



Masters Project

Final Report

(MCS)

2019

Project Title	A digitally developed methodology to address the pre-processing phase of capturing data from manually filled feedback forms
Student Name	J. M. M. S Kularathna
Registration No. & Index No.	2013mcs082(13440821)
Supervisor's Name	Prof. KP Hewagamage

For Office Use ONLY



**A digitally developed
methodology to address the pre-
processing phase of capturing data
from manually filled feedback forms**

**A dissertation submitted for the Degree of Master of
Computer Science**

**J. M. M. S. Kularathna
University of Colombo School of Computing
2019**



**A DIGITALLY DEVELOPED METHODOLOGY TO
ADDRESS THE PRE-PROCESSING PHASE OF
CAPTURING DATA FROM MANUALLY FILLED
FEEDBACK FORMS**

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: JMMS Kularathna

Registration Number: 2013mcs082

Index Number: 13440821

Signature:

Date:

This is to certify that this thesis is based on the work of

Ms. JMMS kularathna

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by: Prof. KP Hewagamage

Supervisor Name: Prof. KP Hewagamage

Signature:

Date:

ABSTRACT

The main objective of my thesis is to generate a new fine tuned Optical Mark Recognition(OMR) which minimize the error count while processing. There are several OMR tools which are used for manual data processing. But, those are not well fine tuned. And there should be a methodology to find out the errors which returns while processing and the reason for the error out come. The newly developed system is consider the option buttons, checkbox and text area as input fields for the questionnaire.

Most of the existing OMR systems working under the same set of rules, so that the intention of this system implementation is to provide a system which can dynamically generated forms with varying set of rules. And the next intention is to increase the reading ability of manually filled questionnaires, since the existing systems rejects the filled forms due to several reasons. So the new software provides a more data extraction percentage that existing tools. Then the new system provides the ability to identify the error occuring scenarios while reading the filled forms. The system marks the respective question with red colored box in order to identify the error for the system user.

The implemented system is independent of dedicated scanners, so that the system provides the capability of reducing the cost for the full cycle of taking feedback.

With regards to the user interface users can create , read and process dynamic forms. And while processing it stores the data in a database which is stored in the computer.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor for his immense support in providing relevant knowledge, advice, supervision and useful suggestions throughout this research. His guidance helped me to complete my work successfully.

I would like to thank course coordinator for the support, advice and encouragement in continuing this research till the end. Further I would like to thank all my colleagues for their help in finding relevant research material and for their encouragement.

Finally I would like to appreciate the support and encouragement given by my family in completing this project successfully.

TABLE OF CONTENT

Declaration of the candidate & Supervisor

Abstract

Acknowledgement

Table of Content

List of Figures

List of Tables

List of Abbreviations

1 Introduction

1.1 Background

1.2 Problem

1.3 Objectives

1.4 Scope

2 Literature Review

2.1 Background

3 Problem Analysis

4. Model /Design

5 Implementation

6 Analysis and Results

7 Conclusion

7.1 Summary

7.2 Limitations

7.3 Future Work

7.4 Contribution

References

CHAPTER 1

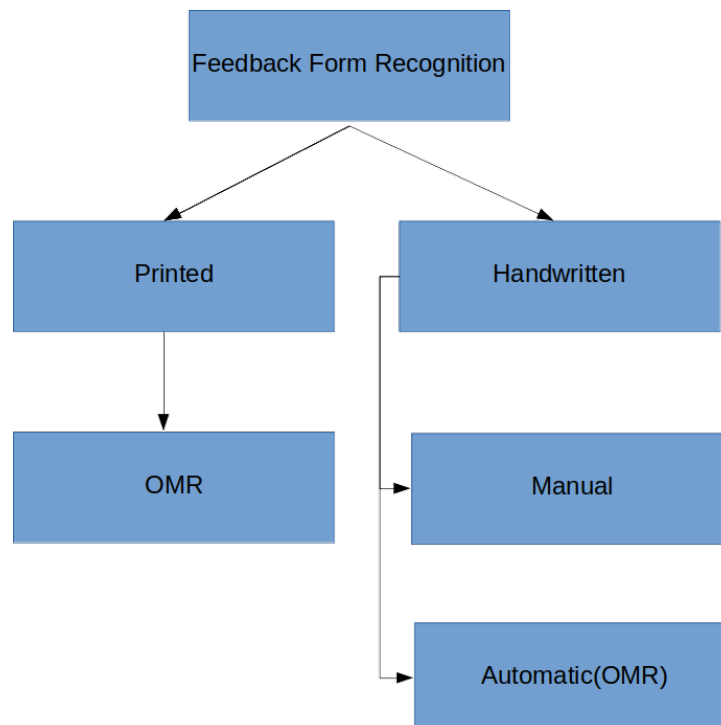
INTRODUCTION

Chapter 01: Introduction

1.1 Background

It is a great chance of having feedback in any kind of event, which is a methodology of measuring the value of given scenario. So, Feedback forms are highly utilized in many organizations thus to induce a concept of their performance, usage and therefore the level of satisfaction of the service provided. With the rise of the number of occurrences of those feedback forms, there ought to be an exact and effective manner of doing each operation relating to generate forms, print, read, retrieve knowledge and make statistical reports. These feedbacks are really powerful force in driving any organization with modifications. From the service utilizers' perspective, they need to have an easy way of expressing their views. Also, the administrators usually apply an appropriate assessment method to gain the maximum knowledge from the survey. So, there are different ways of operating a survey to make a measurement of a system capabilities in different contexts.

Mainly the operating a survey divided into two main categories such as online and offline. Then the offline method is also having multiple ways of doing, but paper based surveys are the most common method exists in Sri Lankan institutes.



The problem comes into the picture from the phase of retrieving the knowledge using manually filled surveys. So, there should be precise method of capturing data in a short period of time and the automated system which developed with the combination of Optical Mark Recognition (OMR) and Optical Character Recognition (OCR) supports the process of retrieving knowledge. Developing world people tend to move on automated processes than manual works. And the people don't want to invest their time on processing these data. Currently there are few free tools available for doing these operations. Most of the Sri Lankan organizations are victims of written version of feedback forms and fill out these by hand writing. Therefore many issues are arising throughout next level of method that is scan the written stuff and retrieve knowledge. The main problem occurs on way to capture correct data from scanned files.

For an example, people won't stick into the margin of text box once manually fill these forms. So that, an automated tool will capture the information that is within the text box. But it will not capture the words which are outside of the given text area. These missing words can cause incorrect analytic results. Therefore there should be a way of handling this kind of exceptional cases to induce the proper data. There may be people who fill bubble in halfway and most of the OMRs will not capture that as a data and returns an error. But there is no way of identifying the reason and the section which leads to an error while reading the form. Since these tools are not trained with Artificial Intelligence (AI) there should be a method of view the error by marking the section and ignore that if the error is acceptable for processing. With regards to the metadata of scanned image, some OMRs supports just for one form of image format. That will not be a good limitation since there can be several formats as per the requirement of the client. And also most of the OMR applications need high quality images for retrieve data properly. But for being a tool this should facilitate at least medium quality images also should facilitate the system.

The proposed system is designed in such a manner that it can preprocess the data inputs. For example, the existing systems process the data but they reject the processing once they stuck in a place while retrieving. So the intention is to develop a system which can view the error occurring section and the user can look into that. Then if the system reported error is human acceptable error, the user can ignore it and give instructions to the system as to proceed. Since a new OMR tool is developed within the project, the error occurring scenarios will decrease more than 50% with compare to existing tools. And the user need not to have any special training to operate the system. And the proposed system can be implemented by using camera instead of scanner.

1.2 Problem

Feedback forms are highly utilized in many organizations thus on induce a concept of their performance ,usage and therefore the level of satisfaction of the service provided. With the rise of the number of occurrences of those feedback forms, there ought to be an exact and effective manner of doing each operation relating to generate forms,print, read, retrieve knowledge and make statistical reports. But in the developing world people tend to move on automated processes than manual works. And the people don't want to invest their time on processing these data. Currently there are few free tools available for doing these operations. Most of the Sri Lankan organizations are victimization written version of feedback forms and fill out these by hand writing. Therefore many issues are arising throughout next level of method that is scan the written stuff and retrieve knowledge. Initial one is the way to capture correct data from scanned files.

For an example, people won't stick into the margin of text box once manually fill these forms. So that, a tool developed with the combination of Optical Mark Recognition(OMR) and Optical Character Recognition(OCR) will capture the information that is within the text box. So that OCR will not capture the words which are outside of the given text area. These missing words can cause incorrect analytic results. Therefore there should be a way of handling this kind of exceptional cases to induce the proper data. There may be people who fill bubble in halfway and most of the OMRs will not capture that as a data and returns an error. But there is no way of identifying the reason and the section which leads to an error while reading the form. Since these tools are not trained with Artificial Intelligence(AI) there should be a method of view the error by marking the section and ignore that if the error is acceptable for processing.

With regards to the metadata of scanned image, some OMRs supports just for one form of image format. That will not be a good limitation since there can be several formats as per the requirement of the client.And also most of the OMR applications need high quality images for retrieve data properly. But related to the image scanner at least medium quality images also should facilitate the system. So, the above mentioned issues will be addressed during the development phase of this research

1.3 Objectives

- 1) Provide an online OMR tool which can pre-process data and generate statistical reports precisely at any given time without looking forward for specifically installed software.
- 2) Decrease minimum 50% of error occurring scenarios while processing data.

1.4 Scope

- a. Identify and specify the marked answers more precisely than existing free software applications.
- b. Compare and validate with Scripts for data acquisition with paper-based surveys(SDAPS)
- c. Identify , mark and display the areas which leads to erroneous results while processing.
- d. Pre-process answer sheets with the aid of previously identified erroneous areas before generating statistics.
- e. Provide a dynamic form reader instead of reading static forms
- f. Provide the ability to process multiple image file formats without limiting to one format.
- g. Provide a system which is independent of color, brightness and illumination
- h. Capture data which went outside of margin and get the correct information as human see the information.

Ex : Textbox , Checkbox , Radio Button

CHAPTER 2

LITERATURE REVIEW

Chapter 02 : Related Work

There exists an overwhelming volume of paper-based data in most of the government organizations in Sri Lanka. The work force for managing and generating statistics by using that data is a huge problem in the present. So the developing countries use digitally developed systems for working faster and more efficiently than human operators. One of the major ways of spending time on paper based document is extracting data from manually filled feedback forms. So, paper-based survey forms can be read by using these systems so that it can perform efficient and accurate content management. There exist several Optical Mark Recognition(OMR) systems which provides tools to design the layout as per the requirement of the user. The users should be able to take the responses as much as required by distributing printouts and get the filled sheets scanned.

So then the scanned images will be used as the inputs to the software for processing. Optical character recognition(OCR) system allows us to extract the text from an image formatted document. These systems can effectively recognize hand-printed or machine printed forms. But most of the systems desired to have neatly entered data in a spaced boxes with acceptable level of space between letters. So without the use of these conventional technologies above systems reject field reading if the people unable to follow the structure filling out the forms.

Most of the OMR products need to have a dedicated scanner device to work with. But it is nice to have a system which can work without dedicated hardware, so that the system can independently used by any community. For that, it can be used simple scanned images of manually filled feedback forms apart from using dedicated scanner. So that there should be a methodology to read these images instead of using dedicated scanner with light beams.

One of the first OMR software packages that used images from common image scanners was Remark Office OMR, made by Gravic, Inc. (originally named Principia Products, Inc.). Remark Office OMR 1.0 was released in 1991. The need for OMR software originated because early optical mark recognition systems used dedicated scanners and special pre-printed forms with drop-out colors and registration marks. Such forms typically cost US\$0.10 to \$0.19 a page. In contrast, OMR software users design their own mark-sense forms with a word processor or built-in form editor, print them locally on a printer, and can save thousands of dollars 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 (bi-tonal) 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 gray-scale to better identify the intent of the marker-internally scantron and NCS scanners used gray-scale. OMR software is also used for adding OMR marks to mail documents so they can be scanned by folder inserter equipment. An example of OMR software is Mail Markup from UK developer Funasset Limited. This software allows the user to configure and select an OMR sequence then apply the OMR marks to mail documents prior to printing. Most of the free feedback management tools support for atomic operations. There is a necessity of taking the support from another tool for the next operation. But there are many other online surveying tools which are not for free such as Client Heartbeat, SurveyGizmo and SurveyMonkey and so on. Basically, for nonprofit organizations cannot afford money on that. And all the operations have done online. But for the countries like Sri Lanka, most of the time people use handwritten feed backs to the organizations. So there should be a way of managing manual operations precisely. Systems like 'SDAPS' provide there service freely. But this system is a standalone application and its installation is a bit harder, since its having some dependencies. So implementing an online tool will be easier to users even-though they haven't knowledge on installing a software properly. And the other thing is SDAPS also have some problems like, it can only read 'gif' pictures only. So, we need to support other commonly used picture formats like 'jpg', and 'png'. And also SDAPS can read only specific range of RGB values only. So we need to increase the supportive range.

The first mark sense scanner was IBM 805 Test Scoring Machine; that read marks by sensing the electrical conductivity of graphite pencil lead using pairs of wire brushes that scanned the page [7].Chinnasarn et al [2] present a system based on PC-type microcomputer connecting to an image scanner. The system operations can be distinguished into 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.Pegasus Imaging Corporation [3] presented a Software Development Kit for OMR recognition from document images. The SDK supported template recognition mode and free recognition mode. An OMR field is defined as a rectangle area containing a specified number of columns and rows of bubbles to be evaluated. The SDK can scan the region horizontally and then vertically to locate the bubbles apart from the spaces between them. Then, based on the bubble shape specified, it scans the discrete areas of the bubbles,counting dark pixels to determine which bubble areas qualify as "filled in".

Hussmann S. et al [4] 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. However, the resolution and overall system design was not satisfying and lead to further investigation.

Hussmann S. et al [5] describes 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.

Hui Deng [6] also presented an image based approach. In this approach, the questionnaire sheet is having two types of marks: solid marks and hollow marks. The solid mark is for system usage, defined as system mark. The hollow mark is to be filled by students for information recognition, so is defined as information mark. Moreover, solid marks are composed by two types i.e. circular-shaped and rectangular-shaped. The circular-shaped solid marks, which locate in top left and top right of the sheet, are used to correct the tilt of the whole page. The rectangular-shaped solid marks are defined as “flag points” in this paper, are used to search the coordinates of information marks.

The problem with the Pegasus technique is that in the school, the multi-choice answer recognition success rate cannot achieve the requirements of the examination and with the Deng’s image based technique is, to get 100% precise recognition, more manual work need to be performed.

The Proposed technique by Gupta[8] is a low cost solution of OMR process. It neither requires high cost computational machine (reader) for scanning nor expensive high quality paper. Also this is an image based technique which can be used in small scale industries, institutes and schools.

There are four basic steps in this proposed method:

- (i) Template Designing.
- (ii) Image Capturing.
- (iii) Performing 2D transformation and scaling on scanned image to align and size it correctly.
- (iv) Finding marks on questionnaire.

The proposed architecture [8] includes the use of PHP (version 5.0), MySQL for database storage, JavaScript for uploading scans, Adobe Dreamweaver CS2 for designing the interface. For image processing mainly two steps are followed i.e. checking inclination and then checking scaling. After alignment(rotation and scaling) of the questionnaire image, mark recognition is to be done and the results are generated(Figure 1).

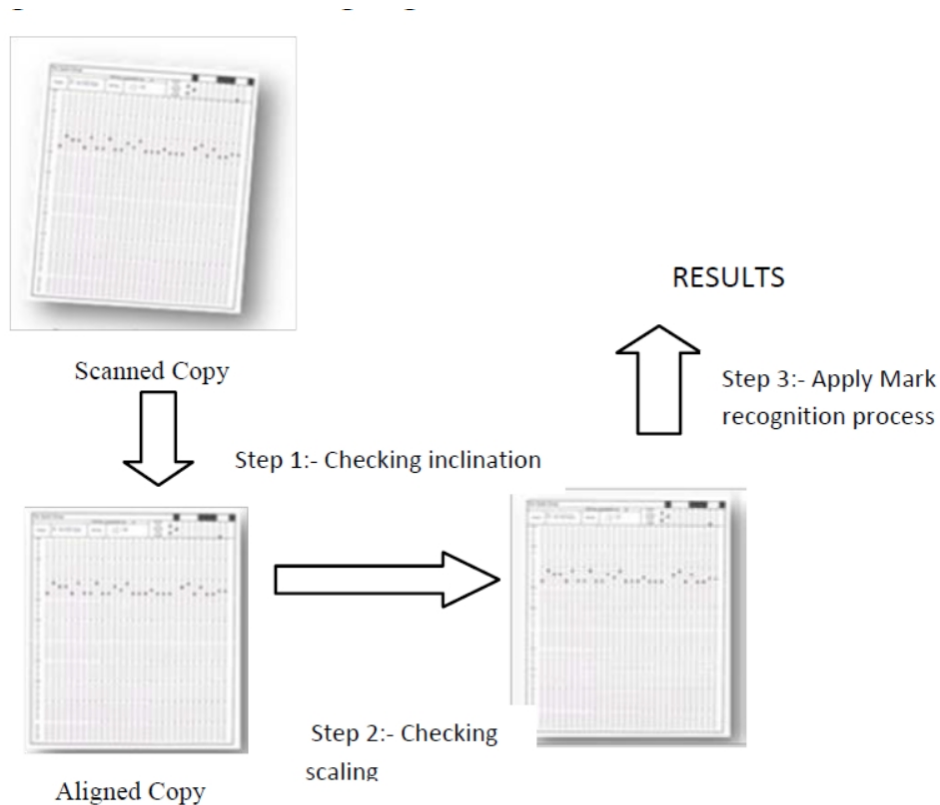


Figure 01

The proposed method by Rakesh et al proposed method that mainly comes under 3 steps. They are Design of Form ,Registration of Forms, Form Evaluation. [9]

When the filled forms are scanned, the position of bubbles corresponding to each question is not identical across all forms. [9] The variance in translation and rotation of position of

corresponding bubbles in different forms is attributed to manual error in the alignment of the form during the process of scanning. Thus all scanned images must be registered to a fixed position before further processing, so that the corresponding position of bubbles in all scanned images is same. This is illustrated in Figure 2.

Registration is done by detecting the square boxes located in the corners of the form. The angle α formed by the line segment joining the end points of two consecutive squares in clockwise sense is calculated using simple trigonometry ($\tan^{-1}(ay/ax)$) as shown in a hypothetical sketch. Similarly β , γ and δ are calculated as mentioned in Figure 2. The image is rotated by the average of α , β , γ and δ in anti-clockwise sense about the center of the smallest rectangle bounding the four squares.[9]

The exact coordinates of these squares are determined, and a suitable transformation matrix is used to translate all the images such that the position of bubbles corresponding to each question in all images is same. [9]

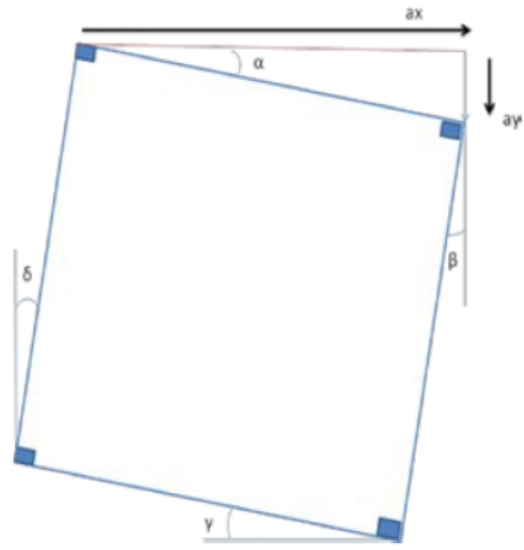
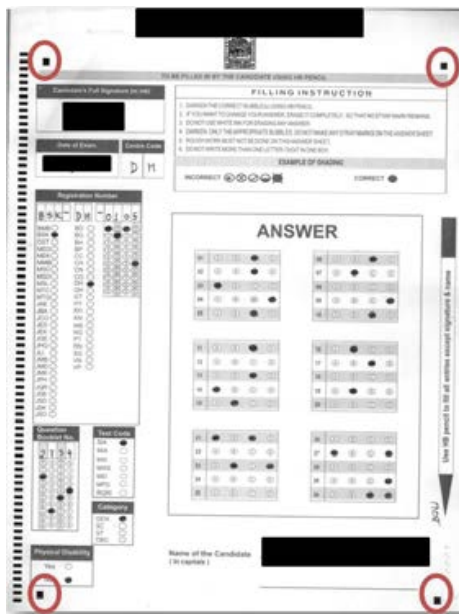


Figure 02

When it comes to form evaluation in this approach [9], after registration of the answer sheets, we detect the rectangular contours of main answer box and sub-answer boxes in the OMR sheet. Relative to these contours, the position of all the bubbles in the form is fixed. The grayscale value of a bubble is defined as the average grayscale value of all the pixels in the smallest rectangular region that completely bounds the bubble. Experimental results show that the mean grayscale value of pixels corresponding to filled-in bubbles is relatively much lower than the unfilled ones, which is clearly depicted. The average grayscale value of the smallest rectangular

region that bounds the bubble completely is much lower for a filled bubble compared to that of an unfilled bubble.

CHAPTER 3

PROBLEM ANALYSIS

Chapter 03 : Problem Analysis

Forms recognition and processing is used all over the world to tackle a wide variety of tasks including classification, document archival, optical character recognition and optical mark recognition. Out of those general categories, OMR is an oft misunderstood and underused feature in document imaging due to the time required to set up OMR based forms and the difficulty of accurately detecting which OMR fields are filled on a scanned document. Creating and processing OMR forms can be a time-consuming nightmare and this white paper will discuss how to alleviate those issues through automated detection, classification and processing.

Most forms contain a small number of OMR fields to capture information such as gender and marital status. These cause little to no difficulties because there are very few fields to deal with. On the other hand, creating and processing forms dominated by multiple choice questions is noticeably more difficult due to the sheer volume of fields that can be packed into a page. Additionally, the small size of check boxes, bubbles and other types of OMR fields creates potential hypersensitivity resulting in more false negatives or positives. This research will examine in more detail how to alleviate both of these common problems by developing an OMR forms recognition application.

The first step in a forms recognition application is to build the master forms. These master forms, or blank form templates, serve two primary purposes. First, it is used to identify what type of form a scanned document is. Second, the fields indicate the areas on the form from which data will be recognized and extracted.

For many systems, creating an OMR based form can be a tedious process due to the amount of repetition involved with surveys, bubble sheets or tests. One could spend hours manually drawing each and every OMR field around the boxes. After finding each zone on the page, you can loop through the collection and add a new OMR field for each OMR zone.

Any organization that collects data on paper-based forms, surveys or applications on a regular basis can get a very high return on investment by automating the data entry with forms processing software. They need to have a significant number of forms to justify the expense at least a hundred forms per month or more depending on how much data is being captured.

Organizations that have many separate departments that collect data on forms can share the budget for forms processing software by re-using it for other projects.

There are several challenges in character recognition. Machine-printed text includes materials such as books, newspapers, magazines, documents and various writing units in the video or still images. The problem of recognition of fixed-font, multifold and omni-font character is relatively

well understood and solved with some constraints. Documents generated on a high quality paper with modern printing technologies allow the systems to exceed 99% recognition accuracy . However, the recognition rate of the commercially available products depends on the age of the documents, the quality of the paper and ink, which may result in significant data acquisition noise. Documents with colored or patterned backgrounds, marked with pens, crooked when scanned, can yield poor OCR results. Some improvement can be done by either adjusting the scanner settings and rescanning the document or manually correcting the electronic data .

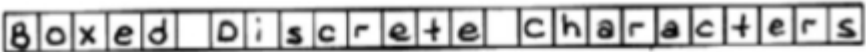
1. 
2. Spaced Discrete Characters
3. Run-on Discretely written Characters
4. Pure Cursive Script Writing
5. Mixed Cursive and Discrete

Figure 03

Hand-written character recognition systems still have limited capabilities even for recognition of the Latin characters and digits. The problem can be divided into two categories: cursive and hand-printed script. Five levels of difficulty in handwritten recognition have been defined in the literature [11].

CHAPTER 4

Model/Design

Chapter 04 : Model/Design

OCR technology replaces manual rewriting of printed documents to electronic form. The OCR application can recognize not only printed text, but also font type and size, paragraph formatting, tables, graphic elements like charts, diagrams and images. A single A4 page recognition time can be about 1 minute, depending on hardware and software configuration. One of specific uses of OCR/ICR technology is recognition of forms, documents which have certain structure or specific layout. Form recognition differs from full-text recognition as it focuses on retrieving particular, predefined pieces of information from a document . OCR/ICR application can convert structured and semi-structured documents such as invoices, payment drafts, bills, etc. During recognition process it locates all the text in the image, including characters and numbers - even if this information is located in stamps, pictures, logos or small text areas. Available ICR technology allows to recognize handwritten documents, but it is limited to difficulty level 1 and 2 (Figure 03). It means that only separate, capital letters or digits can be recognized. Some OCR/ICR engines provide information about recognition reliability and if the character is not marked as certain, the system gives a list of alternatives. This functionality is useful during dictionary verification and it allows to reduce the number of substitutions, which may cause additional errors .

Some OCR/ICR engines have a voting mechanism. Each character is recognized by at least two parallel algorithms and the voting mechanism decides about recognition result [11]. Some OCR/ICR engines can process different types of forms in one batch. They identify the type of the form and in the same process they use predefined templates to recognize proper blocks .

There are several OCR/ICR processing steps. First step is usually page preprocessing. During this process, some of operations like deskewing, noise removal, brightness compensation, contrast and gamma correction, distortion removal can be done. Next operation is page recognition, it usually consists of two main steps: - page layout analysis (finding page elements like: blocks of text, tables, lines, single characters), - character recognition. These steps are strictly connected. During the recognition process, the system creates hypotheses about recognized object (character, part of character or some characters). Next step is the verification of these hypotheses, the system tries to find all the parts of each character (lines, dots, circles) and relations between them. The last step is searching for the context of current recognized character, which allows to recognize damaged characters.

CHAPTER 5

IMPLEMENTATION

Chapter 05 : Implementation

With the purpose of generating a system with overall functionalities on manual surveys, generating dynamic forms are the first phase of the cycle. So there should be the ability to generate dynamic forms, saving them in a database in order to reuse and print. By printing the forms it makes the use of having feed-backs in the mode of handwritten.

Creating dynamic questionnaires as per the requirement of the client include below sub tasks in it.

- Form can be defined using a tool box which includes Radio buttons, Checkbox , Single line input and Multi line comments as question type.
- The system provides the ability to save and print the generated questionnaire

It has been used a free and open source software for the questionnaire generating phase.<https://www.jdsoft.com/jd-esurvey.html>

And this survey includes the statistic generating phase too. So the new system is a combination of jdesurvey and the new OMR system for handwritten questionnaires.

In order to realize the principal purpose of developing a new OMR software is to support the concept of extracting the information accurately and to have the usage in large scale processing. Indeed, there are many software models for that each having several issues within that. So the new system addresses the human needs of an OMR system while processing. For example, most of the systems reject forms while reading and the user doesn't know the reason. So, there should be a methodology to express the place where the error occurred.

The OMR system functionality divided into separate process phases [5]: Scanning, Meta-data Feeding, Preprocessing, Processing Filled Forms and Error Correction. As the First step, system needs to feed the empty template with metadata of questions.

The payload for meta data of a questionnaire should be as follows.

```
{
  "questionnaireName": "MCS Survey Tester",
  "questions": [
    {
      "questionType": "SELECTOR|WRITTEN",
      "selectorType": "OPTION|CHECK",
      "layout": "TABULAR|LINEAR",
      "answerContentPosition": "TOP|LEFT|RIGHT",
      "tabularQuestionPosition": "TOP|LEFT|RIGHT",
      "question": {
        "left": 0,
        "top": 1000,
        "width": 2000,
        "height": 600
      }
    }
  ]
}
```

- Upload empty questionnaire template and feed questionnaire meta-data to the system
- Import the empty image to define layout
- Partition the questions by marking the area
- The area marking ability will give the area coordinates systematically
- Other metadata such as question type(OPTION | CHECK), layout(TABULAR | LINEAR), answer position(TOP | LEFT | LINEAR), tabular question position(TOP | LEFT | LINEAR) and pass the meta data and template file to the back end
- This metadata keep in the memory to use later
- Feed the input files and convert the image from RGB to Grayscale
- Applying threshold values convert the gray image to binary image
- Scan each section and identify result marked area
- Identify the respective result text by using metadata saved earlier
- For error occurring scenarios mark that area with red rectangle in another copy of image and move to next question
- After complete the scanning phase check for all errors and manually check whether the error can be ignored
- Generate statistical report after completion of scanning and preprocess phases.
- Scan manually filled questionnaires and retrieve answers

So, the system is developed in two sections as given below.

- Create dynamic questionnaires
- Process manually filled feed-backs

During the pre-process, some of operations like noise removal, brightness compensation, contrast can be done. Next operation is page recognition, it usually consists of two main steps: - page layout analysis (finding page elements like: blocks of text, tables, lines, single characters), - character recognition. These steps are strictly connected. During the recognition process, the system creates hypotheses about recognized object (character, part of character or some characters) according to the metadata that we have feed during previous step. The questionnaire is divided into separate partitions question by question. According to this partitions, processing is happening throughout the paper. Next step is the verification of these hypotheses, the system tries to find all the parts of each character (lines, dots, circles) and relations between them. The last step is searching for the context of current recognized character, which allows to recognize damaged characters.

The system is a platform independent web application, so that the system can be deployed on any operating system and the user can access the system by using any web browser. OpenCV , Java and AngularJS are the main programming languages that are used for the development of this system. So that the system is developed with the latest technologies which are using now a day in the industry.

The new OMR tool has been written in order to decrease the errors while processing. It is easy to decrease errors before processing. So the recognition phase will be the best phase for preprocessing other than rejecting the forms while reading. For example existing systems

With the comparison of other OMR systems, this OMR system provides below all functionalities through a single system.

1. Ability to design questionnaires/ feedback forms dynamically
2. Ability to specify answer template for evaluation
3. Ability to upload scanned images
4. Read scanned images and retrieve results
5. Marking out error occurring scenarios as a warning
6. Provide statistics for the questions

This newly proposed software is an application which enables implementation of OMR but with lesser count of error occurring scenarios than ordinary scanner. This will be provides support to design the layout of the sheet based on the questionnaire we used. This layout will be the design of the sheet by partitioning the question by question. Every question includes its meta data such as question existing area, answer format such as text , checkbox or radio button and the way of

answers lays on and the question expectation of answers such as multiple or not. Then after initializing the layout, any user can feed scanned images of filled forms as required, preprocess error occurring scenarios and generate statistical report to whom information is desired. The scanned image files will then be provided as input to the software, processing will be done. The back end implementation will be done using Java and OpenCV. At the same time frontend will be implemented using open-source software.

The steps used for the OMR implementation are as follows.

1. Split the image into questions.



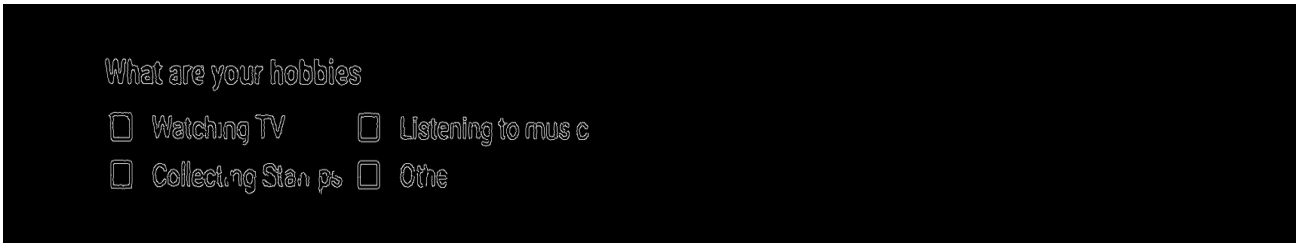
2. For each question preprocessing should run as follows.

```
preprocess {  
    Convert image to gray-scale  
    Blur the image to reduce noise  
    Sharpen the image to identify edges  
    Get the edges of the image  
}
```

The respective code is given below.

```
public static Mat preprocess(Mat original) {  
  
    Mat preprocessed = new Mat(original.rows(), original.cols(), original.type());  
    Imgproc.cvtColor(original, preprocessed, Imgproc.COLOR_BGR2GRAY);  
    Imgproc.GaussianBlur(preprocessed, preprocessed, new Size(5, 5), 4, 0);  
    Imgproc.Canny(preprocessed, preprocessed, 100, 225);  
  
    return preprocessed;  
}
```

The preprocessed image is given below



3. Post process

post-process {

If multiple answers selected for option button - mark the problematic selections

If written - send to an ocr (tesseract) - output not accurate

}

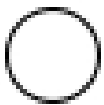
4. Save the response

The questions divided into two sections as per the choices.

1. Checkbox - multiple answers allowed
2. Option button - one answer allowed

So there should be a methodology to know that whether the answer is selected or not. The logic that the implementation follows is to count the number of contours that the system could read after matching the shape. If the answer is not selected, the system reads the number of contours as two for the empty check box or option button. So for the cases which reads the number of contours less than or more than two considered as selected answer.

The system is given inputs for the processing phase as below.



System use this as a sample for the option button questions and so for the empty options it will get the contour count as two.



For the questions with checkbox, it has been used for the sample for shape matching.

Then the logic that the system used for extracting information from survey form is differs from question to question.

1. Linear question {

Group the edges

First group is question section

From the remaining group the edges vertically:

Then you get "symbol, answer, symbol, answer ..." sequence

}

2. Tabular question {

Group the edges

Out of the remaining groups:

Get the top-most and left-most groups those are questions and answers forming the matrix the remaining are the options:

Group them vertically then you will get a matrix for symbols (check-box or option-button)

}

3. Text/written {

Group the edges

First group is question section

Apply canny edge detector with higher threshold. So it will remove the box the remaining are the edges of the written text.

}

The code used for the linear template processing is given below.

```
@Override
public Layout processTemplate(Mat questionnaire, TemplateQuestionTO metadata) {

    AreaTO questionArea = metadata.getQuestion();
    Mat question = questionnaire.submat(questionArea.getTop(), questionArea.getTop() + questionArea.getHeight(),
        questionArea.getLeft(), questionArea.getLeft() + questionArea.getWidth());

    Mat preprocessed = ImageProcessorHelper.preprocess(question);

    MatOfPoint selector = selectorTypeLoader.getSelector(metadata.getSelectorType());

    List<MatOfPoint> contoursPoints = new ArrayList<>();
    Imgproc.findContours(preprocessed, contoursPoints, new Mat(), Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);

    List<Rect> contourRects = contoursPoints.stream()
        .map(Imgproc::boundingRect)
        .collect(Collectors.toList());

    RectangleTO questionRectangle = getQuestion(contourRects);
    String questionText = performOcr(ImageProcessorHelper.getInputStream(question, questionRectangle));

    List<MatOfPoint> selectors = getSelectors(contoursPoints, selector);
    log.debug("selectors count -- " + selectors.size());

    List<Rect> selectorsLocations = selectors.stream()
        .map(Imgproc::boundingRect)
        .collect(Collectors.toList());

    List<OptionLayout> layouts = ImageProcessorHelper.convertTo2DimensionalLayout(selectorsLocations, questionRectangle);

    List<RectangleTO> optionsBoundaries = getOptionsBoundaries(contourRects, layouts, questionRectangle);

    Mat optionsApplier = question.clone();
    List<List<Option>> options = getOptions(optionsApplier, contourRects, optionsBoundaries, layouts, metadata.getAnswerContentPosition());

    LinearLayout linearLayout = new LinearLayout();
    linearLayout.setLocation(FormReaderTransformer.of(metadata.getQuestion()));
    Question layoutQuestion = new Question();
    layoutQuestion.setText(questionText);
    linearLayout.setQuestion(layoutQuestion);
    linearLayout.setSelector(metadata.getSelectorType());

    List<OptionsColumn> columns = options.stream()
        .map(list -> {
            OptionsColumn optionsColumn = new OptionsColumn();
            optionsColumn.setOptions(list);
            return optionsColumn;
        })
        .collect(Collectors.toList());
    linearLayout.setAnswers(columns);

    Imgcodecs.imwrite(TestFormReaderService.OUTPUT_FILE_LOCATION + "linear-full.jpg", question);
    Imgcodecs.imwrite(TestFormReaderService.OUTPUT_FILE_LOCATION + "linear-preprocessed.jpg", preprocessed);

    Mat selectorsDetector = question.clone();
    selectors.forEach(s -> Imgproc.drawContours(selectorsDetector, Collections.singletonList(s), -1, new Scalar(255, 0, 0), 2));
    Imgcodecs.imwrite(TestFormReaderService.OUTPUT_FILE_LOCATION + "linear-selectors.jpg", selectorsDetector);

    Mat questionDetector = question.clone();
    Imgproc.rectangle(questionDetector, contourRects.get(0), new Scalar(255, 0, 0), 2);
    Imgproc.rectangle(questionDetector, FormReaderTransformer.of(questionRectangle), new Scalar(0, 255, 0), 1);
    Imgcodecs.imwrite(TestFormReaderService.OUTPUT_FILE_LOCATION + "linear-question.jpg", questionDetector);

    Mat layoutDetector = question.clone();
    layouts.forEach(layout ->
        Imgproc.rectangle(layoutDetector, FormReaderTransformer.of(layout.getBoundary()), new Scalar(255, 0, 0), 1)
    );
    optionsBoundaries.forEach(o ->
        Imgproc.rectangle(layoutDetector, FormReaderTransformer.of(o), new Scalar(255, 0, 0), 1);
    );
    Imgcodecs.imwrite(TestFormReaderService.OUTPUT_FILE_LOCATION + "linear-layout.jpg", layoutDetector);

    return linearLayout;
}
```


CHAPTER 6

ANALYSIS AND RESULTS

Chapter 06 : Analysis and Results

The new OMR software aims at generating, reading and retrieve handwritten data. So that this software can be applied for any handwritten feedback form applications. The techniques and algorithms used for OMR tool is very accurate and can decrease the error occuring scenarios while reading. The main challenging aspects exist in this new software is to dealing with the front end development, since the system needs the exact coordinates as meta data in order to retrieve information. The empty image needs to be marked with the mouse as per the question by question and the area which the question exist should be sent to back end in order to process the data.

The answer formats that the system could process for the data extraction.

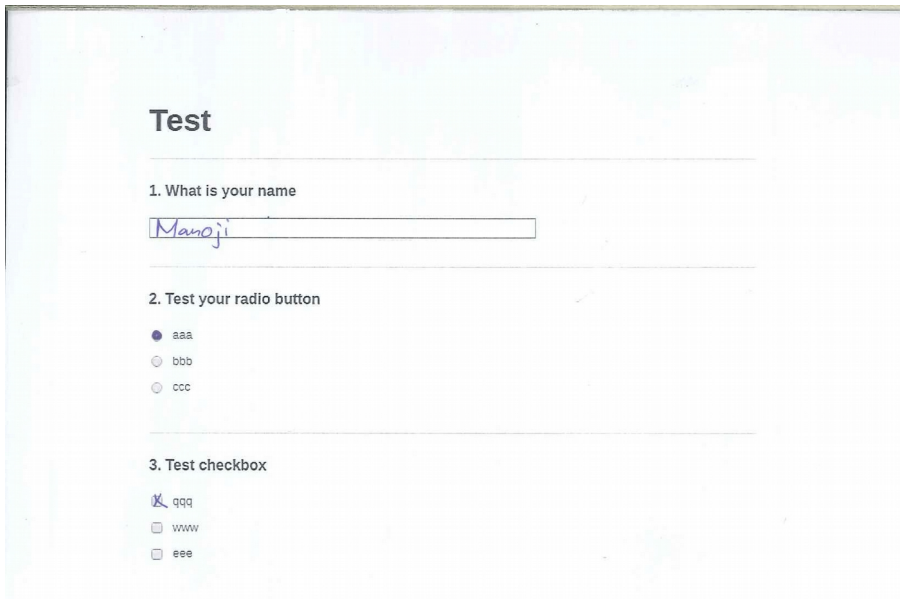
1. Check box
 - Properly marked multiple options
 - Options marking lines lays the outside of the box

2. Radio Buttons
 - Single option marking
 - Multiple option marking
 - Empty answers
 - Darken area lays outside of the radio button

3. Text Area
 - Properly filled in given area
 - Filled text lays out from the text area

Few samples that we have used for the system evaluation is given below.

Ex 1:



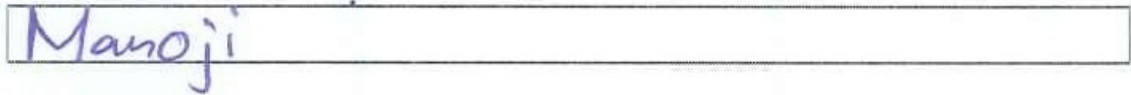
The screenshot shows a web form titled "Test". It contains three sections:

- 1. What is your name**: A text input field containing the handwritten text "Manoji".
- 2. Test your radio button**: Three radio button options: "aaa" (selected), "bbb", and "ccc".
- 3. Test checkbox**: Three checkbox options: "qqq" (checked), "www", and "eee".

Result :

Question1 : What is your name

Answer :



Manoji

Question2 : Test your radio button

Answer : aaa - True

Question3 : Test checkbox

Answer : qq - True

Ex 2:

Test

1. What is your name

Manoji Kularathna

2. Test your radio button

- aaa
- bbb
- ccc

3. Test checkbox

- qqq
- www
- eee

Result :

Question1 : What is your name

Answer :

Manoji Kularathna

Question2 : Test your radio button

Answer : aaa - True

Question3 : Test checkbox

Answer : qqq - True
eee - True

Ex 3:

The screenshot shows a test interface with the following content:

- Test**
- 1. What is your name**
- 2. Test your radio button**
 aaa
 bbb
 ccc
- 3. Test checkbox**
 qq
 ww
 ee

Result :

Question1 : What is your name

Answer :

A screenshot of the text input field from the test form, showing the handwritten answer "Manji".

Question2 : Test your radio button

Answer : This displays the error occurring scenario based on the rules

A screenshot of the radio button options for Question 2, enclosed in a red rectangular box. The options are:

- aaa
- bbb
- ccc

Question3 : Test checkbox

Answer : qq - True

Ww - True

No of samples taken for the evaluation is 200 questionnaires from different backgrounds. The level of education lays on several sets as school, undergraduate, bachelors and others. According to the sample data that the system has taken for the evaluation, 87% of sample extract data properly and the rest marked the error occuring scenario within the question. So that the system user can investigate these errors occuring scenario before they put into report generation phase.

CHAPTER 7

CONCLUSION

Chapter 07: Conclusion

The ultimate goal of any system should be for the well being of the clients. Existing systems facilitates the either one of the requirements such as generating questionnaires or read scanned forms. But in this system we have those facilities in one system.

But in the new system we provide the solution for loopholes of existing OMR systems.

Generally most of the OMR systems do the wrong detections since the system expects a perfect answer sheet marking. But there should be a way to omit those errors since most of them happens due to human error. In the other hand there should be a way to give instructions by saying that how to evaluate the questionnaires. But most of the OMR systems having the same rule set of evaluating questionnaires. For example, for radio buttons the system expects only one answer and for check box questions, the system expects one more answers. Likewise the existing systems have below errors while processing.

- * The cross is not inside the checkbox, but next to it
- * People cross the same box multiple times
- * People use very thick pens
- * Text area filling out is not done properly

As per the sample data processed by the system, it gives 87% accurate results from the input data. And the rest, it marked as the error occurring area with the use of red colored box.

References

- [1] B. Gaikwad ,"*Image Processing Based OMR Sheet Scanning*" ,International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 3, March 2015
- [2] K. CHINNASARN, "*An image-processing oriented optical mark reader*", Applications of digital image processing XXII , Denver CO, 1999.
- [3] Pegasus ICR/OCR/OMR component for in32 and .net , <http://www.pegasustools.com>
- [4] Stephen Hussmann, Leona Chan, C. Fung, M. Albrecht, "*Low Cost and high speed Optical mark reader based on Intelligent line Camera*", Proceedings of the SPIE AeroSense 2003, optical pattern recognition XIV, Orlando, Florida, USA, vol. 5106, 2003. p. 200–08.
- [5] Stephen Hussmann and Peter Weiping 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.
- [6] Hui Deng, Feng Wang, Bo Liang, "*A low-cost OMR solution for educational applications*" Parallel and Distributed processing with Applications 2008, ISPA '08, International Symposium, December 2008.
- [7] K. Toida, "*An Overview of the OMR technology: based on the experiences in Japan*", Workshop on Application of new information technology to population: Paper based data collection and capture, Thailand, 1999.
- [8] A Gupta,S Avasthi, "*Image based low cost method to the OMR process for surveys and research*",International Journal of Scientific Engineering and Applied Science (IJSEAS), Volume-2, Issue-7,July 2016 ISSN: 2395-3470
- [9] Rakesh S, Kailash A, Ashish A ,"*Cost Effective Optical Mark Reader*", International Journal of Computer Science and Artificial Intelligence Jun. 2013, Vol. 3 Iss. 2
- [10]Sebastian Stoliński, Wojciech Bieniecki , "*Application of OCR systems to processing and digitization of paper documents*" ,Computer Engineering Department, Technical University of Łódź, Poland
- [11] C. C. Tappert, Cursive Script Recognition by Elastic Matching, IBM Journal of Research and Development, vol. 26, no. 6, pp.765-771, 1982.