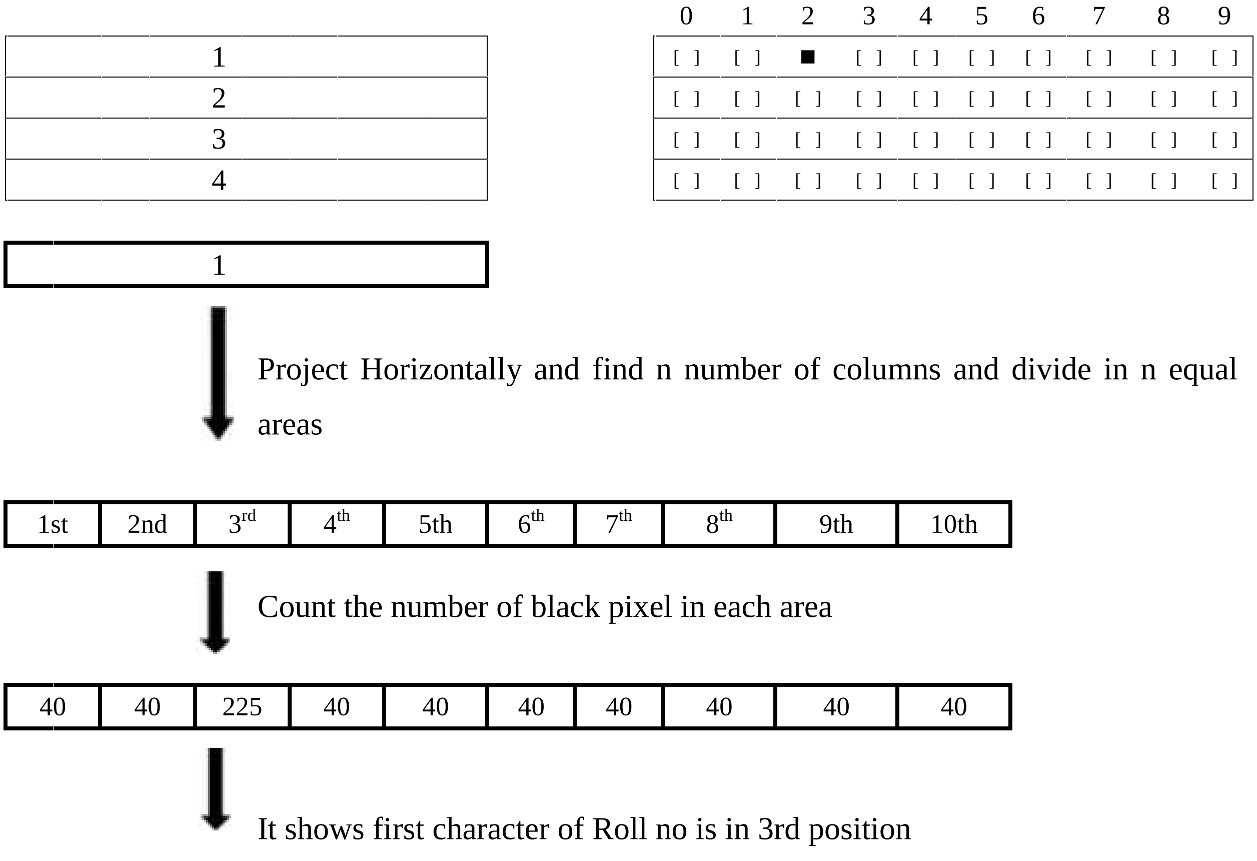


Figure 2.6 Answer Area of Each Question

Visual steps of learning model are shown in figure 2.2 to 2.5. Figure 2.2 shows the prototype of standard answer sheet that is designed for the system. Figure 2.3 show separation of answer area and answer area using reference solid square. Figure 2.4 show the standard area for ten questions and Roll Number. Figure 2.5 shows extraction of answer area for each question and roll number area for each unit number.

2.4 Operation mode

In this mode, the type of each answer sheet has to be identified after preprocessing. The number of black pixels in each boundary is counted. For student Roll number applied a vertical projection profile and means threshold for segments a boundary in each field. Each student Roll No is created from the square position with the maximum number of black pixels.



1
2
3
4

0	1	2	3	4	5	6	7	8	9
[ ]	[ ]	■	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

1

Project Horizontally and find n number of columns and divide in n equal areas

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th

Count the number of black pixel in each area

40	40	225	40	40	40	40	40	40	40

It shows first character of Roll no is in 3rd position

Figure 2. 7 Roll No Extraction Process

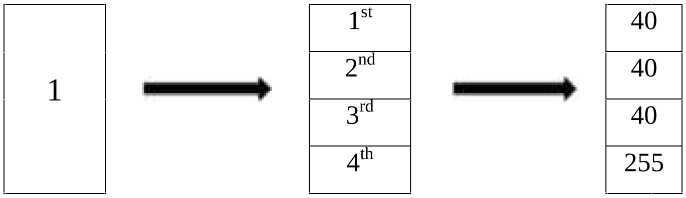
In our designed answer sheet 2 is in 3rd position it shows the first number of Roll No is 2 Similarly we can obtain others position numbers. Figure 2.6 shows the Roll No extraction process.

For the answer check process the number of black pixels in each circle boundary is counted and searches for the maximum value position, for each answer item, the maximum value position is compared with true answer item model, in list. If both are the same position given point else skip to next answer item until end list. But some students sometime marked or fill in two or more positions in each answer item. To solve those problems, it will show both answer not only the maximum one.

Similarly for answer of each question

1	2	3	4	5	6	7	8	9	10

[ ]	[]	■	[ ]	[ ]	[ ]	[ ]	■	[ ]	[ ]
[ ]	[ ]	[]	[ ]	[ ]	■	[ ]	[ ]	[ ]	[ ]
[ ]	■	[]	■	[ ]	[ ]	■	[ ]	■	[ ]
■	[ ]	[]	[ ]	■	[ ]	[ ]	[ ]	[ ]	■



It shows correct answer for first question of current rectangular box is D.

(a)

Output after all checking Process

1st Rectangular Field (1-10)

1	2	3	4	5	6	7	8	9	10
D	C	A	C	D	B	C	A	C	D

2nd Rectangular Field (11-20)

1	2	3	4	5	6	7	8	9	10

(b)

Figure 2. 8 Answer Extraction Process

Figure 2.7 (a) shows the answer extraction process for the answer of single question; Figure 2.7 (b) shows similarly detected mark answer for answer of ten questions.

3 RESEARCH METHODOLOGY

3.1 Preprocessing

Preprocessing steps are described in Algorithm 3.1. All steps are described in section 3.1.

Algorithm 3.1 Image Preprocessing

1. RGB to Gray Conversion
2. Edge Detection and Skew Correction using Hough transform
3. Adaptive Binary Conversion to detect text and background images.
4. Reference Point Detection and Rotation to Correct top and bottom of sheet.
5. Crop out Roll Number Field and Answer Field
6. Change border line to horizontal or vertical by Horizontal and Vertical stretching to correct internal skew of sheet.
7. Use vertical and horizontal projection profile to detect number sub answer area.
8. Similar Process is used to detect Answer field area for each question and Roll No area.
9. Count black pixel for each answer area and detect marked answer.
10. Store marked answer to database and compare with answer to get result.

3.1.1 RGB to Gray Conversion

Within the three-dimensional array of an RGB image, two of the dimensions specify the location of a pixel within an image. The other dimension specifies the color of each pixel The color dimension always has a size of 3 and is composed of the red, green, and blue color bands (channels) of the image. Grayscale images represent intensities. Pixels range from least intense (black) to most intense (white). Pixel values usually range from 0 to 255 or are scaled to this range when displayed. Gray Level image contains 30%Red, 59% Green and 11% Blue in each pixel. RGB image can be converted to Grayscale using following equation.

$$g(x,y)\,\mathrm{X}0.2989 * f_R(x,y)\,\Gamma 0.5870 * f_G(x,y)\,\Gamma 0.114 * f_B(x,y) \qquad (3.1)$$

3.1.2 Edge Detection and Skew Correction

The main idea to detect skew for a regular image and a text image is the same. We have to first turn the pictures to Gray Scale. Then, for regular (non text) images, we have to find the edges or feature lines of the image. There are many algorithms to detect the edges of an image, and in my case, I used the canny edge detection algorithm. This is a multi-step edge detection procedure by Canny. The purpose of the following two methods is to detect edges with noise suppressed at the same time. Algorithm for canny edge detection is given in algorithm 3.2.

Hough transform is used to detect skew angle, remove unwanted area which lies in out of bounded lines. Hough transforms each feature point maps to a sinusoid in the parameter space. Equation of straight line in polar form is given by equation 3.2.

$$\text{°}\ \mathrm{X} x\cos{}^{-}\ \Gamma y\sin{}^{-} \tag{3.2}$$

Where, ° is perpendicular distance from origin to the given line and θ is angle made by x axis in positive direction with perpendicular line. For each detected edge point(x,y), Find value of ° by varying θ from 1 to π . An accumulator A[° ,θ] is increased by 1 for corresponding value of ° and θ. Pick values of A[° ,θ] gives ° and θ for strongest line segments. The algorithm for the Hough transform can be expressed as in algorithm 3.3.

Algorithm 3.2 Canny edge detection

1. Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures.

$$g(m,n)\,\mathrm{X}G_{\pm}(m,n) * f(m,n) \tag{3.3}$$

Where,

$$G_{\pm}\ \mathrm{X}\frac{1}{\sqrt{2\text{®}\pm^{2}}}\exp\ \frac{m^{2}\,\Gamma n^{2}}{2\pm^{2}} \tag{3.4}$$

2. Compute gradient of using any of the gradient operators (Roberts, Sobel, Prewitt, etc) to get:

$$M(m,n)\,\mathrm{X}\sqrt{g_{m}^{2}(m,n)\,\Gamma\, g_{n}^{2}(m,n)} \tag{3.5}$$

And

$${}^{-}(m,n)\,\mathrm{X}\tan^{Z1}\ \frac{g_{n}(m,n)}{g_{m}(m,n)} \tag{3.6}$$

3. Threshold M:

$$M_{T}(m,n)\,\mathrm{X}\ \begin{matrix} M(m,n) & \mathit{if} & M(m,n)\,\Psi T \\ 0 & & \mathit{otherwise} \end{matrix} \tag{3.7}$$

Where, T is so chosen that all edge elements are kept while most of the noise is suppressed.

4. Suppress non-maxima pixels in the edges in M_T obtained above to thin the edge ridges (as the edges might have been broadened in step 1). To do so, check to see whether each non-zero is $M_T(m,n)$ greater than its two neighbors along the gradient direction$^{-}(m,n)$. If so, keep $M_T(m,n)$ unchanged, otherwise, set it to 0.
5. Threshold the previous result by two different thresholds T_1 and T_2 (where $T_1 \Phi T_2$) to obtain two binary images T_1 and T_2. Compared to T_1 and T_2 has less noise and fewer false edges but larger gaps between edge segments.
6. Link edges segments in T_2 to form continuous edges. To do so, trace each segment in T_2 to its end and then search its neighbors in T_1 to find any edge segment in T_1 to bridge the gap until reaching another edge segment in T_2.

Algorithm 3.3 Generalized hough transform

```
Find all of the desired feature points in the image
For each feature point:
    For each pixel i on the target's boundary
        Get the relative position of the reference point from i
        Add this offset to the position of i
        Increment that position in the accumulator
Find maxima in the accumulator
If desired, map each maximum in the accumulator back to image space
```

3.1.3 Thresholding

Most document analysis algorithms are built on taking advantage of the underlying binarized image data [7]. The use of a bi-level information decreases the computational load and enables the utilization of the simplified analysis methods compared to 256 levels of grey-scale or colour image information. Document image understanding methods require logical and semantic content preservation during thresholding. For example, letter connectivity must be maintained for optical character recognition and textual compression [8]. This requirement narrows down the use of a global threshold in many cases.

Binarization has been a subject of intense research interest during the last ten years. Most of the developed algorithms rely on statistical methods, not considering the special nature of document images. However, recent developments on document types, for example documents with mixed text and graphics, call for more specialized binarization techniques.

In current techniques, the binarization (threshold selection) is usually performed either globally or locally. Some hybrid methods have also been proposed. The global methods use one calculated threshold value to divide image pixels into object or background classes, whereas the local schemes can use many different adapted values selected according to the local area information. Hybrid methods use both global and local information to decide the pixel label.

The most conventional approach is a global threshold, where one threshold value (single threshold) is selected for the entire image according to global/local information. In local thresholding the threshold values are determined locally, e.g. pixel by pixel, or region by region. Then, a specified region can have 'single threshold' that is changed from region to region according to threshold candidate selection for a given area. Multi thresholding is a scheme, where image semantics are evaluated. Then, each pixel can have more than one threshold value depending on the connectivity or other semantic dependency related to physical, logical or pictorial contents.

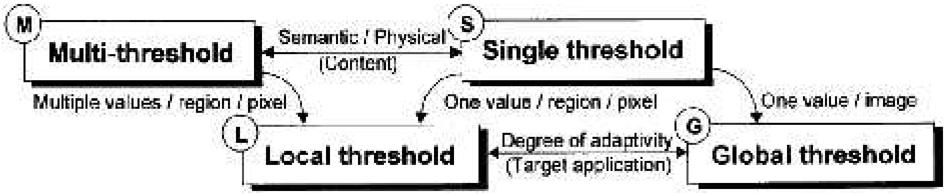
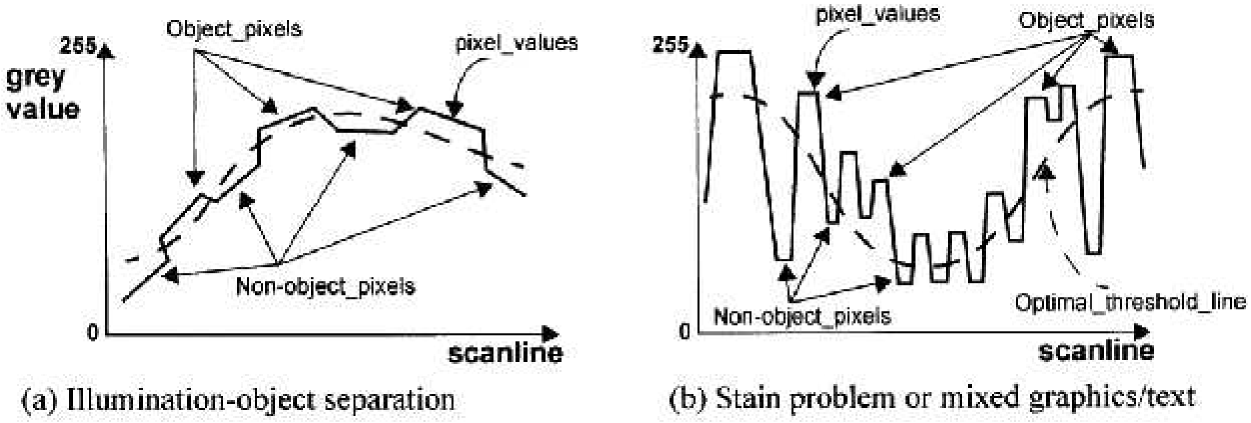


Figure 3.1 Taxonomy of thresholding schemes [9]

Fig. 3.3 depicts two types of typical degradation, when dealing with scanned grey-scale document images. In Fig. 3.3a the threshold 'base line' is changing due to illumination effect or implanted (designed) entity. Then, each object has a different base level that affects the object/ non-object separation decision in selecting threshold(s). In Fig. 3.3b a general type 'stain problem' is presented. In this case, the background and object levels are fluctuating from clear separation to non-clear separation and small level difference between object/non-object. The optimal threshold lines are drawn to both images to depict the base line that a successful binarization algorithm should mimic.



(a) Illumination-object separation

(b) Stain problem or mixed graphics/text

Figure 3.2 Examples of document analysis problem types in binarization [9]

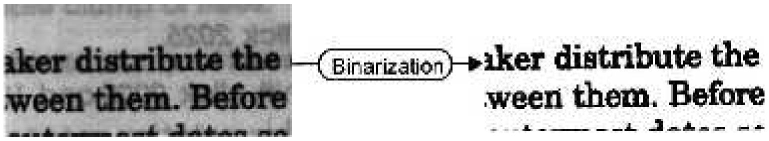


Figure 3.3 Example of good binarization on degraded sample image

Fig. 3 presents another type of problem, frequently occurring in scanned document images: more than two different levels are visible in textual areas due to transparency of the next page. Then, a binarization algorithm should cope with at least two different threshold candidates: background transparent text and background- text. The binarized example presents a correct binarization result.

For document image binarization, we propose a new method that first performs a rapid classification of the local contents of a page to background, pictures and text. Two different approaches are then applied to define a threshold for each pixel: a soft decision method (SDM) for background and pictures, and a specialized text binarization method (TBM) for textual and line drawing areas. The SDM includes noise filtering and signal tracking capabilities, while the TBM is used to separate text components from background in bad conditions, caused by uneven (il) lumination or noise. Finally, the outcome of these algorithms are combined.

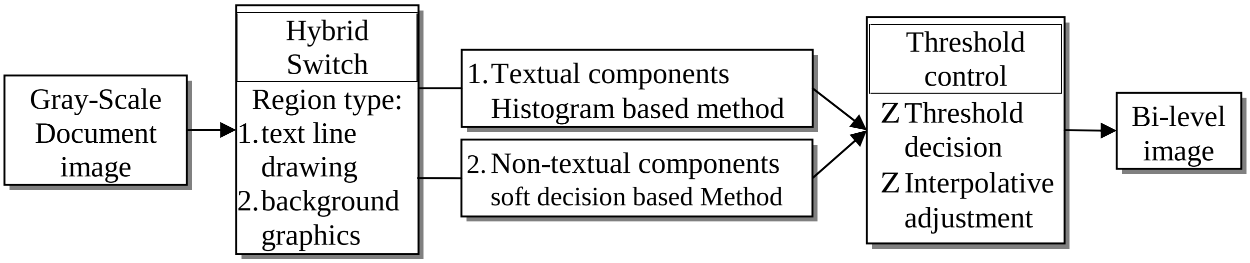

Gray-Scale Document image
Hybrid Switch
Region type:
1. text line drawing
2. background graphics
1. Textual components Histogram based method
2. Non-textual components soft decision based Method
Threshold control
Z Threshold decision
Z Interpolative adjustment
Bi-level image

Figure 3.4 Overview of the binarization algorithm

Binarization technique is aimed to be used as a first stage in various document analysis, processing and retrieval tasks. Therefore, the special document characteristics, like textual properties, graphics, line-drawings and complex mixtures of their layout-semantics should be included in the requirements. On the other hand, the technique should be simple while taking all the document analysis demands into consideration. Fig.4 presents the general approach of the binarization processing flow. Since typical document segmentation and labelling for content analysis is out of question in this phase, we use a rapid hybrid switch that dispatches the small, resolution adapted windows to textual (1) and non-textual (2) threshold evaluation techniques. The switch was developed to cover most generic appearances of typical document layout types and can easily be modified for others as well. The threshold evaluation techniques are adapted to textual and non-textual area properties, with the special tolerance and detection to different basic defect types that are usually introduced to images. The outcome of these techniques represent a threshold value proposed for each pixel, or every nth pixel, decided by the user. These values are used to collect the final outcome of the binarization by a threshold control module. The technique also enables the utilization of multi -thresholds region by region of globally, if desired.

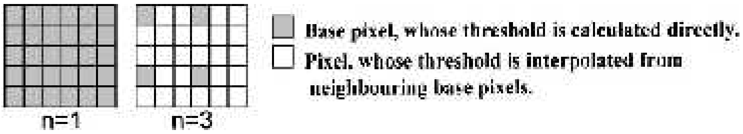

Base pixel, whose threshold is calculated directly.
Pixel, whose threshold is interpolated from neighbouring base pixels.
n=1
n=3

Figure 3.5 Interpolation options for binarization computation

3.1.4 Adaptive Binarization

The problem of document binarization is as old as the document image analysis itself. A large number of binarization techniques have been proposed over the last two decades. These techniques can generally be classified into two categories i.e. global thresholding and local thresholding. Global thresholding methods employ a single intensity threshold value. This value is calculated based on some heuristics or statistics of some global image attributes to classify image pixels into foreground (text) or background (non-text) pixels [10]. The main issue with global methods is that they cannot adapt well to uneven illumination and noise, hence do not perform well on low quality document images. Local thresholding methods, on the other hand, compute a threshold for each pixel in the image on the basis of the content in its neighborhood. As opposed to global thresholding, local methods generally perform better for low quality images.

Algorithm 3.4 Niblack Algorithm

Niblack's algorithm [11] calculates a pixel-wise threshold by sliding a rectangular window over the gray level image. The computation of threshold is based on the local mean m and the standard deviation s of all the pixels in the window and is given by the equation 5.8:

$$T(x,y)\,\mathrm{X}\,m(x,y)\,\Gamma\,k.x(s,y)$$

$$T_{Niblack}\,\mathrm{X}\,m\,\Gamma\,k\sqrt{\frac{1}{NP}\quad (p_i\,\mathrm{Z}\,m)^2}\,\mathrm{X}\,m\,\Gamma\,k\sqrt{\frac{p_i^{\;2}}{NP}\,\mathrm{Z}\,m^2}\,\mathrm{X}\,m\,\Gamma\,k\sqrt{B} \qquad (3.8)$$

where *NP* is the number of pixels in the gray image, *m* is the average value of the pixels p_i, and *k* is fixed to -0.2 by the authors. Advantage of Niblack is that it always identifies the text regions correctly as foreground but on the other handtends to produce a large amount of binarization noise in non-text regions also.

Algorithm 3.5 Sauvola's Algorithm

Sauvola's algorithm [12] claims to improve Niblack's method by computing the threshold using the dynamic range of image gray-value standard deviation, R:

$$T_{Sauvola}(x,y)\,\mathrm{X}\,m(x,y)*\;1\,\Gamma\,k*\;\frac{s(x,y)}{R}\,\mathrm{Z}1 \qquad (3.9)$$

Where, k is set to 0.5 and R to 128. This method outperforms Niblack's algorithm in images where the text pixels have near 0 gray-values and the background pixels have near 255 gray-values. However, in images where the gray values of text and non-text pixels are close to each other, the results degrade significantly.

Algorithm 3.6 Wolff’s Algorithm

To address the issues in Sauvola’s algorithm, Wolf et al. [WOF 03] propose to normalize the contrast and the mean gray value of the image and compute the threshold as:

$$T_{Wolf}\ \mathrm{X}(1\mathrm{Z}k)*m\,\Gamma k*M\,\Gamma k*\frac{s}{R}(m\,\mathrm{Z}M) \qquad (3.10)$$

Where, k is fixed to 0.5, M is the minimum gray value of the image and R is set to the maximum gray-value standard deviation obtained over all the local neighborhoods (windows).This method in most cases outperforms its predecessors. However, degradation is observed in its performance if there is a sharp change in background gray values across the image. This is due to the fact that the values of M and R are calculated from the whole image. So even a small noisy patch could significantly influence M and R values, thus eventually calculating misleading binarization thresholds.

Algorithm 3.7 Feng’s Algorithm

Instead of calculating dynamic range of gray-value standard deviation from the whole image like [13], Feng et al. [14] propose calculating it locally introducing the notion of two local windows, one contained within the other. The values of local mean *m*, the minimum gray-level *M*, and standard deviation *s* are calculated in the primary local window while the dynamic range standard deviation R_s is calculated in the larger window termed as ‘secondary local window’. Binarization threshold is then computed as:

$$T_{Feng}\ \mathrm{X}(1\,\mathrm{Z}\sim_1)*m\,\Gamma\sim_2 *\ \frac{s}{R_s}\ *\text{ƒ}m\,\mathrm{Z}M\,\text{Å}\Gamma\sim_3 *M \qquad (3.11)$$

Where,

$$\sim_2\ \mathrm{X}k_1\ \frac{s}{R_s}^{\text{¥}} \text{ and } \sim_3\ \mathrm{X}k_2\ \frac{s}{R_s}^{\text{¥}}\ .$$

Based on the experimental experiences of authors, ¥ is set to 2 while the values of other parameters, $\sim_1, k_1$ and k_2 are proposed to be in the ranges 0.1-0.2, 0.15-0.25 and 0.01-0.05 respectively. This method addresses well the R-problem in the Wolf’s algorithm. However the introduction of three parameters, determined empirically, leaves the robustness of this method questionable.

In this system there is no any background image only text , lines, solid square , circles or ellipse should be detect so, Niblack’s algorithm will better to used with sliding window of 10 x 10 pixel.

3.1.5 Reference Point Detection

In this system a solid square of 15x15px will used to separate Roll No area and Answer area. At first, solid square should be detect and send it to first quadrant by rotating by right angle till it will not rich at first quadrant, since reference point is fixed at first quadrant of designed sheet and sheet is already skewed to horizontal or vertical direction. Morphological close operation is used to detect different shape in image. The morphological close operation is a dilation followed by erosion.

3.1.6 Morphological Operations

Binary mathematical morphology consists of two basic operations, Dilation and Erosion and several composite relations closing and opening.

Dilation expands the connected sets of 1s of binary image. It can be used for expanding shapes and filling holes, gaps and gulfs.

Algorithm 3.8 Dilation

Let

X= Set of Euclidean coordinates corresponding to the input binary image

K= Set of coordinates for the structuring element.

Kx= Translation of K so that its origin is at x.

Dilation of X by K is simply the set of all points x such that the intersection of Kx with X is nonempty.

In other words, for each background (input) pixel, superimpose the structuring element with the input image,

- If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value.
- If all the corresponding pixels in the image are background however, the input pixel is left at the background value.

Erosion shrinks the connected sets of 1s of a binary image. It can be used for shrinking shapes, removing bridges, branches and small protrusions.

Algorithm 3.9 Erosion

Let

X= Set of Euclidean coordinates corresponding to the input binary image

K= Set of coordinates for the structuring element.

Kx= Translation of K so that its origin is at x.

Erosion of X by K is simply the set of all points x such that Kx is a subset of X.

In other words, for each foreground (input) pixel, superimpose the structuring element with the input image,

- The input pixel is left as it is if it is the foreground pixel in the structuring element.
- If any of the corresponding pixels in the image are background however, the input pixel is also set to background value.

Structuring element is a shape mask used in the basic morphological operations. They can be any shape and size that is digitally representable, and each has an origin.

A binary image can be thought of as a set of points on the Euclidean space $£^{2}$. If Set X is binary image and Set S is structuring element, Set operation is defined as,

$$\begin{aligned} &Dilation : X \quad S \sum p \quad £^{2} : p \, \mathrm{X} \, x \, \Gamma \, s, x \quad X \quad and \quad s \quad S* \\ &Erosion : X \quad S \sum p \quad £^{2} : p \, \Gamma \, s \quad X, \neg_{s} \quad S* \\ &p \, \mathrm{X} f p_{1}, p_{2} \mathrm{A} \mathrm{X} f x_{1}, s_{1} \mathrm{A} \Gamma f x_{2}, s_{2} \mathrm{A} \end{aligned} \tag{3.12}$$

Closing is the compound operation of dilation followed by erosion (with the same structuring element)

$$X \rfloor S \, \mathrm{X} f X \quad S \mathrm{A} \quad S \tag{3.13}$$

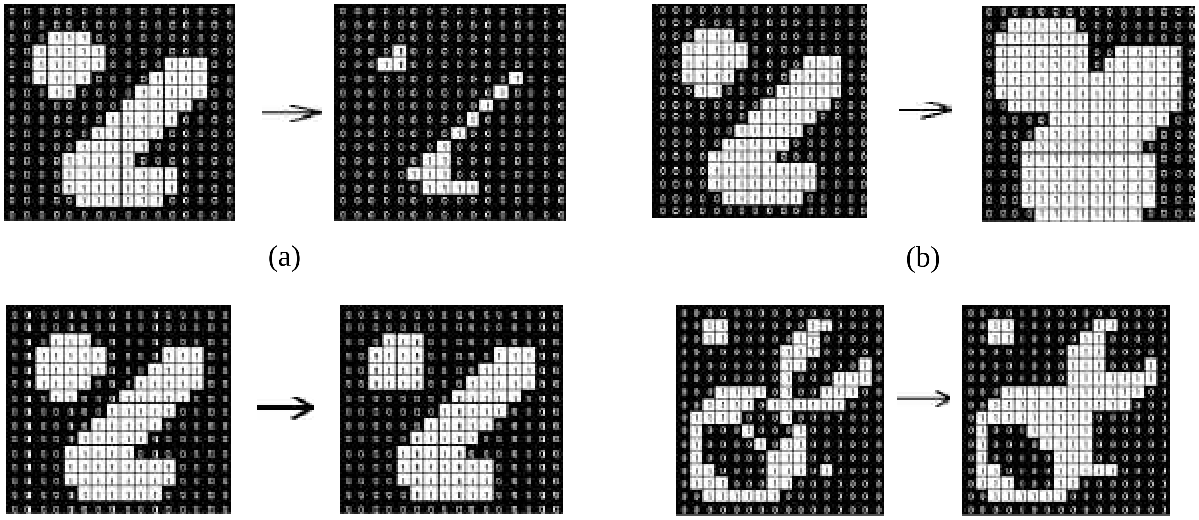
Has similar effects with dilation; but preserves global shape. (filling holes, gaps and gulfs)

Opening is the compound operation of erosion followed by dilation (with the same structuring element)

$$X \circ S \, \mathrm{X} f X \quad S \mathrm{A} \quad S \tag{3.14}$$

Has similar effects with erosion; but preserves global shape. (removing bridges, branches and small protrusions)

Figure 3.6 shows effect of different morphological operations (a) erosion (b) dilation (c) opening (d) closing using a 3×3 square structuring element.



(a) (b)

(c) (d)

Figure 3.6 Effect of (a) erosion (b) dilation (c) opening (d) closing using a 3×3 square structuring element

3.1.7 Projection Profile

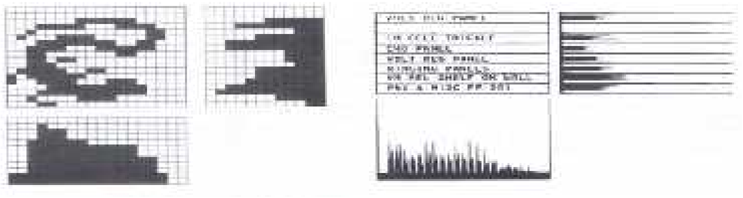
Projection Profile is a histogram of number of ON-pixels accumulated along parallel rows and columns of the image. The horizontal projection profile can be applied to identify the text line in a document image by groping the number of horizontal histogram, and then classified by means threshold. In the same time, the vertical projecting profile is designed for segmentation of columns in multi-column document images. Many researchers applied the horizontal and vertical projection profile to identify a character boundary. In my algorithm, the projection profile was applied to identify a circle boundary and the number of black pixels for a circle are used to be a basic criterion and stored in form library.

The projection of a binary object, may provide useful information related to object's shape. It can be found by summing all the pixels along the rows or columns. Summing the rows give horizontal projection. Summing the columns give the vertical projection.

We can define the horizontal projection *$h_i(r)$* and vertical projection *$v_i(c)$* as:

$$h_i(r) = \sum_{c=0}^{width-1} I_i(r,c) \qquad v_i(c) = \sum_{r=0}^{height-1} I_i(r,c) \qquad (3.15)$$

To find the projection, we sum the number of 1s in the rows and column. Figure 5.3 is an example of horizontal and vertical projection. Figure 5.3(a) is projection of image and figure 5.3(b) is projection of text image. Vertical projection shows that there are seven lines of text and vertical projection show number of character in each line of text.



(a) (b)

Figure 3.7 Horizontal and Vertical Projection profile

4 IMPLEMENTATION

All the algorithms of purposed handwriting recognition system are implemented in MATLAB R version 7.13.0.564 (R2011b). MATLAB is installed on a Intel(R) Core(TM) i5 CPU M460 @ 2.53GHz processor. The Computer has total main memory of 4 Gigabyte and 32-bit Microsoft Windows 7 Ultimate operating system installed in it.

4.1 MATLAB

The name MATLAB stands for MATrix LABoratory. MATLAB is high-level technical computing language having high-performance computation. MATLAB started life in the 1970s as a user-friendly interface. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB has very wide range of applications and can now be used as a very simple and transparent programming language where each line of code looks very much like the mathematical statement it is designed to implement. Typical uses of the MATLAB are given below.

- Mathematical and scientific computation.
- Algorithm development.
- Application development, including Graphical User Interface building.
- Data analysis, exploration and visualization.
- Modelling, simulation and prototyping.
- Graphics, in 2D and 3D, including colour, lighting and animation.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the tool of choice for high-productivity research, development and analysis. The reason behind using MATLAB for my research work experimentation is its wide variety and flexible toolboxes. Toolboxes are comprehensive collections of MATLAB functions (Mfiles) that extend the MATLAB environment to solve particular classes of problems. In this research work, mainly image processing toolbox and neural network toolbox are used along with other basis functionality of the MATLAB.

4.2 Image Processing Toolbox

Image Processing Toolbox (IPT) is very rich MATLAB toolbox for working with images. It includes many powerful and very efficient image processing functions. It provides a comprehensive set of standard algorithms and graphical tools for image processing, analysis, visualization and algorithm development. GUIDE toolbox (guidtool) is used for Graphical User Interface design. Image Acquisition Toolbox (imaqtool) is used for image Acquisition. Implementation of preprocessing steps using MATLAB is shown in section 4.3.

4.3 Preprocessing

4.3.1 Read image

Read single image

```
[fn,pn]=uigetfile('*.jpg','Select an image');
I = imread(fullfile(pn,fn));
```

Read images Directory

```
imageDir=uigetdir;
imageList = dir(fullfile(imageDir,'*.jpg'));
```

4.3.2 RGB to Gray Conversion Adjustment

```
I_gray=rgb2gray(I);
I_gray=imadjust(I_gray);
```

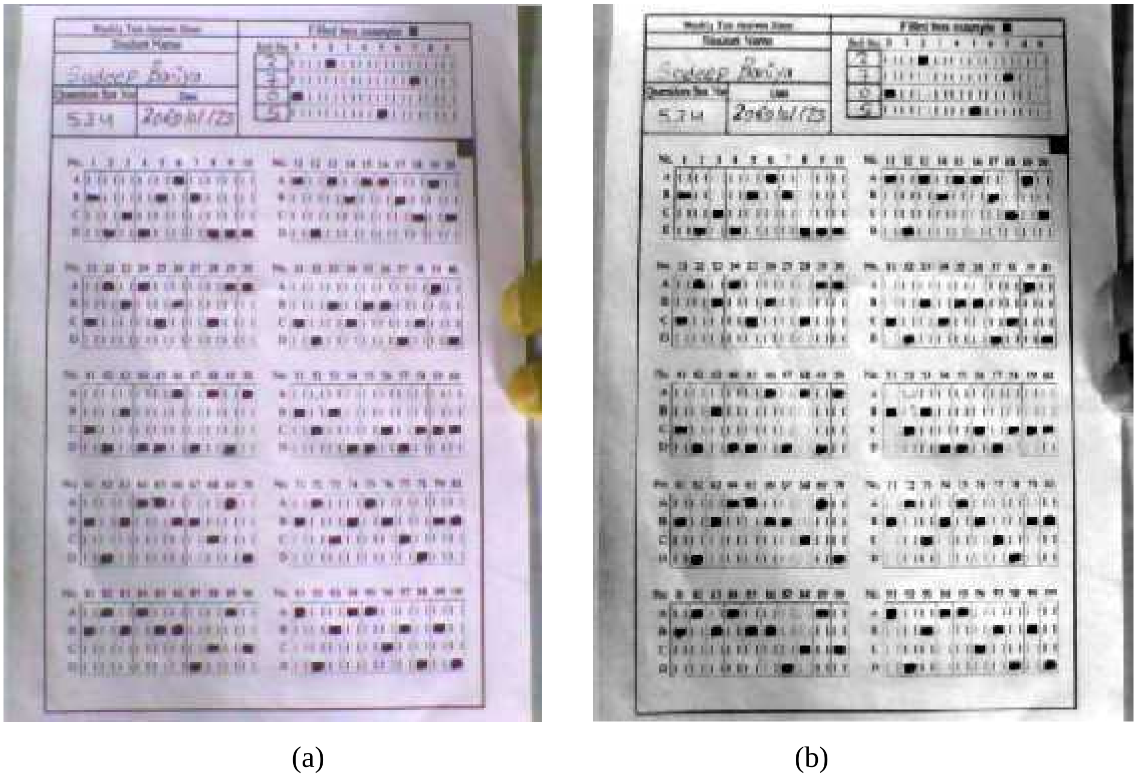
4.3.3 Edge Detection

```
BW = edge(I_gray, 'canny');
```

4.3.4 Hough transform

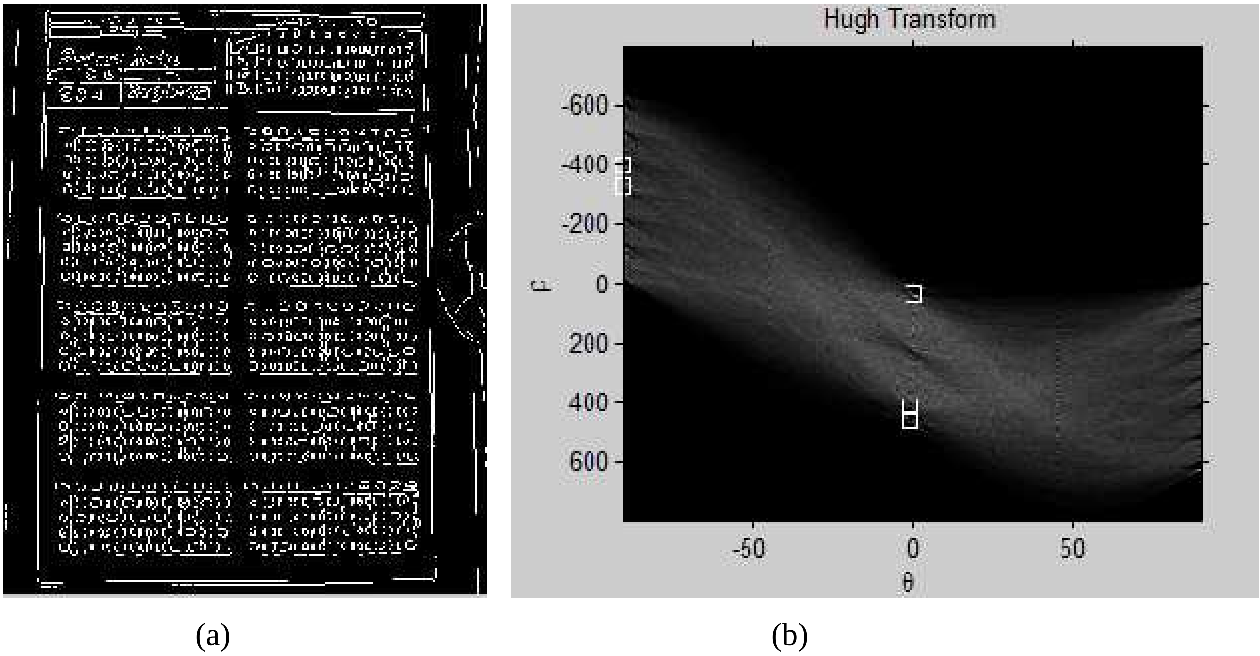
```
[H,T,R] = hough(BW);
skew = houghpeaks(H,5);
SkewAngle=skew(1,2);
```

Figure 4.1 (a) is answer sheet taken from low quality PC Camera; answer is marked in standard sheet given in Figure 2.2. By using MATLAB function `rgb2gray()& imadjust()` figure 4.1(a) converted to figure 4.1(b).



(a) (b)

Figure 4.1 (a) Input Image (b) Gray Image with adjust



(a) (b)

Figure 4.2 Edge Detection & Hough Transform

Figure 4.2 (a) is detection of edge of figure 4.1 (a) and figure 4.2(b) is Hough transform of figure 4.2(a). edge can be detected using MATLAB function `edge()& hough(), p`ick value of hough transform is detected by `houghpeaks()`function.

4.3.5 Adaptive Binary Conversion & Partial Rotation

Define Parameters

```
filt_radius = 10; % filter radius [pixels]
k_threshold = -0.2; % std threshold parameter
```

Build Filter

```
X = double(I_gray);
X = X / max(X(:)); % normalyze to [0, 1] range

fgrid = -filt_radius : filt_radius;
[x, y] = meshgrid(fgrid);
filt = sqrt(x .^ 2 + y .^ 2) <= filt_radius;
filt = filt / sum(filt(:));
```

Mean and Standard Deviation Calculation

```
local_mean = imfilter(X, filt, 'symmetric');
local_std = sqrt(imfilter(X .^ 2, filt, 'symmetric'));
```

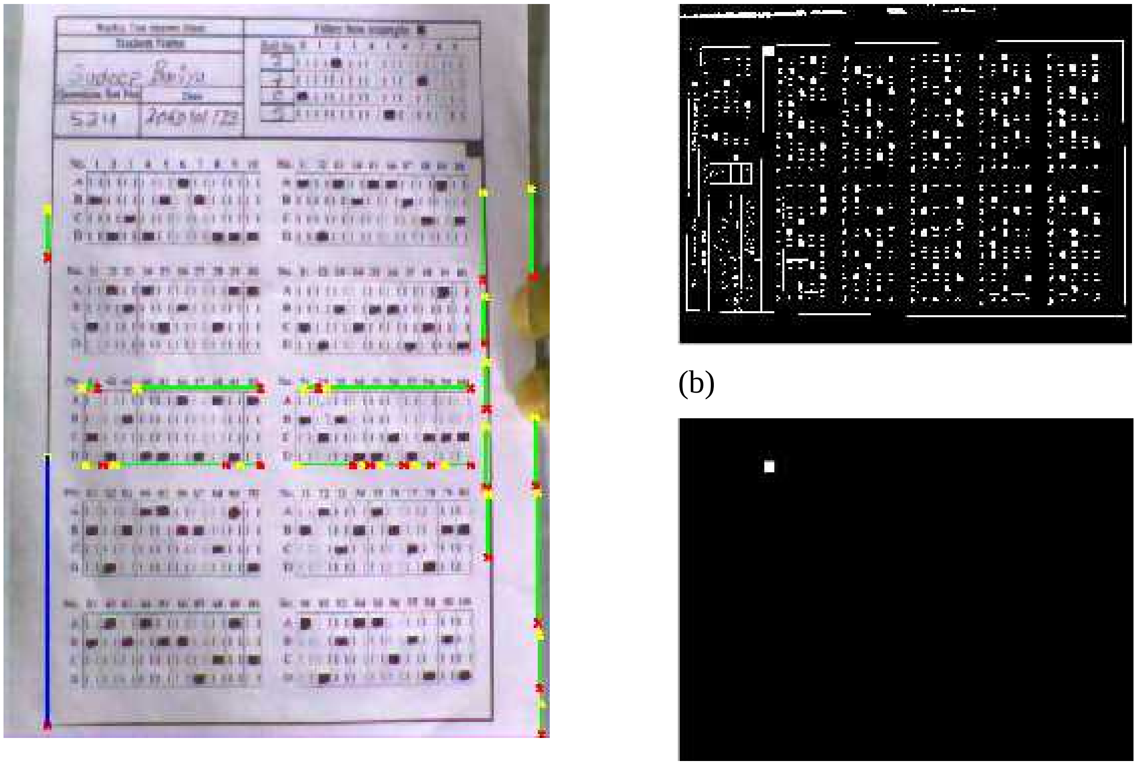
Binary image Calculation

```
X_bin = X >= (local_mean + k_threshold * local_std);
```

Partial Rotatioin

```
rotI = imrotate(~X_bin,SkewAngle);
```

Figure 4.3 (a) is marked standard answer sheet with detection of lines. Detection of slope of line help to correct skew of image since there is only horizontal or vertical lines. Figure 4.3(b) is binary image after rotation of 4.3(a). 4.3(b) is not exactly skewed since longest line either towards vertical or horizontal so it will skew by checking position of reference square. Figure 4.3(c) shows the reference point detection.



(b)

(a) (c)

Figure 4.3 (a) Line Detection, (b) Binary Conversion & (c) Reference square detection

4.3.6 Reference Point Detection final skew correction

```
img = imclose(rotI,strel('square',12));
[r, c] = find(lblImg==1);
rc = [r c];
r1 = rc(1,1);    r2 = rc(end,1);
c1 = rc(1,2);    c2 = rc(end,2);
dist = sqrt((r2-r1)^2+(c2-c1)^2); %distance of first and last pt.
skw = 0;
[rows,cols]=size(rotI);
if(rc(1,1)<rows/2)
   if(rc(1,2)<cols/2)
       skw = -90;
      else
       skw = 0;
   end
end
if(rc(1,1)>rows/2)
   if(rc(1,2)<cols/2)
       skw = 180;
      else
       skw = 90;
   end
end
skwImg = imrotate(rotI, skw);
```

4.3.7 Crop out Roll Number Field and Answer Field

```
    [rows,cols]=size(skwImg);
    imRoll = imcrop(skwImg,[0 0 c2 r1]); %Roll Number Area
    imAnswer = imcrop(skwImg,[0 r1 cols rows-r2]); % Answer Area
```

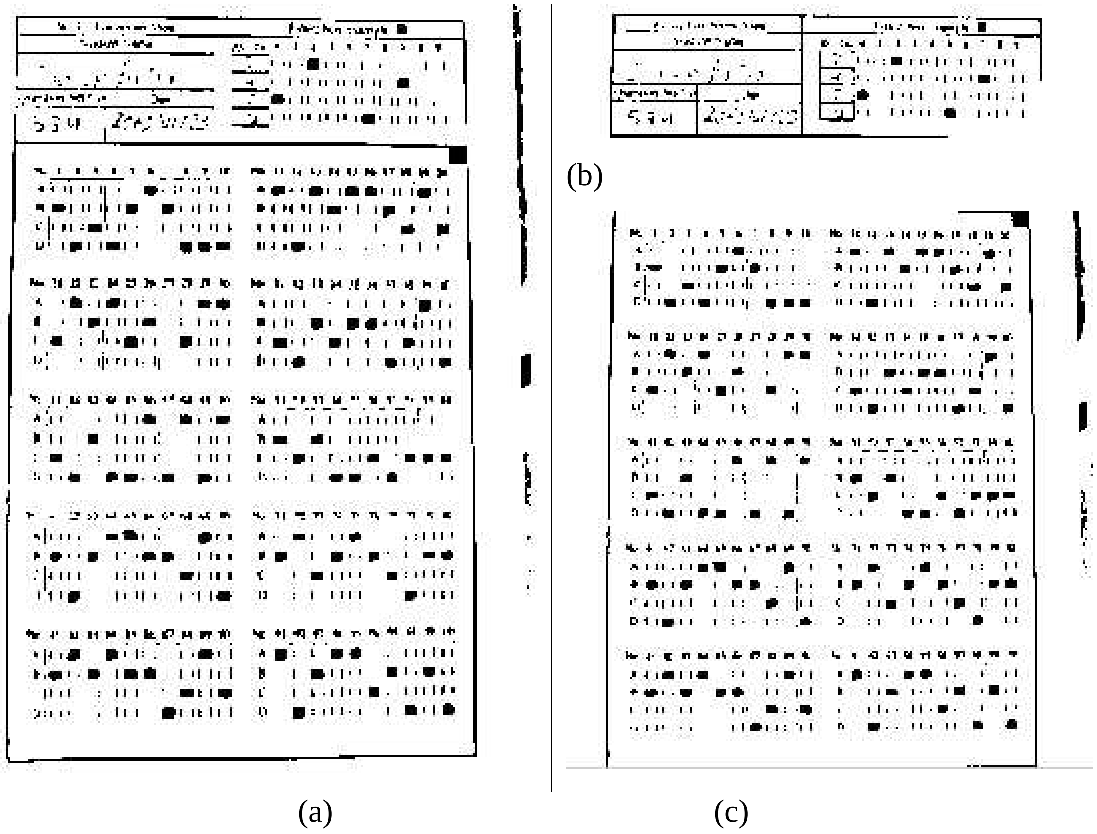


Figure 4.4 (a) Skew Correction, (b) Roll Area & (c) Answer Area

Figure 4.4(a) is binary image which is fully skewed. 4.4(b) and 4.4(c) are Roll No area and Answer area respectively which are separated by using reference square.

4.3.8 Remove unwanted large Areas

```
    im1= ~imAnswer;
    im3 = bwareaopen(im1,500);
    im4 = im1-im3;
```

4.3.9 Projection Profile

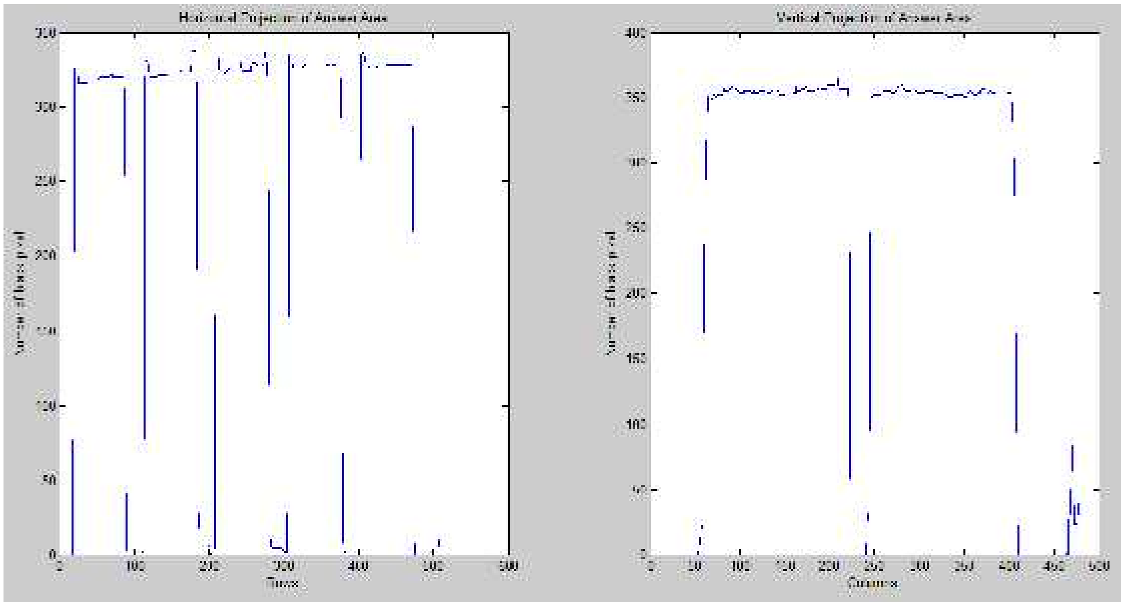
```
    %vertical Projection
    img = ~imopen(~im4,strel('square',15));

    verticalProfile = sum(img, 1); % Sum along columns
    xv = 1:colAns;
    yv = verticalProfile(xv);


    %Horizontal Projection
```

```
        horizontalProfile = sum(img, 2); % Sum along rows
        xh = 1:rowAns;
        yh = horizontalProfile(xh);
        figure,
        subplot(1,2,1),plot(xh,yh);
        title('Horizontal Projection of Answer Area'),
        xlabel('Rows'),ylabel('Number of black pixel');

        subplot(1,2,2),plot(xv,yv);
        title('Vertical Projection of Answer Area'),
        xlabel('Rows'),ylabel('Number of black pixel');
```



(a) (b)

Figure 4.5 (a) Horizontal Projection, (b) Vertical Projection

Figure 4.5(a) is horizontal projection of figure 4.4(c) and figure 4.5(b) is vertical projection. Projection is done after morphological operation on figure 4.4(c).

4.3.10 Horizontal Crop Function

```
function [image_hCrop] = hCrop(imgH,A_or_R)
        addWhite=imgH(1:15,1:end);
        addWhite(:,:)=1;
        imgH=[addWhite;imgH;addWhite];
        imageInputH=imgH;
        if(A_or_R=='A');
            imgH = ~imopen(imgH,strel('square',15));
        end
        if(A_or_R=='R')
            imgH = ~imopen(imgH,strel('square',30));
        end
```

```
      imgH = bwareaopen(imgH,3000);
      [~,colAns]=size(imgH);
      %Horizontal Projection
      horizontalProfile = sum(imgH, 2); % Sum along rows

      %from horizontal Projection.
      lowestValueY = find(horizontalProfile<40);

      sizeLowestValueY = size(lowestValueY);
      DiffY2=diff(lowestValueY);
      DiffY1=lowestValueY(1:sizeLowestValueY-1);
      DiffY = [DiffY1';DiffY2']';
      sortY = sortrows(DiffY,2);
      yCords = sortY(1:end,1:2);
      yCords = sortrows(yCords,1);

      n=1;
      for i=1:size(yCords);
              if(yCords(i,2)>40) %if histogram width meet greather than
      40
                  image_h{i}= imageInputH(yCords(i,1):yCords(i,1)+
      yCords(i,2),1:colAns);
                  %get only non empty image.
                  if(~isempty(image_h{i}));
                      image_hCrop{n}=image_h{i};
                      n=n+1;
                  end

              end
      end
end
```

4.3.11 Vertical Crop Function

```
function [image_vCrop] = vCrop(imgV)
      imageInputV=imgV;

      %imgForCrop = imclose(imgV,strel('square',5));
      imgForCrop=imgV;
      imgV = ~imopen(imgV,strel('square',15));
      imgV = bwareaopen(imgV,2000);

      [rowAns,~]=size(imgV);
      %Vertical Projection
      verticalProfile = sum(imgV, 1); % Sum along columns

      %from vertical Projection.
      lowestValueX = find(verticalProfile<5);
      sizeLowestValueX = size(lowestValueX);
      DiffX2=diff(lowestValueX);
```

```
        DiffX1=lowestValueX(1:sizeLowestValueX(2)-1);
        DiffX = [DiffX1;DiffX2]';
        sortX = sortrows(DiffX,2);
        xCords = sortX(1:end,1:2);
        xCords = sortrows(xCords,1);



        n=1;
        for i=1:size(xCords)-1;
                if(xCords(i,2)>100);

                    image_v{i}=
        imgForCrop(1:rowAns,xCords(i,1):xCords(i,1)+xCords(i,2));
                    %get only non empty image.
                    if(~isempty(image_v{i}));
                        image_vCrop{n}=image_v{i};
                        n=n+1;
                    end
                end
        end
end
```

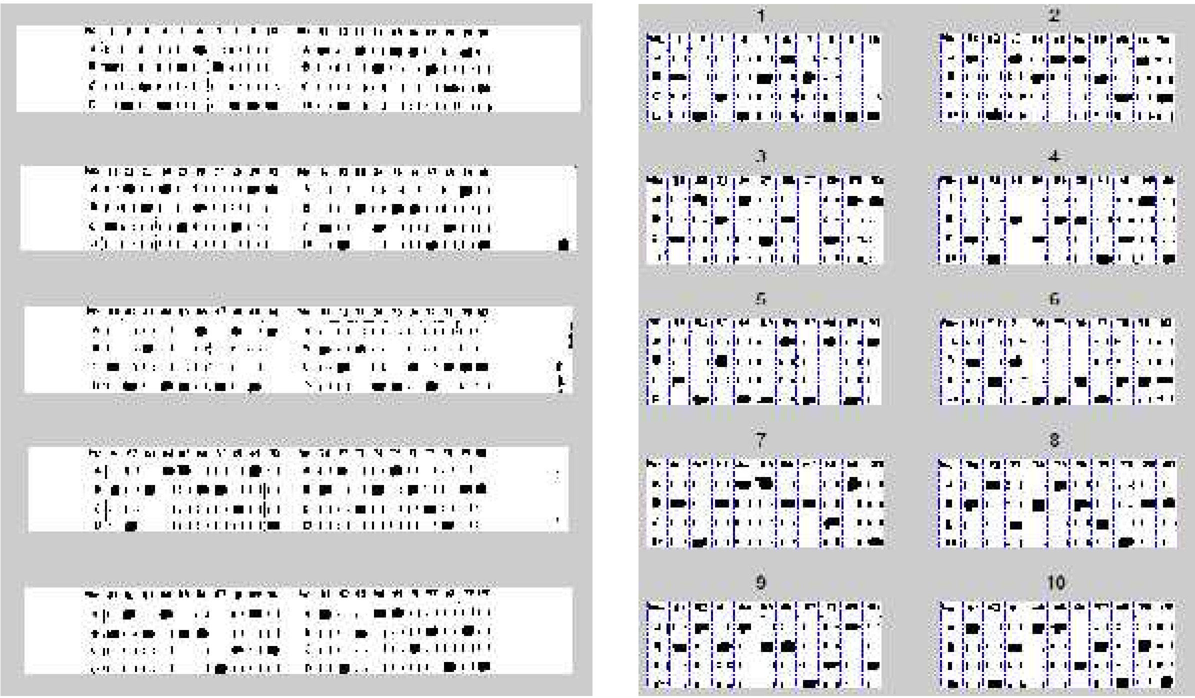
4.3.12 Answer Extraction Function

```
function [questionAnswer] = getAns(imAnswer)
        %% Remove large, unwanted Areas
        im1= ~imAnswer;
        im3 = bwareaopen(im1,500);
        im4 = im1-im3;
```

Sub Answer Area Extraction

```
        imageH=hCrop(~im4,'A');
        for i=1:size(imageH,2)
            subImage1{i} = vCrop(imageH{i});
            for j=1:size(subImage1{i},2);
                 subImage{i,j} = hCrop(subImage1{1,i}{1,j},'A');
              % remove white space at top and buttom of each sub area.
            end
        end


        for i=1:size(subImage,1);
            for j=1:size(subImage,2);
                 subAnsArea{i,j}=imresize(subImage{i,j}{1,1},[50,110]);
            end
        end
```



(a) (b)

Figure 4.6 (a) Horizontal Cropping (b) Vertical Cropping

Horizontal projection determines the number of rows of answer blocks as shown in figure 4.6(a). Vertical projection determines the number of columns of answer block as shown in figure 4.6(b). if there is R number of rows and C number of columns then it determine that there will R*C number of answer blocks each contains ten question. It determines total number of question in answer sheet. Total number of question = R*C*10.

Each Question area Extraction

```
 n=1;
 for i=1:(size(subAnsArea,1));
     for j=1:(size(subAnsArea,2));
         for l=1:10;
           aNumberArea{n}=subAnsArea{i,j}(11:50,l*10+2:(l+1)*10);
           n=n+1;
         end
     end
 end
se=[1,1,1,1;1,1,1,1];
for n=1:size(aNumberArea,2);
     Question{n} = imclose(aNumberArea{n},se);
end
```

Area Calculation

```
allArea=[,];
y=1:10:50;
for n=1:size(Question,2);
     for i=1:4
        region{n,i}= Question{n}(y(i)+1:y(i)+8,2:8);
        area=0;
        for k=1:8;
            for l=1:5;
                if(region{n,i}(k,l)==0)
                   area=area+1;
                end
            end
        end
        Area{n,i}=area;
     end
end
```

Algorithm to Conform Answer

```
for n=1:size(Area,1);
   for i=1:4;
        qAnswer{n,i}= 0;
   end;
   [Y,I] = sort(cell2mat(Area(n,:))); % sort by value of vector
   i_Area=[I;Y]'; % indeces and Area
   if(i_Area(4,2)>=6);
       qAnswer{n,i_Area(4,1)}=1; %if only one answer is selected

       if(i_Area(3,2)>=6) %if 2 answer are selected
            if((i_Area(4,2)-i_Area(3,2))<6)
                qAnswer{n,i_Area(3,1)}=1;
            end
       end

       if(i_Area(2,2)>=6) %if 3 answer are selected
            if((i_Area(4,2)-i_Area(2,2))<6)
                qAnswer{n,i_Area(2,1)}=1;
           end
       end

       if(i_Area(1,2)>=6)%if 4 answer are selected
           if((i_Area(4,2)-i_Area(1,2))<6)
                qAnswer{n,i_Area(1,1)}=1;
           end
       end
   end
end
```

Alphabetical Answer Conversion

```
      questionAnswer={};
      for i=1:size(qAnswer,1);
          Answer=[];
          for j=1:size(qAnswer,2);
              if(cell2mat(qAnswer(i,j))==1);
                  if j==1;Answer=strcat(Answer,'A');end;
                  if j==2;Answer=strcat(Answer,'B');end;
                  if j==3;Answer=strcat(Answer,'C');end;
                  if j==4;Answer=strcat(Answer,'D');end;
              end
          end
          if(isempty(Answer));
              Answer = '-';
          end
          questionAnswer{i,1} = Answer;
      end
end
```

4.3.13 Roll No Extraction Function

```
function [rollNo] = getRoll(imRoll)
      im1= ~imRoll;
      im3 = bwareaopen(im1,800);
      im4 = im1-im3;
      imageH=hCrop(~im4,'R');
      subImage1 = vCrop(imageH{1,1});
      subImage = hCrop(subImage1{1,end},'R');
      rollArea = subImage{1,1};
      rollArea = imresize(rollArea,[120,240]);
      rollArea = rollArea(40:120,40:240);
```

Roll No Cell Extraction

```
      se=strel('square',5);
      x=1;
      for i=1:20:80;
          y=1;
          for j=1:20:200;
              rollMarkArea{x,y}=imclose(rollArea(i:i+20,j:j+20),se);
              rollMarkArea{x,y}(1:end,1:4)=1;
              rollMarkArea{x,y}(1:end,end-4:end)=1;
              rollMarkArea{x,y}(1:4,1:end)=1;
              rollMarkArea{x,y}(end-4:end,1:end)=1;
              y=y+1;
          end
          x=x+1;
      end;
```

Area Calculation

```
for i=1:size(rollMarkArea,1);
     for j=1:size(rollMarkArea,2);
        area=0;
        for k=1:21;
            for l=1:21;
                if(rollMarkArea{i,j}(k,l)==0)
                   area=area+1;
                end
            end
        end
        rArea{i,j}=area;
     end
end
```

Roll No Calculation

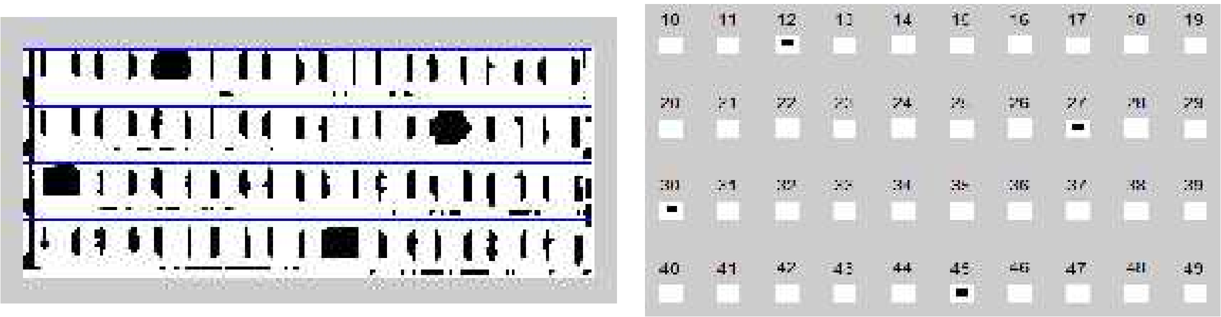
```
rRef=1;
for i=1:size(rArea,1);
    for j=1:size(rArea,2);
        rMark{i,j}= 0;
    end;

   [Y,I] = sort(cell2mat(rArea(i,:))); % sort by value of vector
   i_Area=[I;Y]'; % indeces and Area

   if(i_Area(10,2)>=20);
       %if only one answer is selected
       rMark{i,i_Area(10,1)}=1;

       %if 2 answer are selected
       if(i_Area(9,2)>=20)
            if((i_Area(10,2)-i_Area(9,2))<20)
                rMark{i,i_Area(9,1)}=1;
            end
       end
   end
end
```

```
      for i=1:4;
           if(sum(cell2mat(rMark(i:i,:)))~=1)
               rollNo=0;
           else
               j=find(cell2mat(rMark(i,:))==1);
               rollNo(i)=j-1;
           end
      end
end
```



(a) (b)

Figure 4.7 (a) Roll No Area (b) Roll Number cell

Roll No block is extracted from the Roll No area using MATLAB code of section 4.3.13. Morphological close operation reduces unnecessary black pixels. Figure 4.7(a) is Roll No block extracted from Roll No area and figure 4.7(b) shows that roll Number is 2705. Maximum black pixel in first position is in third position from zero, i.e. 2 and similarly second, third and fourth positions number is determined as 7,0 & 5 respectively.

4.3.14 Result Calculating Function

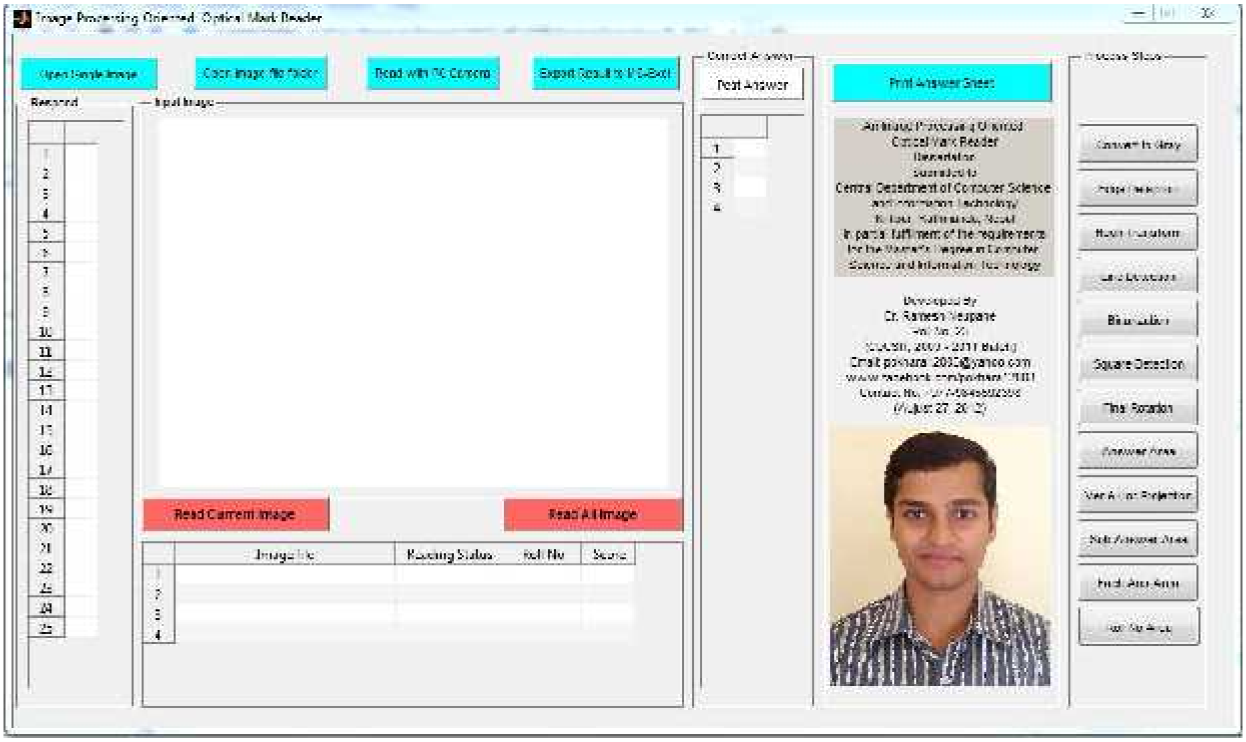
```
function [data, mark] = getResult(markedAnswer, correctAnswer)
     data={};
     mark=0;
     an=[];
     ref=0;
     n=size(markedAnswer,1);
     l=n;
     if(size(markedAnswer,1)>size(correctAnswer,1));
         n=size(correctAnswer,1);
         an=n+1;
         ref=1;
     end

     for i=1:n;
        a=markedAnswer{i,1}(1,:);
         if (markedAnswer{i,1}==correctAnswer{i,1})
             markedAnswer{i,2}=1;
             mark=mark+1;
                data{i,1}=strcat('<html><table border=0 width=400
                                bgcolor=#FFFFFF><TR><TD>',...
                                a,'</TD></TR> </table></html>');
         else
             markedAnswer{i,2}=0;
                data{i,1}=strcat('<html><table border=0 width=400
                                bgcolor=#FF82AA><TR><TD>',...
                                a,'</TD></TR> </table></html>');
```

```
            end
        end

        if(ref==1);
            for i = an:l;
                a=markedAnswer{i,1}(1,:);
                markedAnswer{i,2}=0;
                data{i,1}=strcat('<html><table border=0 width=400
                                 bgcolor=#FF82AA><TR><TD>',...
                                 a,'</TD></TR> </table></html>');
            end
        end
end
```

4.3.15 GUI Design



[illegible] Processing Oriented
Optical Mark Reader
Dissertation
Submitted to
Central Department of Computer Science
and Information Technology
Kirtipur, Kathmandu, Nepal
In partial fulfillment of the requirements
for the Master's Degree in Computer
Science and Information Technology

Developed By
Mr. Ramesh Neupane
Roll No. [illegible]
CDCSIT, 2009 - 2011 Batch
Email: [illegible]@yahoo.com
www.facebook.com/[illegible]
Contact No. [illegible]
(August 27, 2012)

Figure 4.8 GUI Interface Design

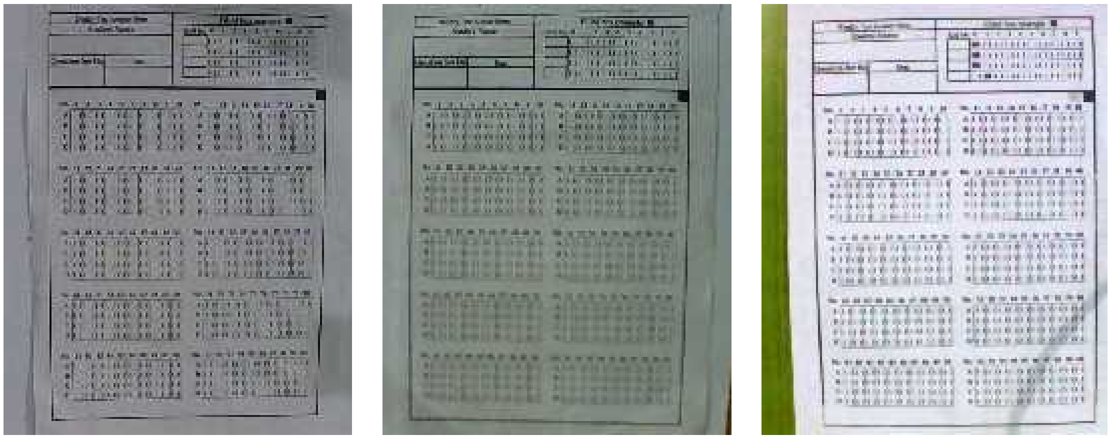
User interface is designed as soon in figure 4.8.Appendix A.1 shows detail of user interface.

5 EXPERIMENTATION AND RESULTS

There are mainly two critical points in this system. One is thresholding and Answer selection. All the system depends on thresholding. If it will not being correctly all system will effected and result will be incorrect. So a lot of algorithms are experimented, compared and modified to get best algorithm to fulfill requirement of this system.

To fulfill the objective of this system, system should have high accuracy, efficient, low cost and should be read dynamic images that will design by user. For experiment propose images are categories in to six types, Blank Sheets (B) ,Normal Sheets (N), Incorrectly Marked Sheets (IM), External Object Mixed Sheets (EM), Incorrectly Roll No marked Sheets (IR) and Different No of question of images (DN).

5.1 Sampled Images for Experiment



B1 B2 B3

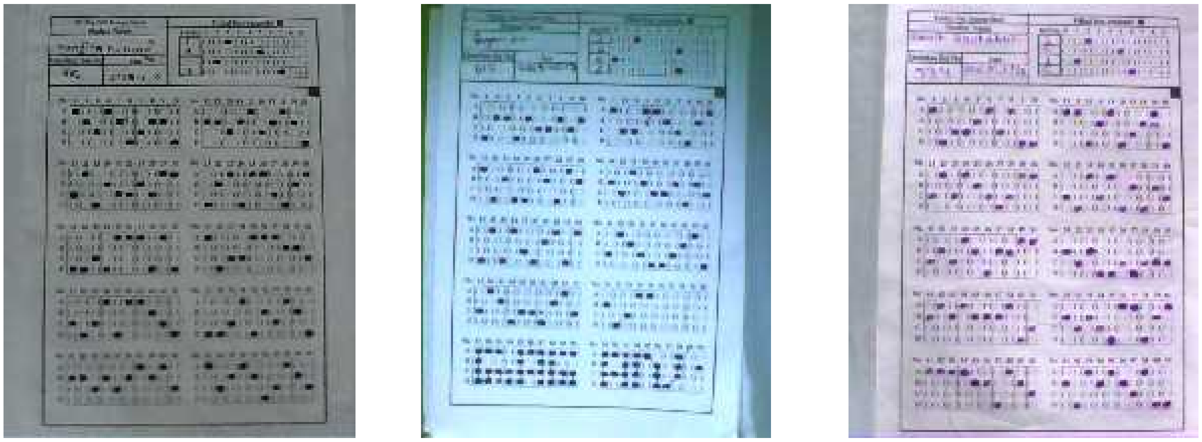
Figure 5.1 Blank Sheets

5.1.1 Blank Sheets (B)

Figure 5.1 shows three different blank sheets, taken from different intensity and different PC Camera which are used in our experiment in this chapter. Blank sheets are name as B1, B2 & B3.

5.1.2 Normal Sheets (N)

Figure 5.2 shows differently marked sheets. Images are named as N1, N2 & N3. N1 is normal sheet of 100 Questions all are marked, in sheet N2 there is both multiple selection and non selection of single question can be seen. N3 has shadow behind sheets. In our experiment those types of sheets are considered as normal sheets.



N1 N2 N3

Figure 5.2 Normal Sheets

5.1.3 Incorrectly Marked Sheets (IM)

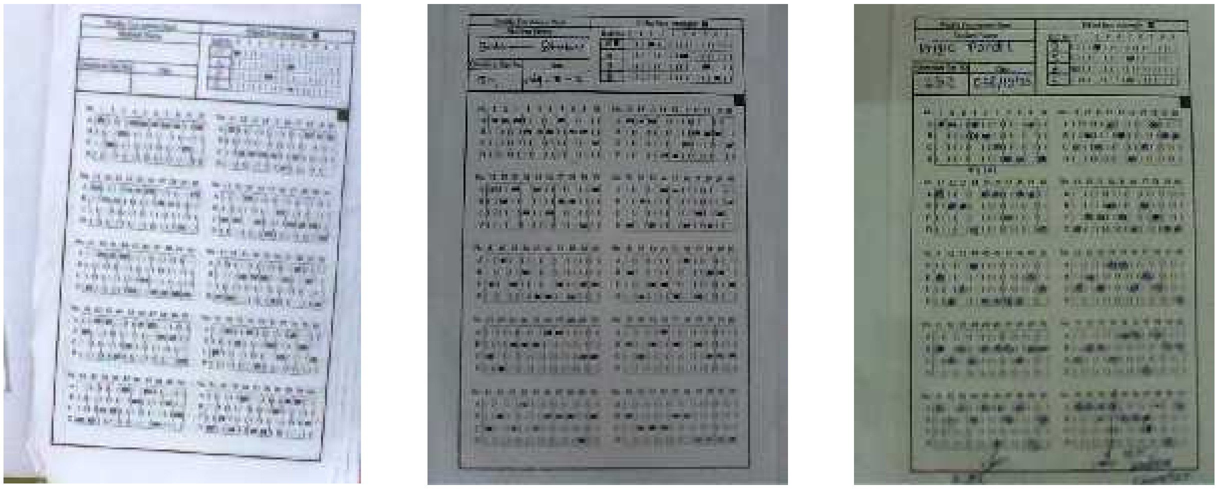
Figure 5.3 shows such sheets which are not marked correctly. Small marking, light marking, large marking etc are considered as incorrectly marked. Sheets M1, M2 & M3 shows light marking, small marking and large marking respectively.

5.1.4 External Object Mixed Sheets (EM)

Figure 5.4 shows such sheets which are disturbed by external object. In sheet EM1, EM2 & EM3 hand is seen in left, top and right site of sheets respectively.

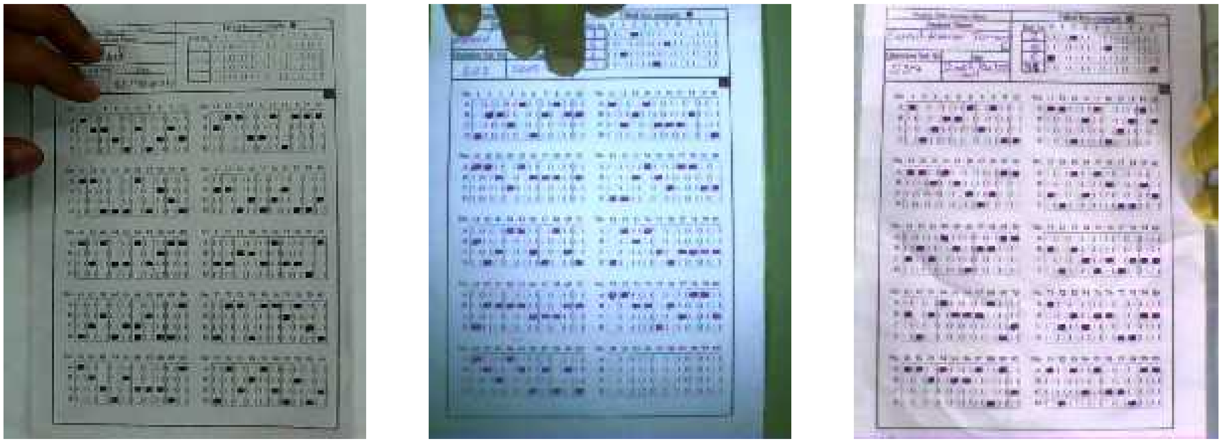
5.1.5 Incorrectly Roll No marked Sheets (IR)

Figure 5.5 is concern on roll number, some time roll number is not marked correctly. In sheet IR1 first position of roll number is marked at 2 & 6, it is not possible to take more than one number in a position. So it is necessary to mark exactly one mark each row of roll number area. More than one mark or blank in a row considered as roll Number error. Sheet IR2 shows all number blank and sheet IR3 shows missing mark on 4th position and double mark in 3rd position. Our system should detect such errors that are shown in experiment of section 5.5.



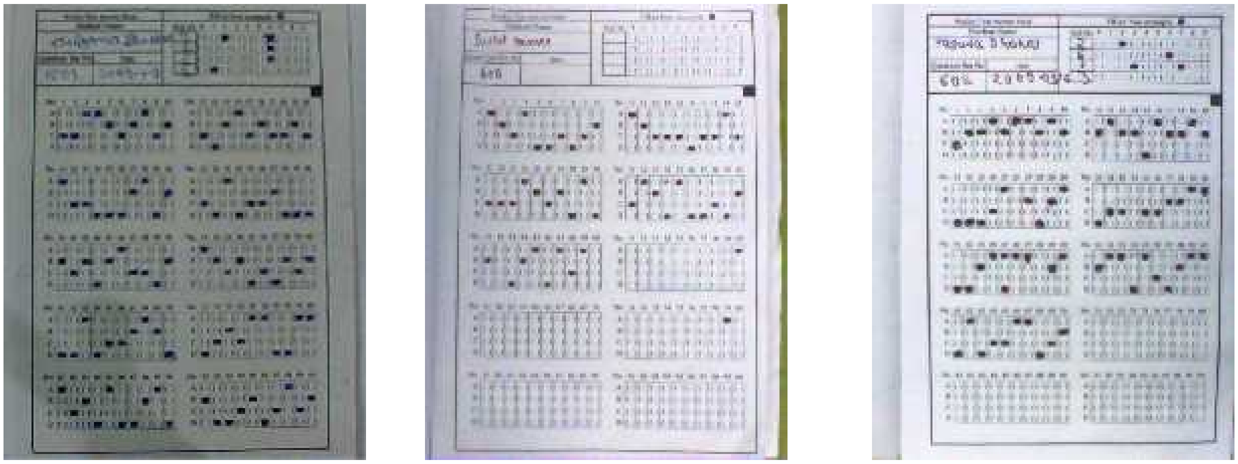
IM1 IM2 IM3

Figure 5.3 Incorrectly Marked Sheets



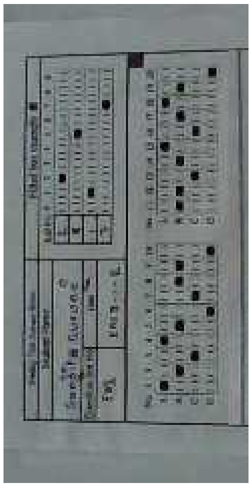
EM1 EM2 EM3

Figure 5.4 External Object Mixed Sheets

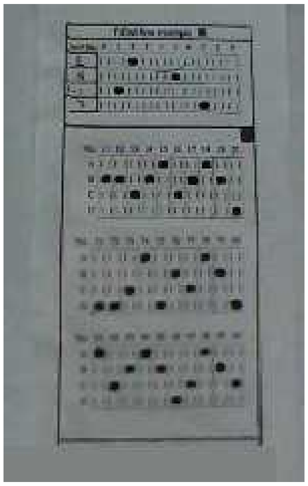


IR1 IR2 IR3

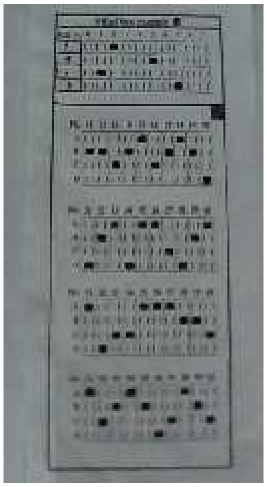
Figure 5.5 Sheets with Incorrectly marked Roll No



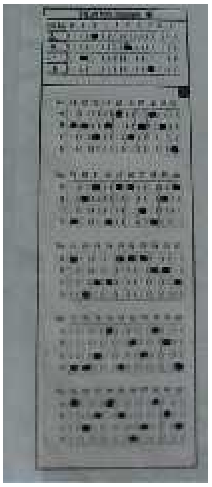
DN1



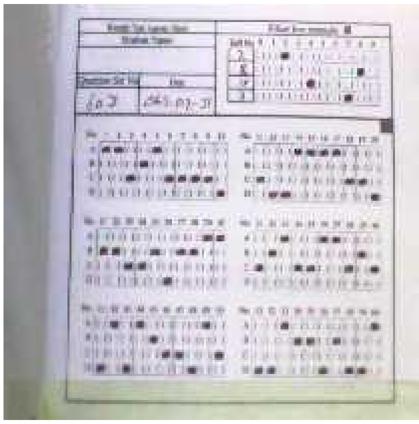
DN2



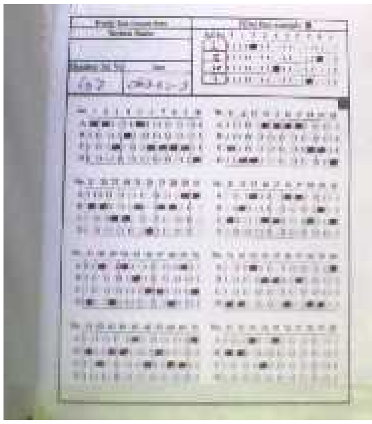
DN3



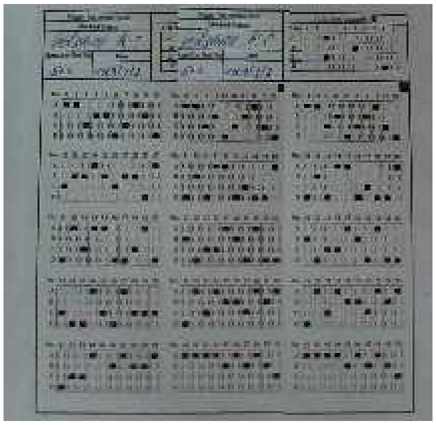
DN4



DN5



DN6

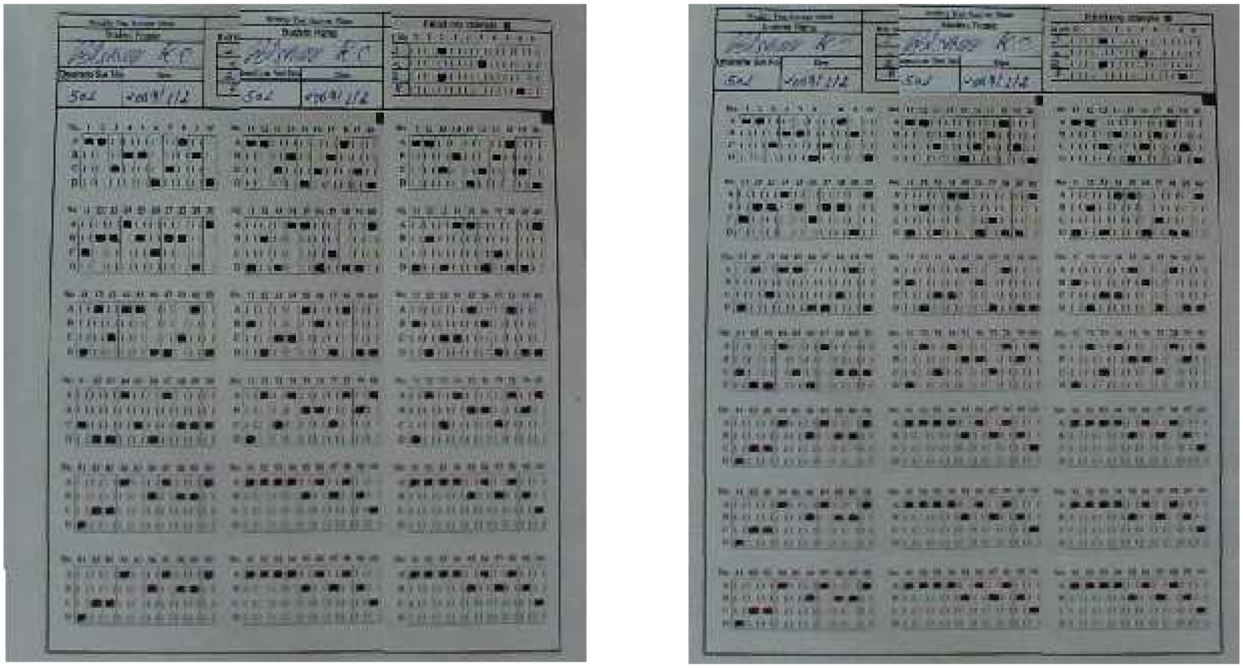


DN7

5.1.6 Different number of question of images (DN)

Figure 5.6 contains nine different sheets with different number of questions. Our system should read all of this correctly. Sheets DN1, DN2, DN3, DN4, DN5, DN6, DN7, DN8, DN9 has 20, 30, 40, 50, 60, 80, 120, 180, 210 number of questions respectively in each sheets.

There is also experiment the result of other such type of sheets in this chapter. To fulfill our objective all type of answer sheets that described in section 5.1 should read correctly by this system. Those experiments are described in later section of this chapter.



DN8 DN9

Figure 5.6 Sheets with different number of question

From Table 5.1 we can conclude that Niblack’s and Sauvola’s algorithm is similar in general case. Our modification on Niblack’s Algorithm by adding filature and changing value of k to 0.-1 is sufficient for our all types of sheets.

Area of each region is just the number of black pixel inside it. Table 5.2 show how area is converted to selected answer. There will some critical points to convert area to answer, because some time small dots should be detected as answer and sometime large area should reject due relative mark area. For example in Q.No 20 of B1, area of 6 is marked as answer but in Q.No. 81 & 85 of N2, area of 15 and 17 are not marked as answer. This is possible due to relative answer detection method used in the system. The system marked as answer if area is above 6 and there should be difference of 6 with largest area to mark more than one answer for single question. In case of Q.No 20 of B1 areas are detected as A=0, B=0, C=0 and D=6, so D is marked as answer. In case of Q.No. 85 of N2 areas are detected as A=24, B=20, C=19, D=17, in this condition A,B & C are only mark as answer D is not mark as answer because 24-17=7 > 6.

5.2 Image Binarization

The result of binarization using different algorithm are shown in table 5.1.

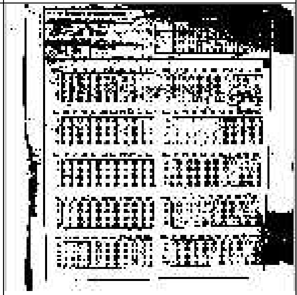
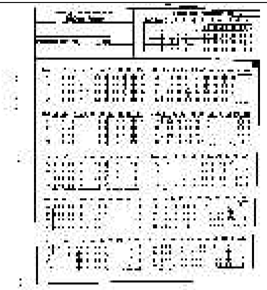
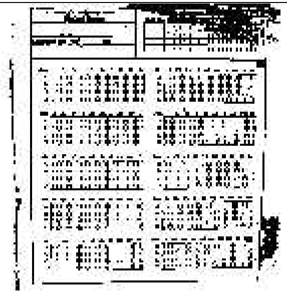
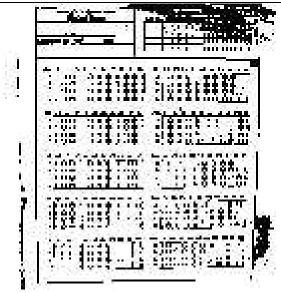
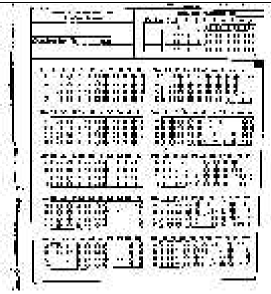
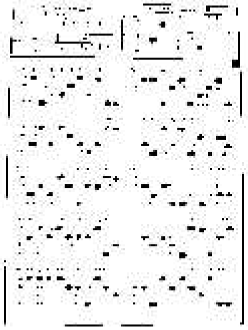
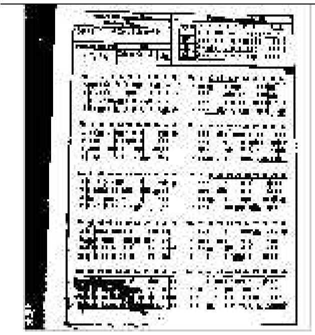
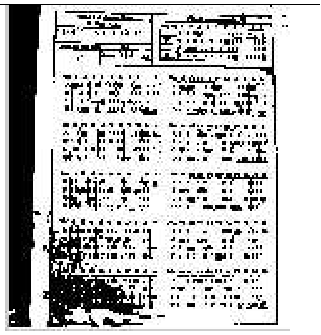
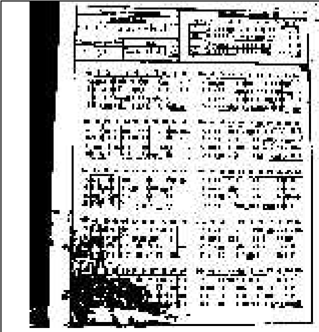
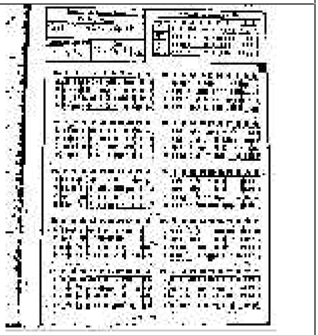
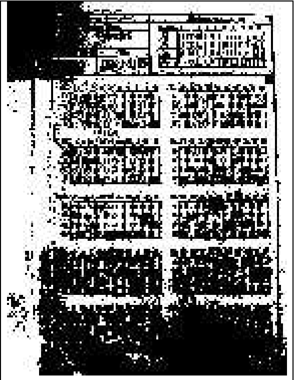
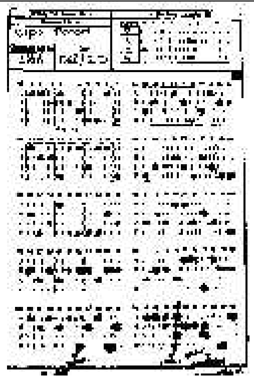
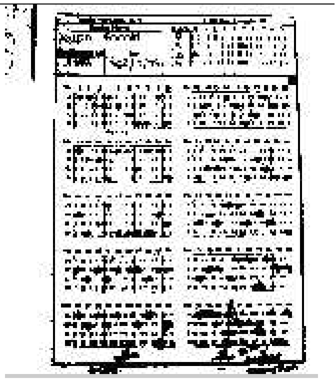
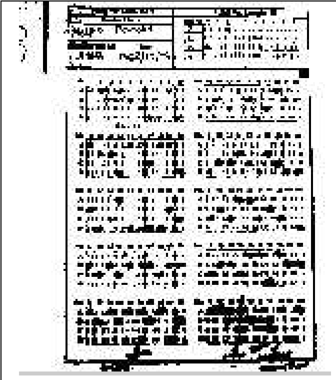
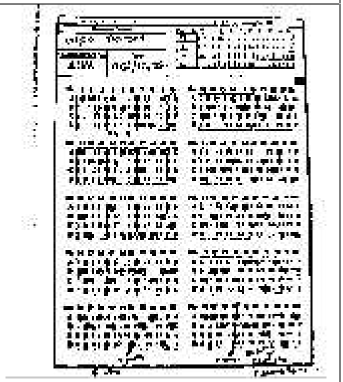
Image	Normal BW	Ostu's Algorithm	Niblack's Algorithm	Sauvola's Algorithm	Used Algorithm
B1					
N3					
IM3					

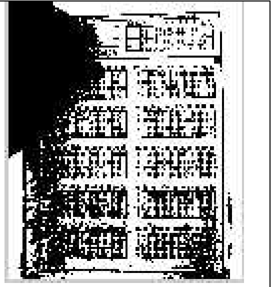
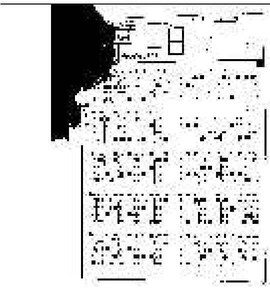
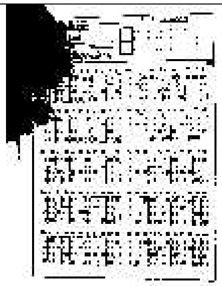
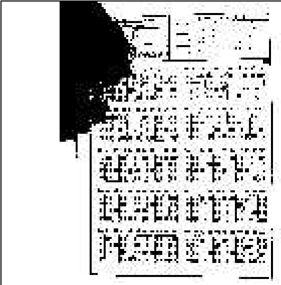
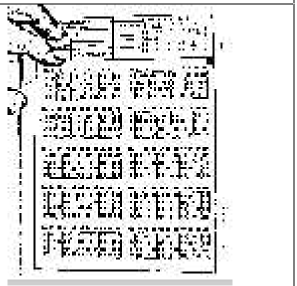
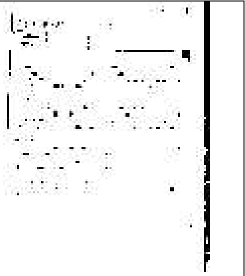
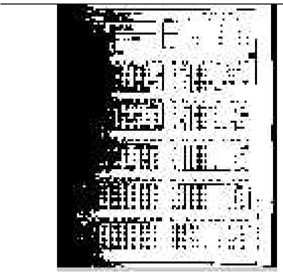
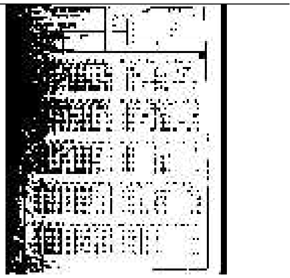
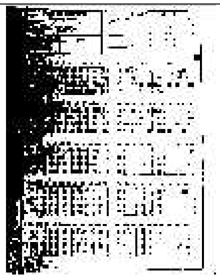
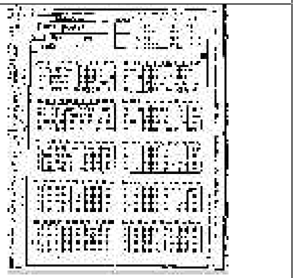
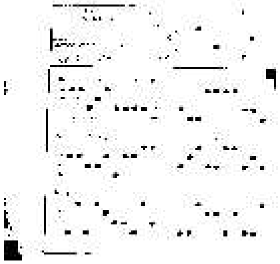
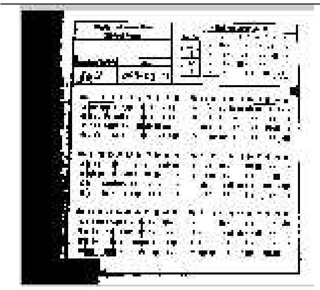
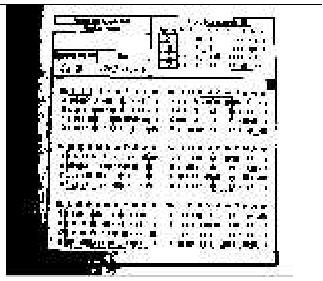
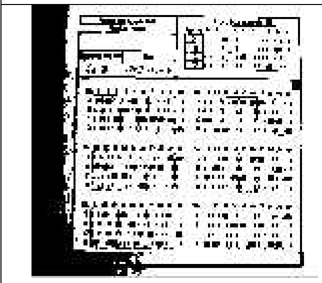
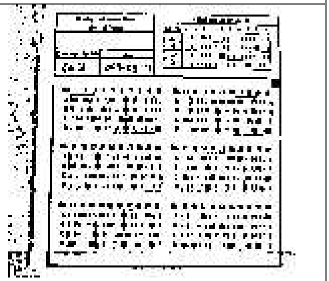
Image	Normal BW	Ostu’s Algorithm	Niblack’s Algorithm	Sauvola’s Algorithm	Used Algorithm
EM1					
IR2					
DN5					

Table 5.1 Comparison of Different algorithm for Image Binarization

5.3 Area Calculation and Answer Conformation

Sheet	Q.	No of Black Px				Sorted Area				Difference				Answer (T/F)				Result	
	No	A	B	C	D	i	j	k	l	l-i	l-j	l-k	l	A	B	C	D	Ans Area	Answer
B1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-
B1	5	5	5	4	0	0	4	5	5	5	1	0	5	0	0	0	0	-	-
B1	7	4	4	4	0	0	4	4	4	4	0	0	4	0	0	0	0	-	-
B1	15	0	0	5	0	0	0	0	5	5	5	5	5	0	0	0	0	-	-
B1	20	0	0	0	6	0	0	0	6	6	6	6	6	0	0	0	1	6	D
B1	36	2	0	0	8	0	0	2	8	8	8	6	8	0	0	0	1	8	D
B1	48	0	0	0	10	0	0	0	10	10	10	10	10	0	0	0	1	10	D
B1	49	0	0	4	14	0	0	4	14	14	14	10	14	0	0	0	1	14	D
B2	26	2	4	4	4	2	4	4	4	2	0	0	4	0	0	0	0	-	-
B3	90	6	0	0	6	0	0	6	6	6	6	0	6	1	0	0	1	6,6	AD
B3	91	0	0	6	6	0	0	6	6	6	6	0	6	0	0	1	1	6,6	CD
B3	78	6	7	6	2	2	6	6	7	5	1	1	7	1	1	1	0	6,7,6	ABC
N1	42	4	4	3	25	3	4	4	25	22	21	21	25	0	0	0	1	25	D
N1	99	0	24	2	2	0	2	2	24	24	22	22	24	0	1	0	0	24	B
N2	80	3	0	0	0	0	0	0	3	3	3	3	3	0	0	0	0	3	-
N2	81	23	20	20	15	15	20	20	23	8	3	3	23	1	1	1	0	23,20,20	ABC
N2	82	23	0	19	19	0	19	19	23	23	4	4	23	1	0	1	1	23,19,19	ACD
N2	83	22	0	21	15	0	15	21	22	22	7	1	22	1	0	1	0	22,21	AC
N2	84	0	0	16	19	0	0	16	19	19	19	3	19	0	0	1	1	16,19	CD
N2	85	24	20	19	17	17	19	20	24	7	5	4	24	1	1	1	0	24,20,19	ABC
N2	87	24	22	21	19	19	21	22	24	5	3	2	24	1	1	1	1	24,22,21,19	ABCD
N2	88	27	19	22	18	18	19	22	27	9	8	5	27	1	0	1	0	27,22	AC
N2	89	26	0	20	19	0	19	20	26	26	7	6	26	1	0	0	0	26	A
N2	90	28	0	22	24	0	22	24	28	28	6	4	28	1	0	0	1	28	AD
N2	91	19	18	18	17	17	18	18	19	2	1	1	19	1	1	1	1	19,18,18,17	ABCD
N2	92	20	0	19	0	0	0	19	20	20	20	1	20	1	0	1	0	20,19	AC
N2	93	19	19	20	21	19	19	20	21	2	2	1	21	1	1	1	1	19,19,20,21	ABCD
N2	94	26	19	18	18	18	18	19	26	8	8	7	26	1	0	0	0	26	A
N2	96	0	22	26	23	0	22	23	26	26	4	3	26	0	1	1	1	22,26,23	BCD
N2	98	25	25	23	0	0	23	25	25	25	2	0	25	1	1	1	0	25,25,23	ABC
N2	100	3	24	0	0	0	0	3	24	24	24	21	24	0	1	0	0	24	B
N3	64	24	0	0	0	0	0	0	24	24	24	24	24	1	0	0	0	24	A
N3	65	0	25	0	0	0	0	0	25	25	25	25	25	0	1	0	0	25	B
N3	66	0	25	3	0	0	0	3	25	25	25	22	25	0	1	0	0	25	B
IM3	94	15	0	10	8	0	8	10	15	15	7	5	15	1	0	1	0	15,10	AC
IM3	95	4	5	0	3	0	3	4	5	5	2	1	5	0	0	0	0	-	-
IM3	96	4	4	0	6	0	4	4	6	6	2	2	6	0	0	0	1	6	D
IM3	97	4	13	0	0	0	0	4	13	13	13	9	13	0	1	0	0	13	B
DN5	26	0	0	0	26	0	0	0	26	26	26	26	26	0	0	0	1	26	D

Table 5.2 Area to answer Conversion process

All probable conditions are shown in table 5.2. It there no selection of answer in answer sheet probability of marking answer for blank space will increase. In case of Q.No. 78 of B3 area are detected as 6, 7, 6 and 2 for A, B, C, D respectively, in reality this is detected due to a line, which is in blank sheet since the area will greater than 6 and difference is only 1 A,B,C all are marked as answer. It will effect on examination of negative marking system.

5.4 Processing Time (Second)

SN	Sheet	Edge Detection	Hough Transform	Binarization	Skew Correction	Answer Extraction	Result Calculation	All Processing
1	N1	0.4992032	0.2496016	0.2964019	0.2652017	0.5928038	0.0156001	2.0904134
2	N1	0.4212027	0.2340015	0.3276021	0.2496016	0.5928038	0.0312002	1.9500125
3	N2	0.5460035	0.156001	0.4368028	0.3432022	0.5616036	0.0156001	2.1372137
4	N3	0.6396041	0.1872012	0.2808018	0.2340015	0.5772037	0.0156001	2.0124129
5	IM1	0.4836031	0.2652017	0.2964019	0.2964019	0.5460035	0.0312002	1.9968128
6	IM2	0.624004	0.1404009	0.2964019	0.3588023	0.5772037	0.0156001	2.1060135
7	IM3	0.4524029	0.2652017	0.3276021	0.2652017	0.5460035	0.0156001	1.9500125
8	EM1	0.5304034	0.1404009	0.2808018	0.3432022	0.4992032	0.0312002	1.9032122
9	EM2	0.5460035	0.0936006	0.3276021	0.3588023	0.5928038	0.0312002	2.0124129
10	EM3	0.624004	0.1716011	0.312002	0.3588023	0.5148033	0.0156001	2.0904134
11	IR1	0.5460035	0.2496016	0.3432022	0.312002	0.5460035	0.0156001	2.1060135
12	IR2	0.6084039	0.156001	0.2808018	0.3900025	0.5304034	0.0156001	2.0592132
13	IR3	0.5772037	0.2496016	0.2808018	0.0780005	0.5460035	0.0156001	1.8252117
14	IR3	0.468003	0.2496016	0.2964019	0.0624004	0.5148033	0.0312002	1.7472112
15	IR1	0.5304034	0.2184014	0.3588023	0.2340015	0.6084039	0.0312002	2.1216136
Average		**0.5397635**	**0.201761293**	**0.316162**	**0.2766418**	**0.5564036**	**0.0218401**	**2.00721287**

Table 5.3 Processing time of each process

From the result of table 5.3 we can conclude that average processing time for each sheet is 2 second. 50% of time is loosed before Skew correction, so if we used efficient skew algorithm, processing time will reduced by 1 second. Most of the time is spend on Edge detection and Answer Extraction. If we use binary image in state of Edged image, time of Hough transform will slightly increase but all time of edge detection will reduced. Binary image will work as edge image so it will better to remove Edge detection process.

Table 5.4 shows the experiment of 50 randomly selected sheets. Table describes detection of reference point. If reference point is not detected than image is considered as invalid. Projection should be clear to detect number of rows and columns of answer block, if projection is not clear than sheets also considered as invalid image. Accuracy for detection of Roll No and Answer is calculated below by the help of table 5.4.

5.5 Result of Randomly selected Sheets

SN	Reference Detection	Projection	No of Question	No of Mark	Correctly Read	Correct Roll No	Read Roll No	Accuracy (%)	
								Answer	Roll No
1	Yes	Accept	100	100	100	2042	2042	100	100
2	Yes	Accept	100	100	100	2302	Error	100	-
3	Yes	Accept	100	100	99	2024	2024	99	100
4	Yes	Accept	100	100	100	2116	2116	100	100
5	Yes	Accept	100	100	100	2004	2004	100	100
6	Yes	Accept	100	98	98	2007	2007	100	100
7	Yes	Accept	100	100	100	2037	2037	100	100
8	Yes	Accept	100	100	100	2018	Error	100	-
9	Yes	Accept	100	101	101	2012	2012	100	100
10	Yes	Accept	100	86	86	2001	2001	100	100
11	Yes	Accept	100	86	86	2002	2002	100	100
12	Yes	Accept	100	100	100	2003	2003	100	100
13	Yes	Accept	100	100	100	2017	Error	100	-
14	Yes	Accept	100	100	100	2028	Error	100	-
15	Yes	Accept	100	102	102	2015	2015	100	100
16	Yes	Accept	100	101	101	2031	2031	100	100
17	Yes	Accept	10	10	10	2517	2517	100	100
18	Yes	Accept	150	149	149	2528	2528	100	100
19	Yes	Accept	180	178	178	2528	2528	100	100
20	Yes	Accept	20	20	20	2517	2517	100	100
21	Yes	Accept	210	207	207	2528	2528	100	100
22	Yes	Accept	30	30	30	2517	2517	100	100
23	Yes	Accept	40	40	40	2517	2517	100	100
24	Yes	Accept	50	50	50	2517	2517	100	100
25	Yes	Accept	60	60	60	2847	2847	100	100
26	Yes	Accept	80	75	75	2847	2847	100	100
27	Yes	Accept	100	99	99	3010	Error	100	-
28	Yes	Accept	100	101	101	3016	3019	100	75
29	Yes	Accept	100	86	86	3002	3002	100	100
30	Yes	Accept	100	95	93	3018	3018	97.9	100
31	Yes	Accept	100	100	100	3024	3024	100	100
32	Yes	Accept	100	98	98	3020	3020	100	100
33	Yes	Accept	100	97	97	3021	3021	100	100
34	Yes	Accept	100	101	101	3017	3017	100	100
35	Yes	Accept	100	100	100	3006	3006	100	100
36	Yes	Accept	100	100	100	3028	3028	100	100
37	Yes	Accept	100	100	100	3003	3003	100	100
38	Yes	Accept	100	100	100	3025	3025	100	100
39	Yes	Accept	100	100	100	3027	3027	100	100
40	Yes	Accept	100	99	99	3004	3004	100	100

41	Yes	Accept	100	102	102	3034	3034	100	100
42	Yes	Accept	100	101	101	2528	0018	100	25
43	Yes	Accept	100	99	99	2538	2538	100	100
44	Yes	Reject	100	-	-	2532	-	-	-
45	Yes	Accept	100	98	98	2540	2540	100	100
46	Yes	Accept	100	85	85	Blank	Error	100	-
47	Yes	Accept	100	93	93	2541	2541	100	100
48	Yes	Accept	100	99	99	2539	1529	100	50
49	Yes	Reject	100	-	-	2531	-	-	-
50	Yes	Accept	100	100	100	2030	2030	100	100
Total				4646	4644				

Table 5.4 Result of 50 randomly selected Sheets

Total number of samples are not accepted = 2

Total number of samples in which sheet was read = 50-2 = 48

For Answer,

Total No. of marks in 48 samples = 4646

Total No of mark detected = 4644

Number of marks not detected = 2

$$\text{Overall Accuracy for Answer X}\frac{\text{No. of marks correctly detected}}{\text{Total No. of marks detected}}\text{X}\frac{4644}{4646}\text{X99.99\%}$$

For Roll Number,

Total number Roll number = 48

Total number of Errors in Roll No = 6

Number of correctly read Roll no = 48 - 6 = 42

Number of Marks detected = 42*4 = 168

Number of marks incorrectly detected = 6

Total No of mark correctly detected = 168-6 = 162

$$\text{Overall Accuracy for Roll No X}\frac{\text{No. of marks correctly detected}}{\text{Total No. of marks detected}}\text{X}\frac{162}{168}\text{X96.45\%}$$

This Result shows that system is accurate for answer detection and it will improve the algorithm for answer detection. But it is 100% accurate if image will take correctly. Some time due to contrast of image, it will rejected at the point of Projection because number of row and column of answer area is calculated at that point. Another probable Rejection point is not detection the reference object (Square in system). If answer sheet is not printed darkly probability of not detection the reference will increased. Reference object work two main functions. One is Skew angle detection and another is separation of Roll No area and Answer area.

6 CONCLUSION

6.1 Conclusion

The process of evaluation of exam papers requires a very high degree of accuracy which may not be possible during manual evaluation as a human being tends to get fatigued due to monotonous nature of the job. It also takes too much time and concentration. To overcome this problem, many efforts have been made by the researchers across the globe for last many years. A similar effort has been made in this work to develop an accurate and automatic examination evaluation. It has been tested for 50 sheets containing questions from 10 to 210 with different position and abstraction around it. In the process of final evaluation, we get an accuracy of 100% for correctly scanned sheets and 99.99% for overall system. Roll No recognition by the system is only 96.45% of accuracy. It is concluded that the research has been by and far successful.

From practical experience it takes minimum 5 minutes to check one sheet of 100 Questions, and minimum of 2 minutes to calculate result. Development of this system with our algorithm time will reduce 8 minutes to 1 second. From Result of the experiment, we can conclude that average processing time for each sheet is 2 seconds. Fifty percent of time is lost before Skew correction, so if we use efficient skew algorithm, processing time will be reduced by 1 second. Most of the time is spent on Edge detection and Answer Extraction. If we use binary image in state of Edged image, time of Hough transform will slightly increase but all time of edge detection will be reduced. Binary image will work as edge image so it will better to remove Edge detection process.

6.2 Future Scope

In case of our country most of the system was going manually. There are few institutions they are using such system for exam evaluation. In case of government, there we can found really use of OMR technology for exam evaluation. Mostly objective exams are taken for Entrance examination. According to my best knowledge such application is using by Institute of Engineering since more than last 10 years, but Institute of Medicine is using such system only from 2010 and Council for Technical Education and Vocational Training (CTEVT) is using it from 2011.Public Service Commission also starting to use this system from this year. It shows that the use of OMR technology is increasing year by year. Until now most of institution and college are used manual system for result publishing which is mostly due to lack of knowledge on such system and their ability to afford such system. The current system is cost efficient and easy to use which shows the system has a bright future scope.

6.3 Further Study

OMR technology is only basic step, if we do not considered the accuracy of system it will become very simple, now it will turn to develop such system which can also recognize Mark, Number, Later, tables and all things that are in a single image. At first it will necessary to extract the place where those are located in image, for that we can used pattern recognition, projection profile and different binarization algorithms. In our country traffic system is necessary to control by the system, for that it is necessary to recognize the number plate automatically. The system of design of number plate in our country is different than other country so, it should necessary to study and developed number plate system for existing plates. For that it will necessary to study on, how to extract number plate position? If position of number plate will find out, this study help to extract each character from plate. After that it will need to study how to recognize the character?

This study also can be extended to the study of,

1. How to convert hard copy to soft copy?
2. How data can extract from survey form?
3. How to extract tables from image?
4. Character recognition
5. Signature Recognition
6. Thumbprint recognition
7. Other many more…….

7 BIBLIOGRAPHY

[1] C. Roger, Palmer, “The Basics of Automatic Identification [Electronic version]”, Canadian Datasystems, 21 (9), p. 30-33, Sept 1989.

[2] F. Herbert, Schantz, “The History of OCR, Optical Character Recognition”, Manchester Center, VT: Recognition Technologies Users Association, 1982.

[3] K. Chinnasarn Y. Rangsanseri “An image-processing oriented optical mark reader” Applications of digital image processing XXII, Denver CO, 1999.

[4] H. Stephen, C. Leona, C. Fung, M. Albrecht, “Low Cost and high speed Optical mark reader based on Intelligent line Camera”, Proceedings of the SPIE Aero Sense 2003, optical pattern recognition XIV, Orlando, Florida, USA, vol. 5106, 2003. p. 200–08.

[5] H. Stephen, W. Peter Deng, “A high speed optical mark reader hardware implementation at low cost using programmable logic”, Science Direct, Real-Time imaging, Volume 11, Issue 1, 2005. 77.

[6] H. Deng, F. Wang, B. Liang, “A low-cost OMR solution for educational applications” Parallel and Distributed processing with Applications 2008, ISPA ’08, International Symposiun, Dec 2008.

[7] J. Sauvola, M. Pietika, A. Kinen, “Page segmentation and classification using fast feature extraction and connectivity analysis”, International Conference on Document Analysis and Recognition, ICDAR '95, Montreal, Canada, 1995, pp. 1127-1131.

[8] H. Baird, “Document image defect models, Proceedings of the IAPR Workshop on Syntactic and Structural Pattern Recognition”, 1990, pp. 38-46.

[9] J. Sauvola, M. Pietika, “Adaptive document image binarization” Machine Vision and Media Processing Group, Infotech Oulu, University of Oulu, 1999.

[10] N. Otsu, “A threshold selection method from grey level histogram”, IEEE Trans. Syst. Man Cybern., vol. 9 no. 1,pp. 62-66 (1979).

[11] W. Niblack, “An Introduction to Digital Image Processing”, Prentice Hall, Englewood Cliffs, (1986).

[12] J. Sauvola, T. Seppanen, S. Haapakoski, M. Pietikainen, “Adaptive Document Binarization”, 4th Int. conf. on document analysis and recognition, Ulm, Germany, pp.147-152 (1997).

[13] C. Wolf, J-M. Jolion, “Extraction and Recognition of Artificial Text in Multimedia Documents”, Pattern Analysis and Applications, 6(4):309-326, (2003).

[14] M-L. Feng, Y-P. Tan, “Contrast adaptive binarization of low quality document images”, IEICE Electron. Express, Vol. 1, No. 16, pp.501-506, (2004).

[15] C. Galambos, J. Kittler, and J. Matas, “Progressive probabilistic hough transform for line detection”, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 1:1554, 1999.

[16] A. Mahmoud, Al. Shatnawi, K. Omar, “Skew Detec-tion and Correction Technique for Arabic Document Images Based on Centre of Gravity”, published in Journal of Computer Science 5 (5):363-368, 2009.

[17] W. Steven Smith, “Digital Signal Processing: A Practical Guide for Engineers and Scientists”, 2003.

[18] Z. Huiyu, Wu. Jiahua, Z. Jianguo, “Digital Image Processing”, ISBN 978-87-7681-541-7, 2010.

[19] R. O. Duda, P. E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” Comm. ACM, Vol. 15, pp. 11–15 (January, 1972)

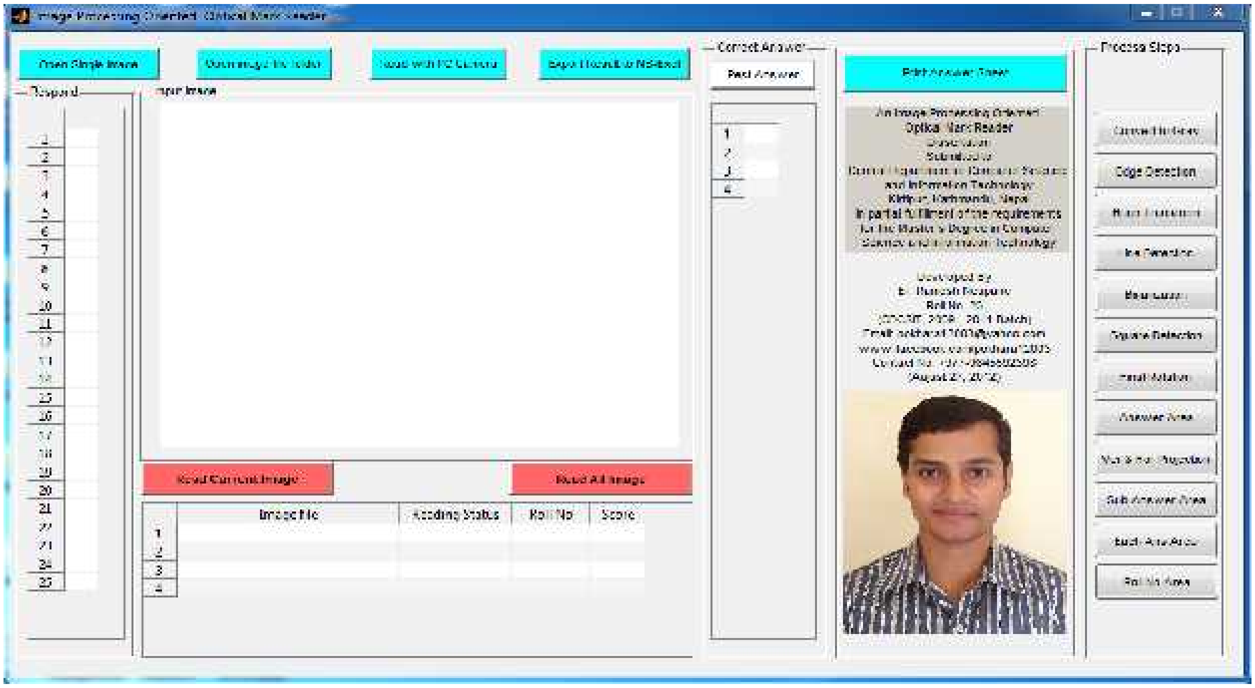
[20] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981.

[21] W. Zhi, H. Sai-xian, “An Adaptive Edge-detection Method Based on Canny Algorithm”, Measuring & Testing Technology and Communication Engineering Ministerial Key Laboratory, School of Electronics Information, Wuhan University, Wuhan 430079.

[22] M. Aradhya V N, H. Kumar, P. Shivakumara, “Skew Detection Technique for Binary Document Images based on Hough Transform”, published in International Journal of Information Technology, 2007.

8 APPENDIX

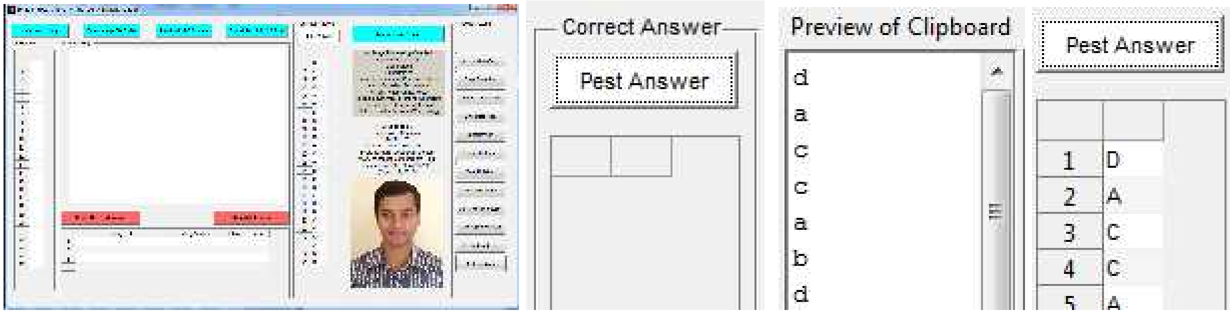
A.1 User Interface



An Image Processing Oriented
Optical Mark Reader
Dissertation
Submitted to
Central Department of Computer Science
and Information Technology
Kirtipur, Kathmandu, Nepal
In partial fulfillment of the requirements
for the Master's Degree in Computer
Science and Information Technology

Developed by
[illegible]
Roll No. [illegible]
(CDCSIT, 2068 - 2011 Batch)
Email: [illegible]@yahoo.com
www.facebook.com/[illegible]
Contact No.: +977-[illegible]
(August 21, 2012)

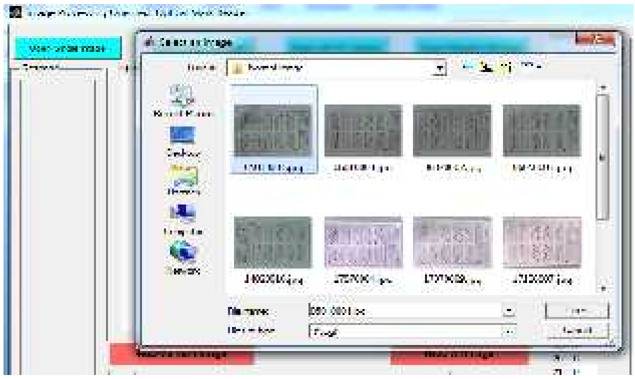
Figure 6.1 Blank User Interface Design



d
a
c
c
a
b
d

1	D
2	A
3	C
4	C
5	A

Figure 6.2 Answer Import from Clipbord



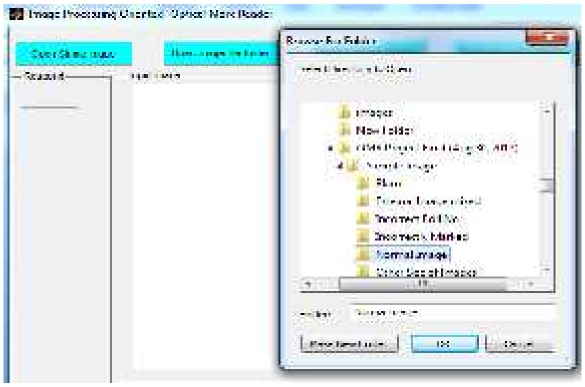
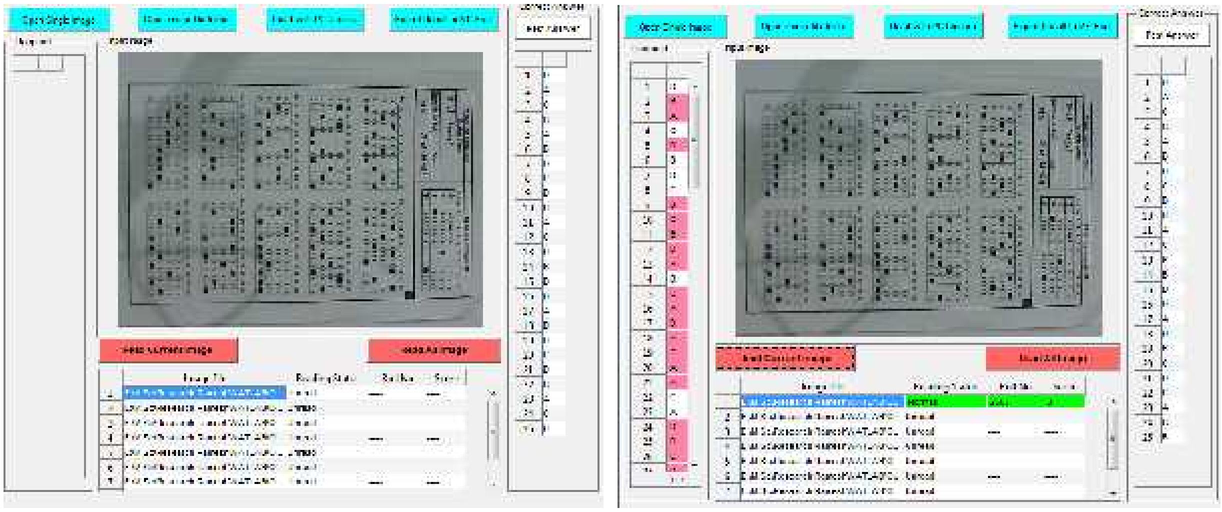
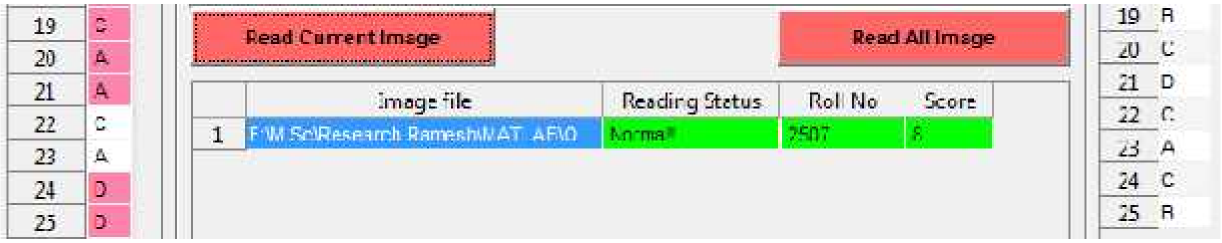


Figure 6.3 Open Image file or Image folder



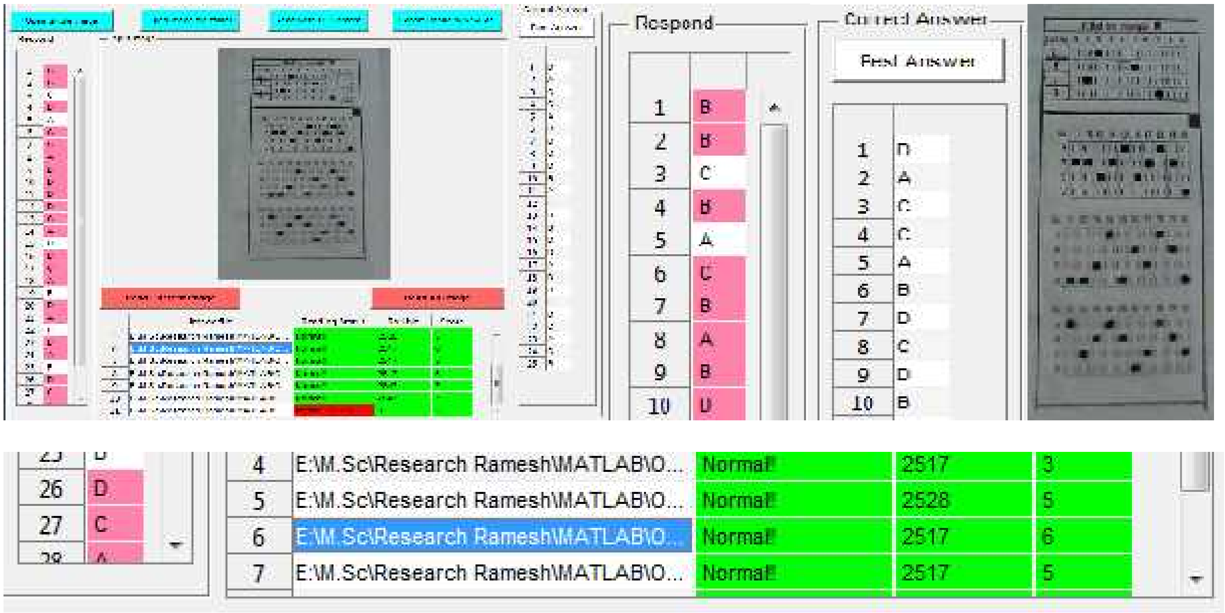


19	C
20	A
21	A
22	C
23	A
24	D
25	D

	Image file	Reading Status	Roll No	Score
1	E:\M.Sc\Research Ramesh\MATLAB\O	Normal	2507	8

19	B
20	C
21	D
22	C
23	A
24	C
25	B

Figure 6.4 Reading Current Image



Respond

1	B
2	B
3	C
4	B
5	A
6	C
7	B
8	A
9	B
10	D

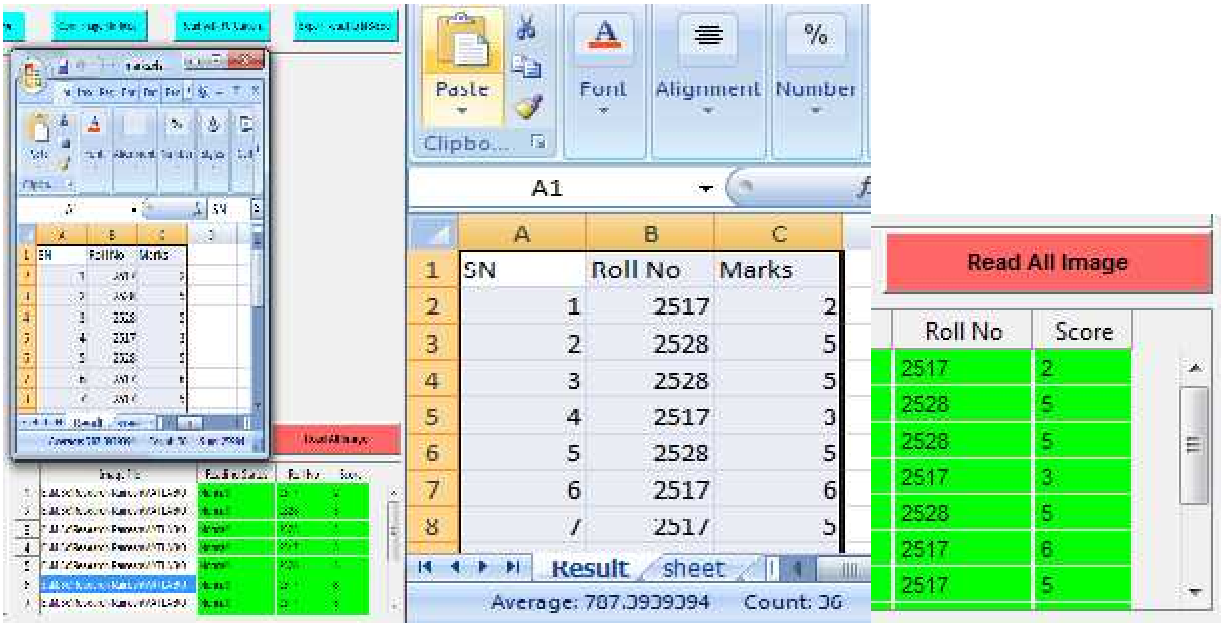
Correct Answer

1	D
2	A
3	C
4	C
5	A
6	B
7	D
8	C
9	D
10	B

26	D
27	C

4	E:\M.Sc\Research Ramesh\MATLAB\O...	Normal	2517	3
5	E:\M.Sc\Research Ramesh\MATLAB\O...	Normal	2528	5
6	E:\M.Sc\Research Ramesh\MATLAB\O...	Normal	2517	6
7	E:\M.Sc\Research Ramesh\MATLAB\O...	Normal	2517	5

Figure 6.5 Read All Image and Show Result



	A	B	C
1	SN	Roll No	Marks
2	1	2517	2
3	2	2528	5
4	3	2528	5
5	4	2517	3
6	5	2528	5
7	6	2517	6
8	7	2517	5

Average: 787.3939394 Count: 36

Roll No	Score
2517	2
2528	5
2528	5
2517	3
2528	5
2517	6
2517	5

Figure 6.6 Export Result to Ms-Excel

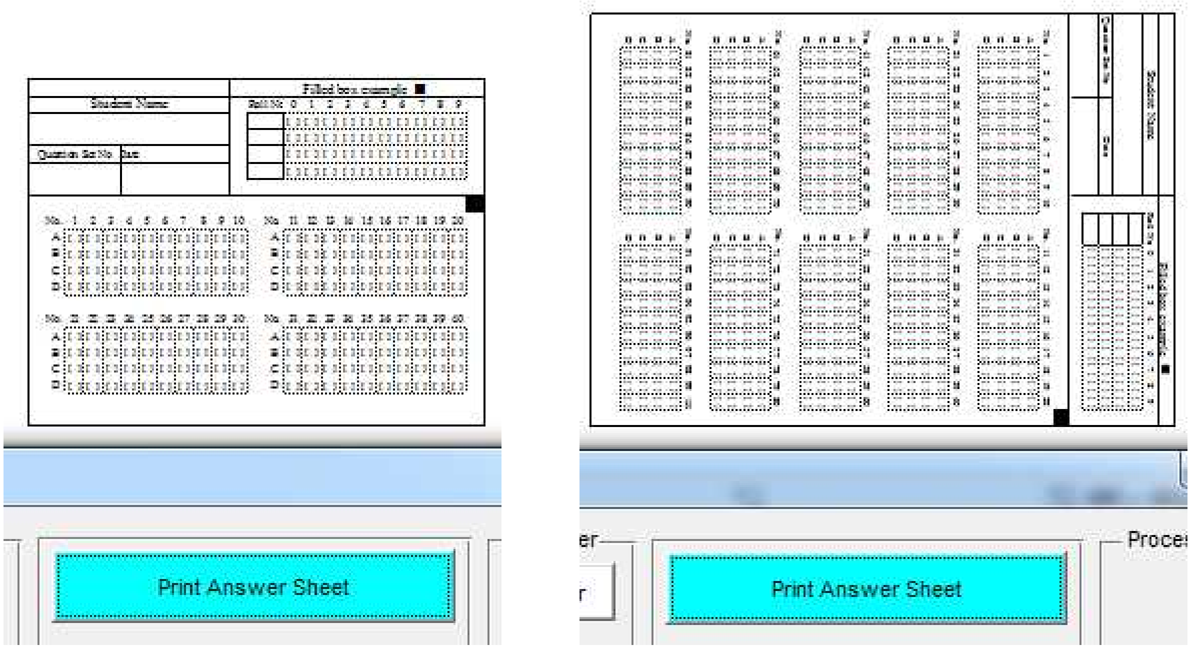


Figure 6.7 Print Answer Sheet

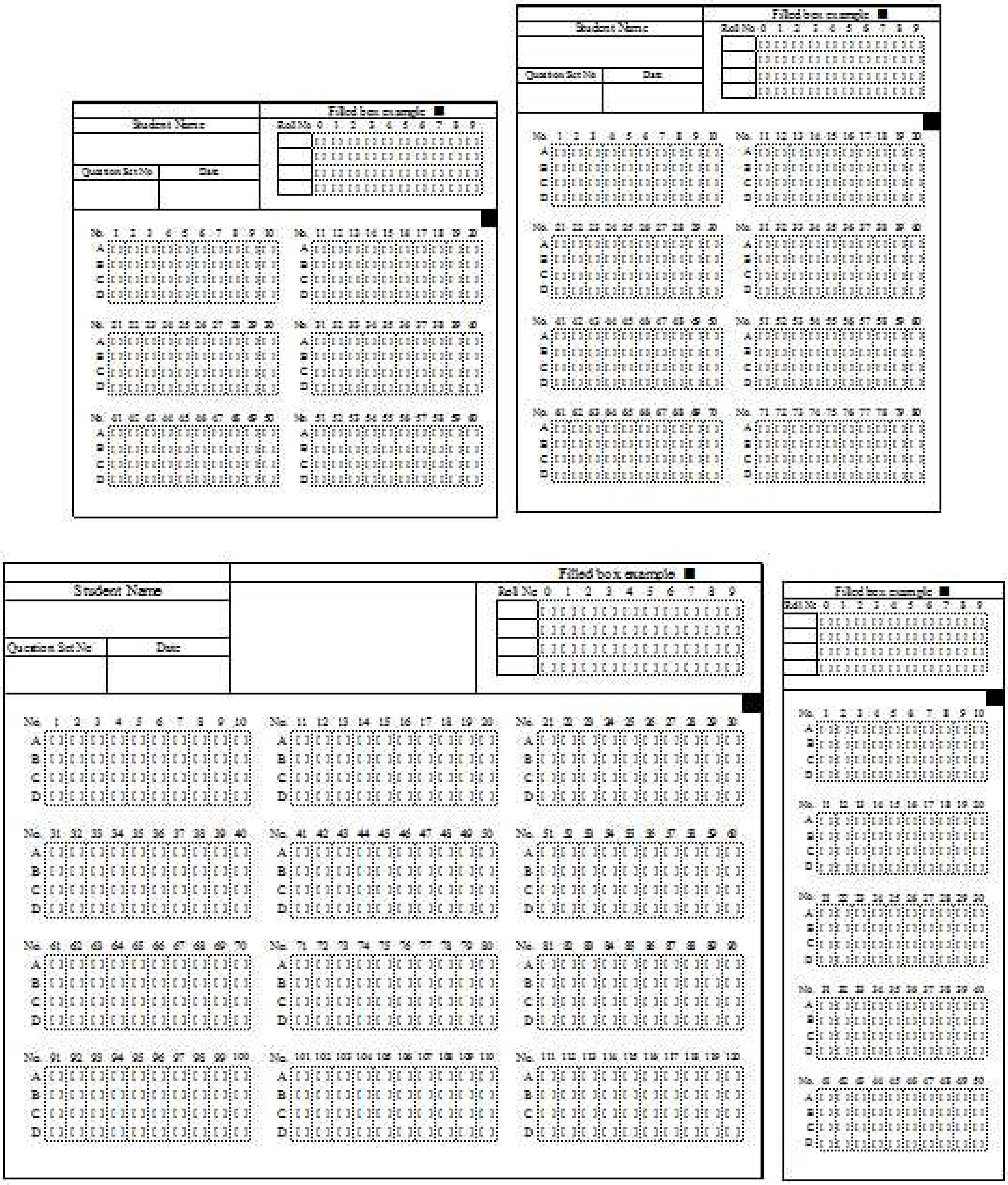


Figure 6.8 Sample of Answer Sheets with different number of questions