Filled-in Handwritten Data Extraction from Documents based on Geometrical Features

A Thesis Report

submitted in partial fulfillment of the requirements

for the award of degree

of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Piyush Jain

(801432020)

Under the supervision of:

Dr. Shivani Goel

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2016

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Filled-in Handwritten Data Extraction from Documents based on Geometrical Features***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Computer Science and Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Shivani Goel*** and refers other researcher's work which are duly listed in the reference section.

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

Piyush Jain
801432020
ME (CSE)

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

Dr. Shivani Goel
Assistant Professor
Computer Science and
Engineering Department
Thapar University

Countersigned by

Dr. Maninder Singh
Head
Computer Science and Engineering Department
Thapar University
Patiala

Dr. S. S. Bhatia
Dean (Academic Affairs)
Thapar University
Patiala

i

ACKNOWLEDGEMENT

I would like to express my most sincere appreciation and deep sense of gratitude and indebtedness to my guide **Dr. Shivani Goel** , Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala for providing me continuous guidance throughout the research. It has been possible to proceed in the correct direction and successfully complete the thesis with her valuable suggestions. I would like to thank her for encouraging my research and for being actively involved in my work.

I am also thankful to Dr. Maninder Singh, Head of Department, CSED and Dr. Ashutosh Mishra, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I am also very thankful to the entire faculty and staff members for the direct-indirect help, cooperation and support.

Piyush Jain
(801432020)

ABSTRACT

Filling out forms is one of the oldest and widely used methods for collecting information in different fields from an applicant. Automating the scanning process for large volume of office data for example, bank cheque, aadhar card forms, driving license forms, new account opening form etc. not only escalates the office productivity but also reduces time consumption. A simple and systematic approach has been proposed based on geometrical features of the form, to extract handwritten data. The system is first fed with some geometrical and field details from an empty form, it then recognizes different fields within filled-in form using normalized correlation coefficient techniques and then estimates relative position of handwritten data. The handwritten data obtained is then processed with Hough Transform for removing unnecessary lines occurred due to bounding boxes, created to direct users for data entry area in blank form. The experiment was conducted over four different kind of forms with more than 120 filled forms, 30 instances of each type. An average accuracy of 93.82% has been recorded. Thus system extracts handwritten data efficiently.

Keywords: Handwritten data extraction, Forms, Normalized Correlation Coefficient Technique, Hough Transform.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Filling out forms is one of the oldest and widely used methods for collecting information in different fields from the applicant. Automating the scanning process of large volume of office data such as cheques, aadhar card forms, driving license forms, new account opening forms, can escalate the office productivity as well as reduce time consumption. With the recent advances in technology, the manual filling of data is widely replaced by computerized data. Almost all the forms and data are submitted online. Also there is a deep requirement that the information collected offline using form filling should be available online for faster access in near future. Data available online can also be easily manipulated. Automation is basically demanded in places such as in income tax offices, banks, post office, municipal department, colleges, university, where large amount of data is to be manipulated. This problem is very recent as there is a rapid emergence of data collection offline. Many researchers are working over this issue and have developed numerous algorithms. Some of them are based on predefined form knowledge while other methods estimate the form setup automatically. Forms that estimate form data and handwritten data automatically are usually more error prone. It is always beneficial to first convey to the system about the form from which data is to be extracted That is why, handwritten data extraction system which is form specific is more accurate and will extract data with lesser errors. Moreover, it has been observed that in offices the forms that are being distributed are static, which means the fields do not change rapidly over time. So, it is again a good idea to go for a handwritten data extraction system which is form specific.

Extracting data from handwritten filled form may have several hurdles. The application form to be filled may have check boxes, dotted lines, signature, boxes, straight lines etc. Some application forms have square boxes for data entry where each box can only contain single character or may either be left blank. The single character written inside could be printed or handwritten. As of now most of the data entry is performed online and this has paced the process. But in majority of the offices data entry is still offline. They are collected over sheet of paper and then typed back

into computer manually. Automating this process with the help of computer vision may include several steps. One of the major tasks is extracting the handwritten data from the application form. The extracted data can be used for many purposes for example archiving and documenting. The extracted data can also be given to optical character recognition engine to convert it to corresponding Unicode number. This will help organize data and may improvise data processing.

Recognizing relative locations of information within form is another important process. Some papers suggest that by recognizing lines using Histogram techniques to estimate the location of data, but this may fail if the lines are hiding behind a content occupying several lines at front. This process can be improved if the form format is known. Template matching can be used for analyzing the relative position of data fields and then estimating the location of handwritten data, this will improve the probability of finding required data accurately, reducing false positive results.

Another commonly found element in an application form is straight lines. So their extraction is very important. Straight lines are often found around data to be extracted. Straight line detection problem is often found when we are designing a system where data to be extracted is in a completely unknown form format or there is a very huge skewness or rotation is involved. This problem can be reduced if form format is already known and the user is instructed to perform scanning operation in a controlled environment. This kind of setup will not only improve the accuracy but also increase the flexibility to involve not only the handwritten text data but also signatures, fingerprints, color photographs and so on to be entered into database.

Selection of appropriate feature recognition method is the most important aspect for data extraction from form images. Several methods for feature recognition have been proposed till date.

1.2 Brief Background

The various techniques which have already been researched are discussed in the following section. Some of them like correlation techniques is discussed in section 1.2.1 .These methods can also be used optionally to check performance of the proposed system.

1.2.1 Correlation Techniques

In this work, in place of only using a single matching algorithm as described in section 4.3, we can use various other techniques discussed in the following sections.

There are basically 3 main correlation techniques and one normalized technique [16]:

i. Square Difference Matching Methods
ii. Correlation Matching Methods
iii. Correlation Coefficient Matching Methods
iv. Normalized Methods

1.2.2 Square Difference Matching Methods

In this method, a squared difference between pixel values of input image and template image is calculated. This is equivalent to finding error like variance between images, where first set of values come from input image and second set of values comes from template image. The template is placed over each pixel and each time a sum of square difference is calculated. The sum of square difference having the least value is considered to be the best match and higher value is considered as bad match. A match with sum of square difference equal to zero is called an exact match. The more it is closer to zero, the better the match. Sum of square differences can be represented with the following equation [23] and result is shown in Figure 1.1.

$$(x, y) = \sum_{x',y'} [T(x', y') - I(x + x', y + y')]^2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.1)$$

Where,

$$\mathrm{R(x, y)} = \text{Sum of square difference at position x, y}$$

$$\mathrm{T(x', y')} = \text{Pixel value in template at position x' and y'.}$$

$$\mathrm{I(x + x', y + y')} = \text{Pixel value in input image at position x + x' and y + y'.}$$

Figure 1.1: Square Difference Matching Result

1.2.3 Correlation Matching Methods

In Correlation Matching method, instead of finding the difference we simply multiply the pixel values of template image with the pixel value of input image. Here also, the template is placed over input image and the calculation is performed for each and every pixel. The pixel which is being modified after obtaining result is usually taken at the centre of template. The resulting pixel intensity with higher value is considered to be a best match while a lower value is taken as a bad match. Its polarity of result is just opposite of what we have discussed in the previous method, square difference matching method. Correlation matching method can be presented mathematically in the following way [14] and result is shown in Figure 1.2:

$$R(x,y) = \sum_{x',y'} [T(x',y').I(x+x',y+y')]^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1.2)$$

Where,

$$\mathrm{R(x,y)} = \text{Intensity value after correlation matching at position x, y}$$

$$\mathrm{T(x',y')} = \text{Pixel value in template at position x' and y'.}$$

$$\mathrm{I(x+x',y+y')} = \text{Pixel value in input image at position x + x' and y + y'.}$$

Figure 1.2: Correlation Matching Result

1.2.4 Correlation Coefficient Matching Methods

This method is a successor of the correlation matching method. The basics of this algorithm is same as correlation matching and includes few more steps like finding mean. In this method instead of simply finding a multiplicative match between template and input image, multiplicative match relative to the mean value of template with the mean value of corresponding section of input image is done. Here unlike the correlation matching method, a good match will yield value close to 1 and a complete mismatch will yield value equal to -1, while 0 will show that there is no match (correlation) between template and input image segment. Correlation coefficient matching methods can be shown mathematically in the following way [18] and result is shown in Figure 1.3.

$$R(x,y) = \sum [T'(x',y').I'(x+x',y+y')]^2 \dots\dots\dots\dots\dots\dots (1.3)$$

$$T'(x',y') = T(x',y') - \frac{1}{w.h\sum T(x'',y'')} \dots\dots\dots\dots\dots\dots\dots\dots (1.4)$$

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{(w.h)\sum I(x+x'',y+y'')} \dots (1.5)$$

Where,

$R(x, y) =$ Intensity value after correlation matching at position x, y

$T(x'', y'') =$ Pixel value in template at position x'' and y''.

$I(x + x'', y + y'') =$ Pixel value in input image at position $x + x''$ and $y + y''$.

$T(x', y') =$ Pixel value in template at position x' and y'.

$I(x + x', y + y') =$ Pixel value in input image at position $x + x'$ and $y + y'$.

$T'(x', y') =$ mean subtracted pixel value at position x' and y'.

$I'(x + x', y + y')$

$=$ mean subtracted Pixel value in input image at position x $+ x'$ and $y + y'$.

$w =$ Width of template image

$h =$ height of template image

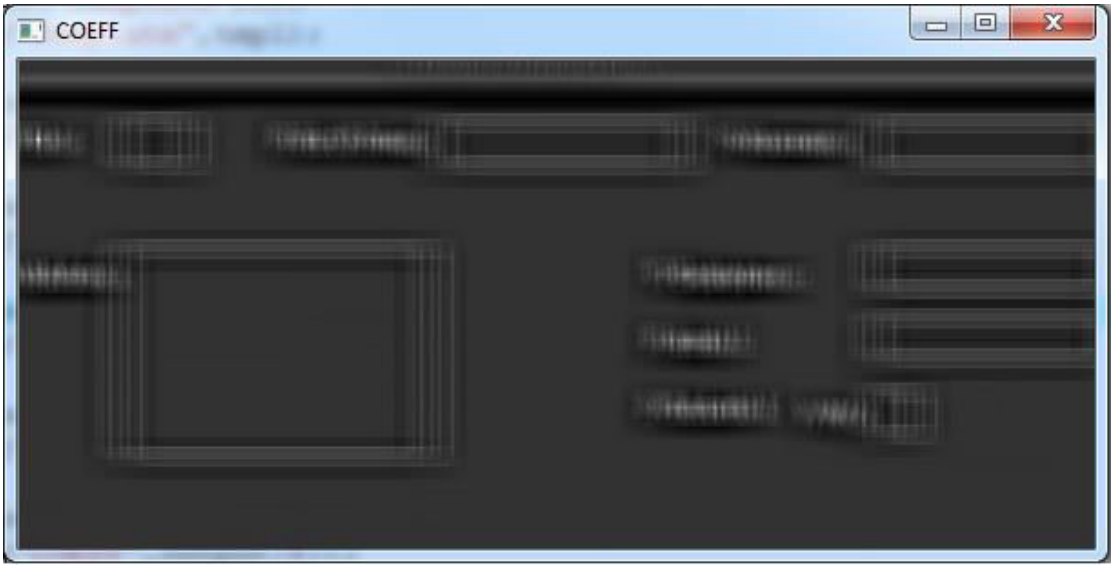


Figure 1.3: Correlation Coefficient Matching Result

1.2.5 Normalized Methods

Normalized methods are like an add-on. It can be combined with any of the above methods to form a new advanced mathematical equation. Basically the advantage of normalized equation over each of the above non normalized equation is that it reduces the lighting effect difference between template image and input image section. The normalization coefficient is calculated by the same formula in each of the above mentioned methods. For normalization, the normalization equation can be derived as follows [30].

$$Z(x,y) = \sqrt{\sum_{x',y'} T(x',y')^2 . \sum_{x',y'} I(x+x',y+y')^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1.6)$$

Where,

$$\mathrm{Z(x,y)} = \text{Normalized coefficient}$$

$$\mathrm{T(x',y')} = \text{Pixel value in template at position x' and y'.}$$

$$\mathrm{I(x+x',y+y')} = \text{Pixel value in input image at position x + x' and y + y'.}$$

By analyzing the equation, it is observed that the normalization equation is the square root of the product of squares of the summation of pixel values in template and input image section.

For each of the above equations, the normalized values for pixel substitution can be computed as follows [32] [30] [4]. Figure 1.4 shows normalized Squared Difference result, Figure 1.5 shows Normalized Correlation Result and Figure 1.6 shows Normalized Correlation Coefficient Result.

$$R_{sqr_{diff_{normed}}}(x,y) = \frac{R_{sqr_{diff}}(x,y)}{Z(x,y)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.7)$$

$$R_{corr_{normed}}(x,y) = \frac{R_{corr}(x,y)}{Z(x,y)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . (1.8)$$

$$R_{corr_{coeff_{normed}}}(x,y) = \frac{R_{coeff}(x,y)}{Z(x,y)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.9)$$

Figure 1.4: Normalized Squared Difference Result

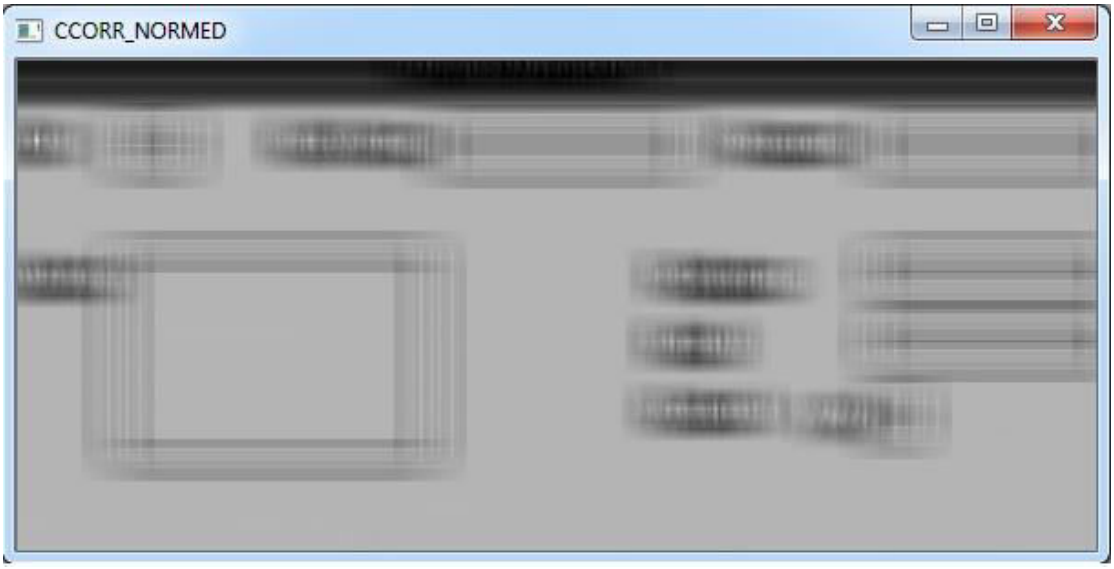


Figure 1.5: Normalized Correlation Coefficient Result

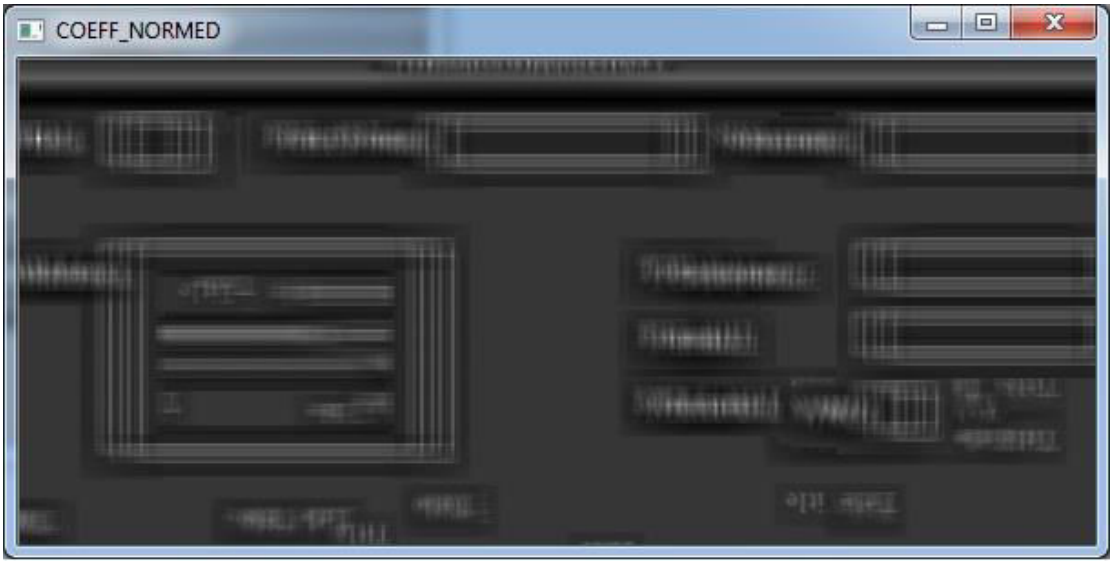


Figure 1.6: Normalized Method Result

Finding a normalized pixel value at position x, y does not consume much resources, it is shown that the normalized valued is simply the division of non-normalized value with the normalization coefficient.

It has been observed that the accuracy of pattern recognition from template matching is giving best results with correlation coefficient with normalization [17]. However simple square difference matching method is the fastest one but not that much accurate [25]. There has always been a tradeoff between accuracy and speed; it is the choice of the user to decide what they require the most.

1.2.6 Hough Line Detection

Hough line transform detects all lines in a given image, even the small ones. To eliminate those lines which need to be removed, a threshold value is placed [7]. The length of line detected is calculated with the help of Euclidian distance formula [12], which is expressed as follows:

$$Dist = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.10)$$

Here (x_1, y_1) and(x_2, y_2) are the start and end points of a line detected with the above method. Any line, whose 'Dist' value is exceeding a specific threshold value, is considered as a part of redundant lines present due to outline box around the handwritten text.

It is known that a line can be represented in two ways [15]:

i. Cartesian coordinate system
ii. Polar coordinate system

In Cartesian coordinate system a line is represented by two parameters, that is,

i. Slope(m)
ii. Intersection point on y axis (c)

While in polar coordinate system we have two parameters as shown in Figure 1.7

i. Length of perpendicular line meeting with origin(d)
ii. Angle made with x axis of the perpendicular line(Ɵ)

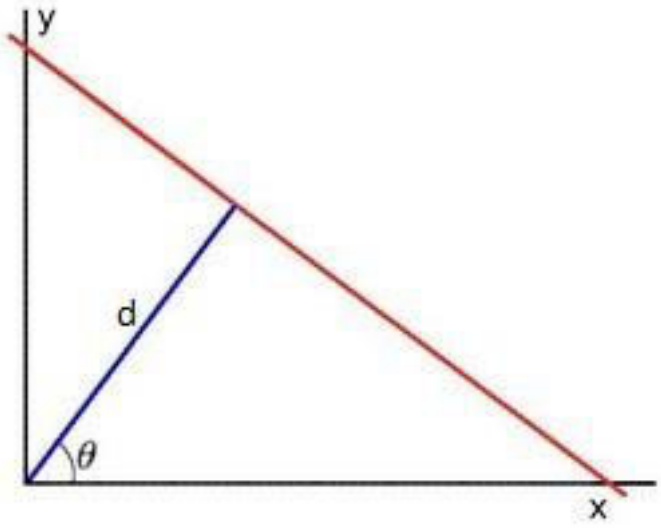


Figure 1.7: Line Representation [8]

A line in polar coordinate [31] can be expressed in the following way

$$d = x\cos\theta + y\sin\theta \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.11)$$

For every point (x',y') we can produce a series o+9-f lines representing it.

$$d_\theta = x'.\cos\theta + y'\sin\theta \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.12)$$

Here (d_θ, θ) is showing each and every line which is passing through the points (x',y').

If we plot each and every line that will pass through the given coordinates (x',y'), then we get a sinusoidal graph as shown in Figure 1.8.

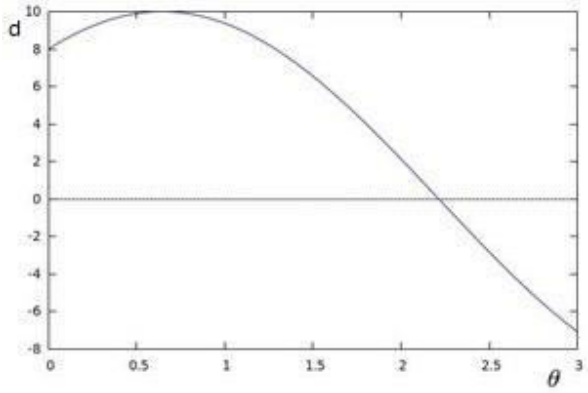


Figure 1.8: Hough Transform Plot [10]

Here we are taking only those points which are having d>0 and$0 < \theta < 2\pi$.

This process is repeated for every foreground pixel in image, and graph is obtained for each of them. If there is an intersection in the graph of any two pixels then it is said that the two points lies in the same line as shown in Figure 1.9.

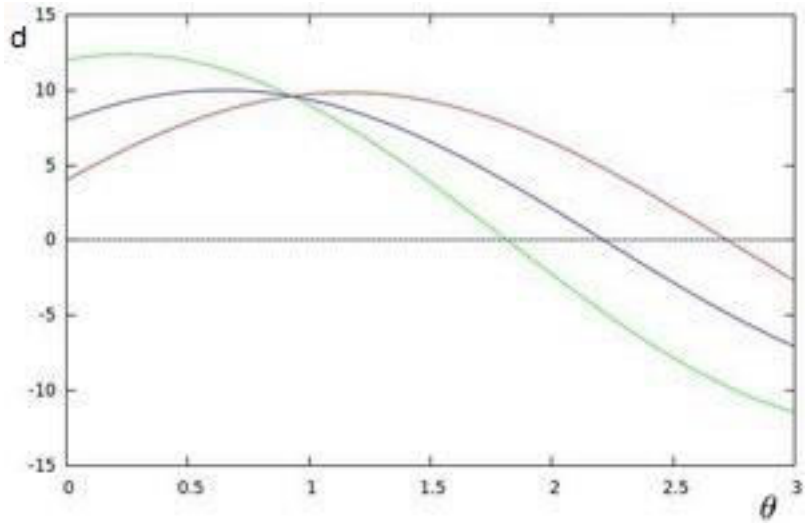


Figure 1.9: Intersection of Hough Transform Plots [10]

If for a combination (d_θ, θ) we see more number of curves intersection with each other, this denotes that it has more number of points covered within a line. In Hough transform a minimum threshold value for number of intersection points is set so as to avoid unnecessary overflow of line detection.

1.3 Organization of the Thesis

The thesis is organized as:

Chapter 2 presents the literature review of the research already done so far in this field. This has helped many researchers to think beyond what has already been done and develop a new idea based on already known methodologies.

Chapter 3 describes the problem statement and the aim of the proposed work.

Chapter 4 explains the complete methodology and implementation of the proposed work.

Chapter 5 shows the results obtained after implementing the proposed work.

Chapter 6 concludes the work done and consists of the discussions about future scope.

CHAPTER 2

LITERATURE SURVEY

There have been many techniques and methodologies developed for handwritten data extraction from a filled in form some of them are discussed briefly in the following section.

Al Faris and Ngah[1] Suggested a handwritten data extraction system based on common patterns like lines bounding the filled data. Here the system rigorously searches for straight lines in both horizontal and vertical direction and then decides area bounding handwritten data. This approach has limitation over the type of form that it can recognize. The entered data must be inside bounding rectangle, forms having straight lines or no lines can be difficult for this system to recognize. Though this method has good flexibility over handling scaling and skew, but it may consume more time as it requires running a CPU intensive operation like line detection using Hough line transform.

Aksak,Feist and Kiiko[20] Suggested a method in which the handwritten data extraction is based on color of ink used to fill data. For different set of field a variety of colors are used to fill them. During processing the system recognizes the color of filled in data and extract them accordingly. Handwritten data which collides with base form foreground is gone through a series of processing for the reconstruction of extracted handwritten data. This method is easy to implement but there is a serious drawback, it requires the person filling the form to use different ink colors for distinct fields, which can be very tedious and may consume more time for users. After all these drawbacks like limiting users, this approach have very good efficiency as all the data that needs to be extracted have distinct features. Making it easy to recognize and separate. Also this method has good tolerance towards skew and non uniform scaling.

Safari, Narsimhamurthi, Shridhar and Ahmadi[29] proposed a system in which the system requires correction of geometric transformations within form, maintaining a standard dimension, it then subtracts the complete form template with the corrected template. The correction of input form fixes various kind of transformations like rotation, non-uniform scaling, left skew, right skew. Proposed system makes use of

histogram and various other techniques for correcting transformations. This method has a strong drawback that it is subtracting the complete form template to the input form. This will in most of the cases may leave unwanted data around the handwritten text, or it may also include form field data, if transformations are not resolved correctly.

Chen and Lee[21] proposed a unique system, they have used strip projection method to extract the form geometric features. At first they have segmented the form taken from a scanner into non uniform vertical and horizontal strips. They have exploited a pattern in strips alignments; they projected the horizontal strips in vertical fashion and vertical strips horizontally. The maximum matching in the projection of the above step denote the possibility of the presence of a line. This is same as Hough line recognition but they have followed different algorithm which they claim to run with less time complexity. After all the lines have been recognized, it is then extracted from the source image. With the extraction of all lines the extra lines are eliminated with line removal technique and the broken lines found are repaired with another algorithm called line merging algorithm. They claimed to run all the above mentioned process with complete recognition and extraction of handwritten data within three seconds from any a4 sized paper form. They have claimed that their procedure is giving results much faster than if used with Hough line transform algorithm.

Fujisawa, Nakano, Kurino [9] suggested an algorithm which is pattern oriented giving a guess about the structure of form document. The first step for extraction of character is the identification of handwritten touching numerals. A search for connected components is performs in place of simple pixels in image. After extraction an interrelation between different pattern found are analyzed which are grouped together to develop a sense along with them. The shape and patter of different strokes are analyzed if there are characters touching each other. About 95% of the touching numeral are recognized correctly with the proposed methodology. If any ambiguities are found then they are handled with various hypotheses and confirmation processes. Document images which are in tabular form are analyzed and can be extracted easily. A semantic relationship between fields like data and label frames information on the input document is recognized properly.

Casey et al. [22] this method is based upon the straight lines within the form which is generally present in every form. Here a series of pixels from the blank form is stored in first array. This array is scanned to find straight connected lines, this signifies the data mask in the form and the data mask represents the data location in the form. Now same process is repeated for filled in form, and a second array is generated. A set of straight lines are detected in the second filled in scanned form and is matched with first array lines for corresponding fields. Thus handwritten data is extracted with knowledge of data mask offset values.

I. Washington, J. Lilling, and W. Zitomer [26] makes use of viewpoints which is used to recognize interior and neighboring pixels of the field form from the scanned input image, thus extracting the handwritten data.

Yu and Jain[24] This paper describes a standard system for form handwritten data segmentation and recognition when the characters or symbols are either touching or coinciding with the form frames. They proposed a technique to take apart these characters from form frames whose coordinates are not known in most of the cases the character strokes in the form are found to be either coinciding with the form frame of simply touching the frame lines, hence the following 3 issues has been addressed closely . first, recognition of from frames; second, extraction of characters and form layout; and third repair of broken strokes introduced after extraction of data. A common pattern of straight lines are exploited to find form frames with the help of block adjacency graph technique. Form frame extraction and character repairing are implemented using this graph. This system claims to learn the structure of forms automatically. it automatically recognizes the form template from a blank form and self adjusts itself with the presence of a little degree of skewness. And from this recognized form features it easily extracts the handwritten data. This system is claimed to have immunity from involvement of skew, but they talked nothing about the scaling problem.

CHAPTER 3

PROBLEM STATEMENT AND OBJECTIVES

3.1 Problem Statement

Currently, Automatic extraction of handwritten text is one of the foremost problems. In many offices, data filled by users are fed into system manually, thus elevating time and efforts. An automated approach for extraction of handwritten data and then passing the obtained results to Handwriting recognition System may reduce efforts made by user. For handwriting recognition system to run efficiently, an efficient handwritten data extraction system is required. The proposed system presented in this work deals with the extraction of handwritten data from a filled input form.

3.2 Gaps in study

Based on the literature survey, the following gaps in the handwritten data extraction from a filled-in form are analyzed:

i. Most of the work is concentrated only in exploiting the characteristics like surrounding boxes and some geometrical features of handwritten texts, but the proposed method focuses on the geometry of the input form as well as some features of handwritten data which are different for each data field. This method has potential of giving more accurate data extraction with fewer computations in a controlled environment.

ii. Most of the proposed systems have limited flexibility over the type of forms, which is being satisfied in the proposed system to a good extent by focusing over the geometry of forms rather only over handwritten data and its surrounding pattern. Also, it provides flexibility over extra margins involved while scanning input form.

iii. Hough line transform for line detection is very resource intensive process. In our proposed system, this algorithm is used in a very limited and smart way, that is, it is used for redundant line removals only which is done after the extraction of the handwritten data and thus reducing the overall resource consumption.

iv. In other proposed systems, the handwritten data extraction was performing a blind extraction, which means it would extract all the handwritten data, while our proposed system has the ability give an option to the user to extract only the data which the user wants to extract , avoiding unnecessary computations performed over extraction of other fields

v. In addition to the above mentioned facilities, our proposed system also tackles with the reconstruction of handwritten data after removal of lines.

vi. In addition to only extracting handwritten data, our proposed system can also be extended to extract data written in checkboxes and other kind of data like signatures from their respective position, which gives a major upper hand over other proposed systems listed so far.

3.3 Aims and objectives

The Basic aims and objective of the proposed system are:

i. To automatically extract the handwritten data from a filled in input form from the user.

ii. To create a precise, efficient and robust system for handwritten data extraction.

iii. To allow users flexibility to choose the fields they want to extract, avoiding unnecessary computations.

iv. A system which has the capability to be used in the real world supporting a variety of form formats.

3.4 Research methodology

There are a total of five phases involved in the proposed system.

Phase 1: In the first phase, a dataset is collected, which is collected through real audience, by handing over the form to them and asking them to fill in accordingly, a total of four different types of forms were taken. Each of the forms had different style and pattern.

Phase 2: In the second phase, a base form from all of the filled in form is selected and is fed to the proposed system along with its geometrical features.

Phase 3: In the third phase, the learned form geometry is tested over other forms filled-in by users. The proposed methodology is applied over it for handwritten data extraction.

Phase 4: In this final phase, Results and accuracy is analyzed and the drawbacks (if any) are taken into account and are resolved.

CHAPTER 4

PROPOSED SYSTEM AND IMPLEMENTATON

In this chapter a proposed approach for extraction of handwritten filled on text is discussed in detail. The proposed approach includes several steps like Acquire Image, Preprocessing, Geometric Recognition of fields and Geometric Estimation of handwritten text. An outlined set of steps are shown in the Figure 4.1:

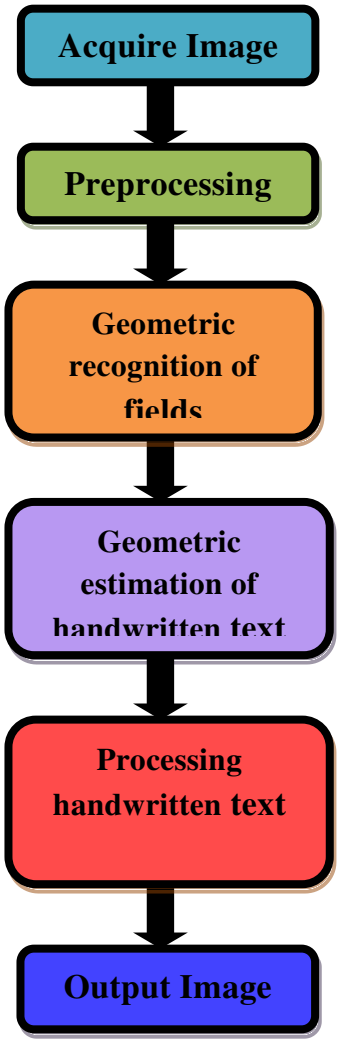


Figure 4.1: Outline of proposed system

Each of the above steps may include one or several sub steps. Each of the steps may or may not have special requirements and same will be discussed in their respective sections. The first step of the proposed system is as follows:

4.1 Acquire Image

The proposed system begins with the Acquisition of a sample form. The sample form taken does not necessarily be empty, even a non blank form can be taken as an input form, as it is only need to care about the geometric positions of the field and handwritten text information in the future steps. The form image acquired can have specific commonly used image format like jpeg, bmp, tiff, png [5]. The sample form can be taken from commonly used input devices like a scanner, digital camera or any other digital input device. For this proposed system it is advised to use scanner since it produces less commonly found image transformations, for example, shear, rotation, scaling, reflection [2]. The proposed system have no limitation over the number of channels within image, the sample form can either be gray scale or colored. A sample form taken during experimentation is shown in the Figure 4.2 below.

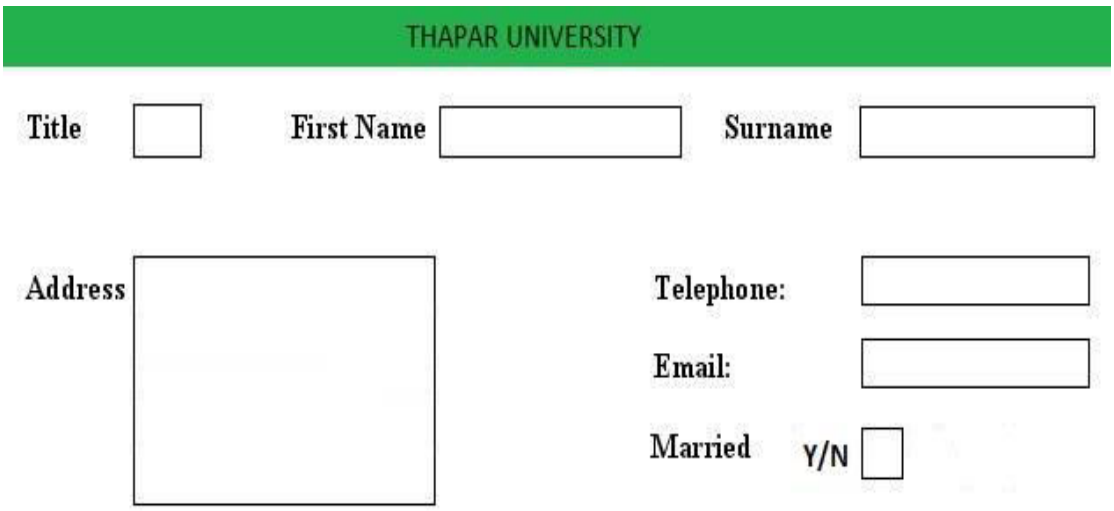


Figure 4.2: Scanned Image

After acquiring a sample form, template images for each of the data fields are extracted manually. This is necessary because in future step called geometric estimation of fields an efficient template matching algorithm to find the estimated position of each of the field. After extracting a template image we then feed in the coordinates of the handwritten text area. The system will calculate distance between template and handwritten text area and save in its database for use in geometrical

estimation of handwritten text module. The distance is calculated with the following formula:

Let,

$$T_x = \text{top right x} - \text{coordinate of template w.r.t sample form.}$$

$$T_y = \text{top right y} - \text{coordinate of template w.r.t sample form.}$$

$$H_x = \text{top left x} - \text{coordinate of handwritten text area.}$$

$$H_y = \text{top left y} - \text{coordinate of handwritten text area.}$$

Then,

$$X' = H_x - T_x \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.1)$$

$$Y' = H_y - T_y \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

Where X' and Y' represents the x coordinate and y coordinate gap between template and handwritten text.

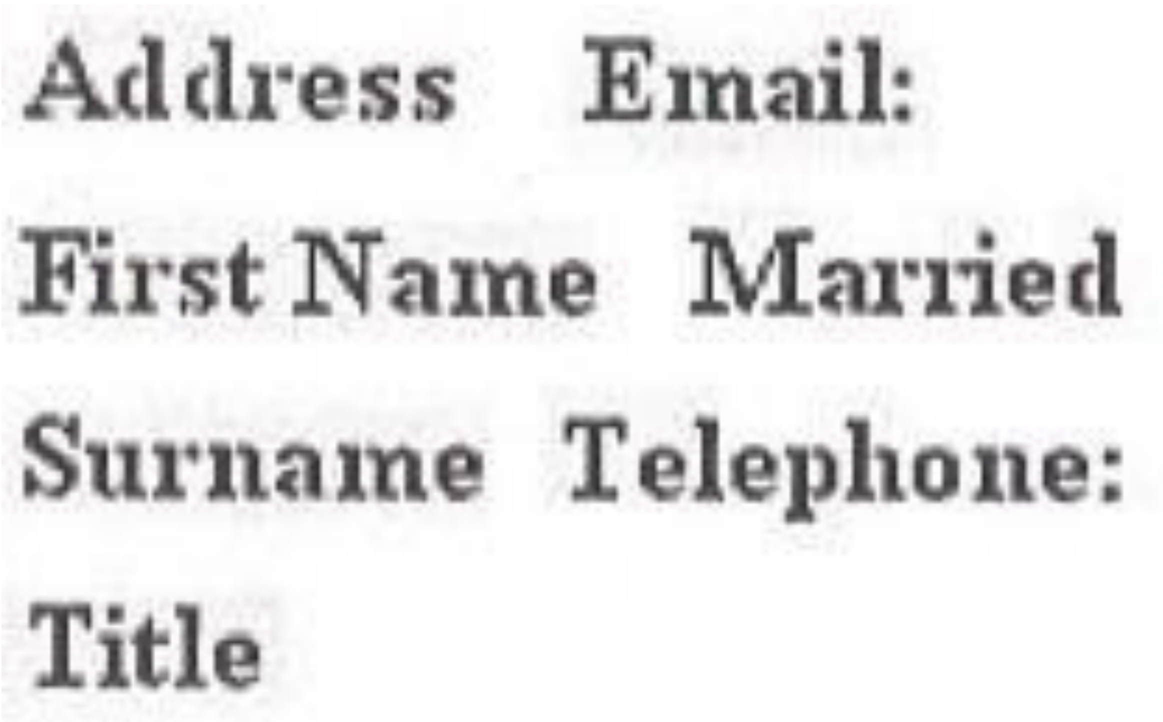


Figure 4.3: Extracted Templates of Fields

After all the form template images (as shown in Figure 4.3) and handwritten coordinate are being stored we can now start feeding in input forms and extract handwritten data from it. The input form can be taken in the same way as described above using scanner. There are no hard restrictions but least skew is preferable. The proposed system can handle uniform scaling efficiently. However a non uniform scaling may result in bad estimations. It is recommended to use a good quality scanner for taking inputs, for better efficiency.

4.2 Preprocessing

The input form scanned from input device is raw. It will be much easier if some operations are performed over this raw input image and then extract the features that we will need for field recognition [33]. The major problem associated by directly processing the raw input is that they include much different kind of noises, for example salt and pepper noise, shades and coloring problems. These noises deteriorate the input image quality and finally reduce the efficiency for accurate recognition. Each of the noise found may involve different methodology to remove them. Some of them are hard to remove, which require some special intelligent techniques, while other are easy to remove but they include some cost of reducing image quality. There are many different general commonly found techniques evolved by various researchers. One of them is by using erosion and dilation technique. This technique will remove salt and pepper noise [28]. There are few preprocessing steps involved before we actually feed the form for next step. These are Gray Conversion, Binarization, and noise removal as shown in the following Figure 4.4.

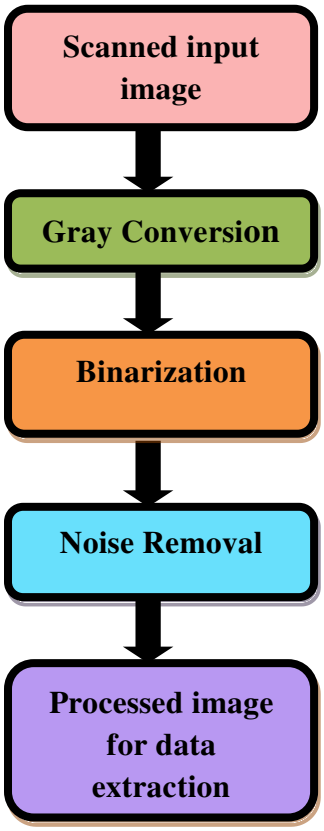


Figure 4.4: Preprocessing steps

4.2.1 Gray Conversion

The scanned image obtained so far has three channels, Red, Green and Blue, each of which has values ranging from 0-255 and can be represented by a byte in computer system [6]. The problem with processing with a three channel image is that they are heavily dynamic with even a slight change in environment. With the grey channel there is a slightly less impact on image from the surrounding changes. Grey images have only one channel with the values ranging from 0-255. Each pixel can be represented with a single 8 bit (byte) in computer. Grey channel represents intensity of darkness or light. They are also called black and white image, though they should not be confused with a binary image since a binary image only contain two values. A general formula for grey image conversion from an RGB image is as follows [3] and the results are displayed in figure 4.5:

$$\text{Pixel}_{\text{value}} = 0.3\text{R} + 0.6\text{G} + 0.1\text{B} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.3)$$

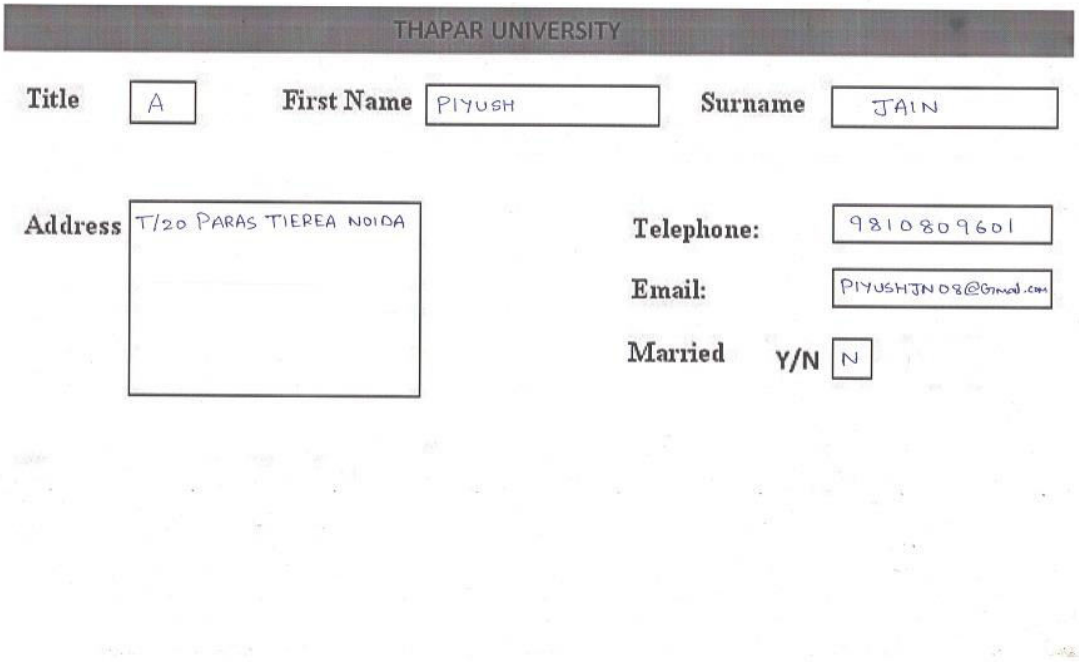


Figure 4.5: Gray Scale Converted Image

By analyzing the equation we can see that green channel has given more than 50% of weightage than other two channels and blue channel has only 10% of the total weightage. Though not necessary but the complete process could also be applied over three channels distinctly, this is a more general and widely accepted approach.

4.2.2 Binarization

In this process, the grey image is converted into a binary image having two values, either 0 (black) or 255(white). This step is necessary because in separates foreground image with the background image. The foreground represents actual part of image that will be considered for recognition and feature extraction, while the background part of the image represented by white pixels are the ones which denotes the passive unimportant parts whose existence does not contribute to the final result. Hence we do not count background into our calculation [27]. Binarizing an image reduces the calculation overhead for any system, since the foreground is easily available for calculations. In this system for binarizing an image we have used a threshold value that is below which the pixel will be considered a part of foreground section, and

above which it will be considered as a background and will be set to intensity value of 255.

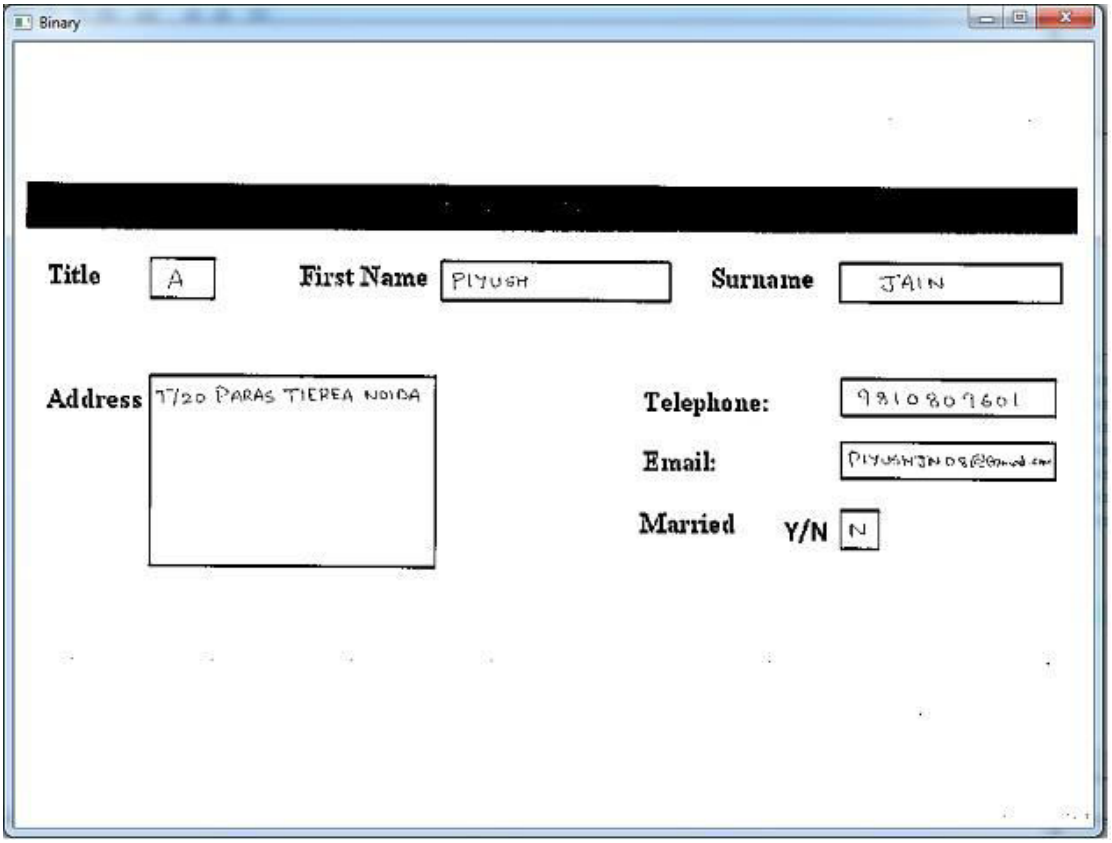


Figure 4.6: Binarized Image

The foreground is given 0 pixel value because as it is known in a form the ink for writing text is slightly dark while pages are generally white in color as shown in figure 4.6. Using this property it is easier to extract foreground (text) from the background (paper). In this system a threshold value of 150 is being used.

$$\text{Pixel\_value} = \begin{bmatrix} 0 & \text{if intensity is below threshold value} \\ 255 & \text{if intensity is above threshold value} \end{bmatrix}$$

4.2.3 Noise Removal

After binarization, some noise might be present in it which may contribute to false recognition and unnecessary computations. There are various kinds of noise which are commonly found in images, but the ones which have a very high probability of occurrence in our system is the presence of unnecessary dark blobs found due to having lower threshold value encountered during binarization process. The small

blobs generally have small radius, these are like small dots in within image. To solve this problem we will perform two operations,

i. Dilation
ii. Erosion

In dilation the brighter region that is pixels having a value of 255 are expanded, leaving behind the darker spots with lower diameter to shrink and disappear. Dilation is like convolution of an input binary image with a kernel. A kernel can be any shape or size, generally having a square shape with a point of interest at its centre. The kernel taken can be imagined as small template which we run over every pixel of binary image. Dilation is represented in the following way[13].

$$I \oplus T = \bigcup_{a \in T} A_a \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4)$$

Where,

$$A_a = \text{Translation of A by a}$$

Performing only dilation will also result in reduction of number of black pixels in the text foreground, which might reduce the recognition accuracy of extracted field template in the first step with input for due to reduction in number of useful foreground pixels. So to overcome this problem we will perform erosion over the image obtained after dilation.

Erosion is a process which is just opposite of what dilation results into. It expands the black region which in turn reduces the white region. The gaps in the letters of a text are produced because of dilation will fill up as well. This will recover most of the damage done by dilation. The result obtained may not look the same way as it was before but it is sufficient enough to gain the recognition rate back to normal. In erosion we have an image and a template which will run over each pixel of entire image. Template has a point of interest, which is generally considered at the center of it. Let there be a 3x3 grid, now we run this grid over every pixel, for each pixel at centre we find the pixel having least value and substitute it to the centre pixel. In a binary image the least value will always be 0. This process will be applied over every pixel. Process of erosion can be represented in the following way [19].

$$I \ominus T = \bigcap_{b \in T} I_{-b} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.5)$$

Dilation followed by erosion works because after dilation all the blobs with small radius are removed, and after erosion, since the small blobs are already removed therefore there is no chance for reappearance or expansion of those small blobs back again, and only the text pixels are recovered [11].

With this we obtain a processed image for analyzing in further process.

4.3 Geometric recognition of fields

This step is specially concerned about the recognition of important fields within the input form to estimate its geometric structure. For every correct field that we recognize, we get to know its geometric coordinates and also we can estimate the corresponding coordinates of handwritten text area. The importance of this step is that it will not only help to estimate the handwritten text area, but it can also we used to counter image distortions like scaling and indefinite margin introduced during scanning process. For most of the cases it is being assumed that the input image we provide has constant dimensions and there are as little distortions as possible, while there is no hard restriction over abnormal margin introduced at any side of form image.

For recognition of fields in the input form we are using template matching algorithms, using the templates that we had collected for understanding the format of form. Before we pass collected templates for recognition process we perform same preprocessing algorithms discussed above. The importance of this step is that it will match the style of template image with the input image. With this we mean that, since the input image will have binary values, but the template images saved earlier will have 3 channels, so it is not possible to compare both of them.

After obtaining the resultant template image and the input image we can now finally start recognizing the fields within input form using any of the following template matching algorithms. The advantage of choosing template matching algorithm is that we can exploit some features of the input form. As we know that the machine printed text like name, address, telephone, gender etc. will have same format in every input scan we make. So it is easier to recognize them using a simpler and less time

consuming template matching algorithms. Some of the recognizing algorithms using template is discussed below.

There are basically 3 main correlation techniques and one normalized technique which are:

i. Square Difference Matching Methods
ii. Correlation Matching Methods
iii. Correlation Coefficient Matching Methods
iv. Normalized Methods

From the above mentioned methods, in our proposed system we have used Normalized Correlation Coefficient Method. The results are shown in the following Figure 4.7.

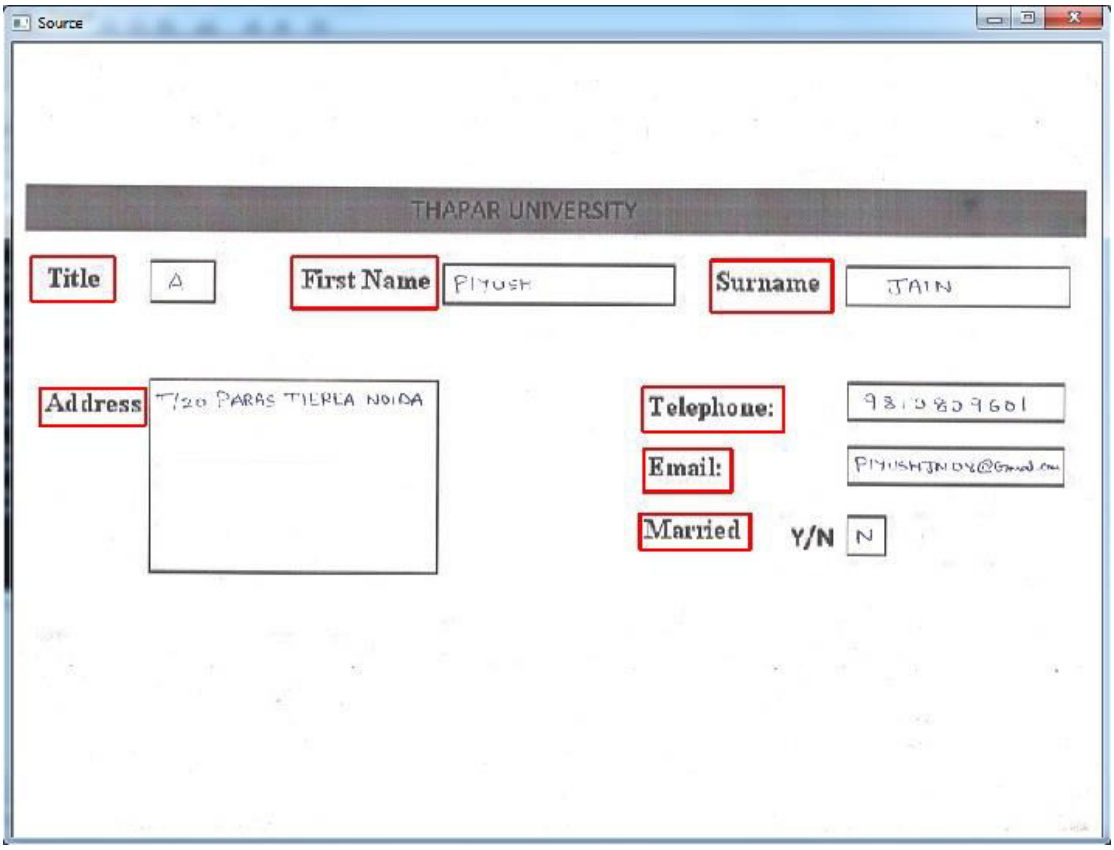


Figure 4.7: Recognized Geometric Locations of Fields

4.4 Geometric estimation of handwritten text

After applying the above method, geometric recognition of handwritten text, we obtain a rough idea about the location of field and its dimensions. Using these dimensions we can estimate the handwritten text starting top left coordinates and its bounding distances. For estimation process we will need a previously known form format including gap between template image and handwritten text area bounding width and height. From these two information we can counter uniform scaling introduced within image efficiently. Before processing let us assign some of the frequently used terms to a variable.

Let,

$$\mathrm{A_x} = \text{learned top left x} - \text{coordinate pixel gap of handwritten text area}$$

$$\mathrm{A_y} = \text{learned top left y} - \text{coordinate pixel gap of handwritten text area}$$

$$\mathrm{x} = \text{recognized top right x} - \text{coordinate of field}$$

$$\mathrm{y} = \text{recognized top right y} - \text{coordinate of field}$$

In the above assumption A_x is learned by first projecting a straight line from the middle 50% of the template area recognized from the above step, geometric recognition of field. That is, leaving 25% from the top and 25% from bottom of recognized field area. The difference between first foreground pixel value that is being stroked and top right x coordinate of recognized field area is taken as A_x value. With the help of this value we calculate A_y value. The value of A_y changes proportionally with the change in A_x value, value of A_y is calculated with the help of following equation.

Let,

$$\rho_\mathrm{x} = \text{pre} - \text{learned x} - \text{axis gap between template and handwritten text.}$$

$$\rho_\mathrm{y} = \text{pre} - \text{learned y} - \text{axis gap between template and handwritten text.}$$

Then,

$$A_y = \frac{A_x}{\rho_x} * \rho_y \; \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots . (4.6)$$

Now we have all the building blocks we need to finally estimate the handwritten text area within input image for a specific field. It can be obtained with the following equation:

$$x' = x + A_x \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.7)$$

$$y' = y + A_y \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.8)$$

x' and y' represents the estimated handwritten coordinates with respect to recognized field. Each field will have different A_x and A_y values. ρ_x and ρ_y values were learned in the first phase while extracting useful information. An Extracted sample form created by applying above method is shown in the following Figure 4.8.

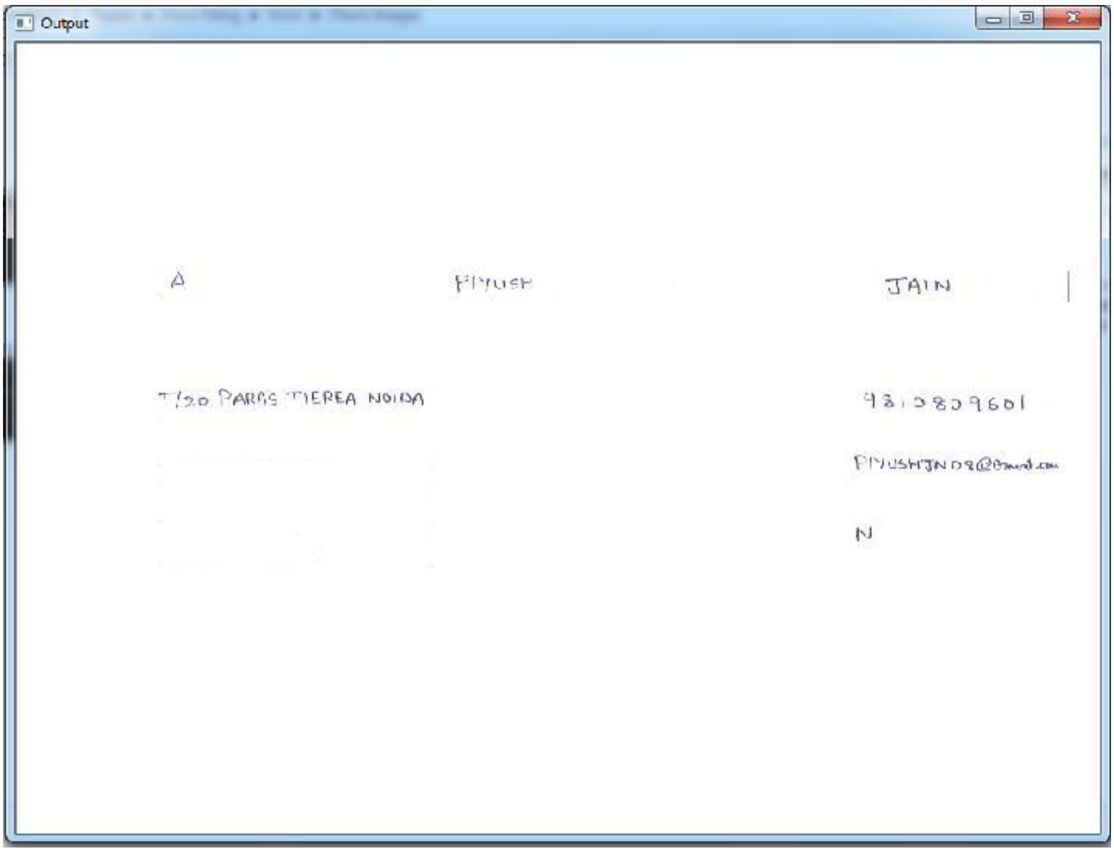
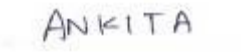


Figure 4.8: Extracted Handwritten Data

4.5 Processing handwritten text

After obtaining the handwritten text area, there might still be some redundant data like lines around text area due to improper handwritten text area estimated in the previous step. The lines found around text area are due to presence of underlines or boxes directing users to write their details in the shown area. This kind of pattern is usually found in every form. To tackle this problem a noble approach of using line detection technique can be used. In our proposed system to detect line Hough line transformation algorithm have been applied.

(a)



(b)

Figure 4.9: Line Removal after Processing Extracted Output

Looking at the above Figure 4.9 (a) it can be observed that the extracted handwritten image had an underline, occurred due to bounding box around it. After applying line removal technique the resultant image is shown in Figure 4.9 (b).

CHAPTER 5

EXPERIMENTAL RESULTS

An application for the proposed system was run and tested in windows 7 operating system. The complete system was made in C with OpenCV library. There were 4 different types of forms taken with 30 instances of each type. Each of them were filled by people in either blue or black color pen. All the forms were filled and then scanned by scanner and saved in their respective directories. A skew of as little as 5 degrees is considered, though there is no restriction on margins introduced on either side of the form. There is no standard method to calculate the accuracy of this system, though an average accuracy of 93.82%.

TABLE 5.1: EXPERIMENTAL RESULTS

Form Type	No. of Samples	No. of Filled Boxes	No. of Bad Results	Bad Rate	Correct Rate	Error Rate(%)
Thapar Registration Form	30	210	8	3.87	96.13	0
Debit Card Form	30	600	62	10.33	89.67	0
Cash Plus Card Form	30	240	12	5.00	95.00	0
Make up Test Form	30	510	28	5.49	94.51	0

Here a result is considered bad if the desired extracted handwritten image contains data of other fields due to poor estimation of coordinates. While a result is considered good if extracted data only contains the desired handwritten data of respective field, and no other handwritten data of other field.

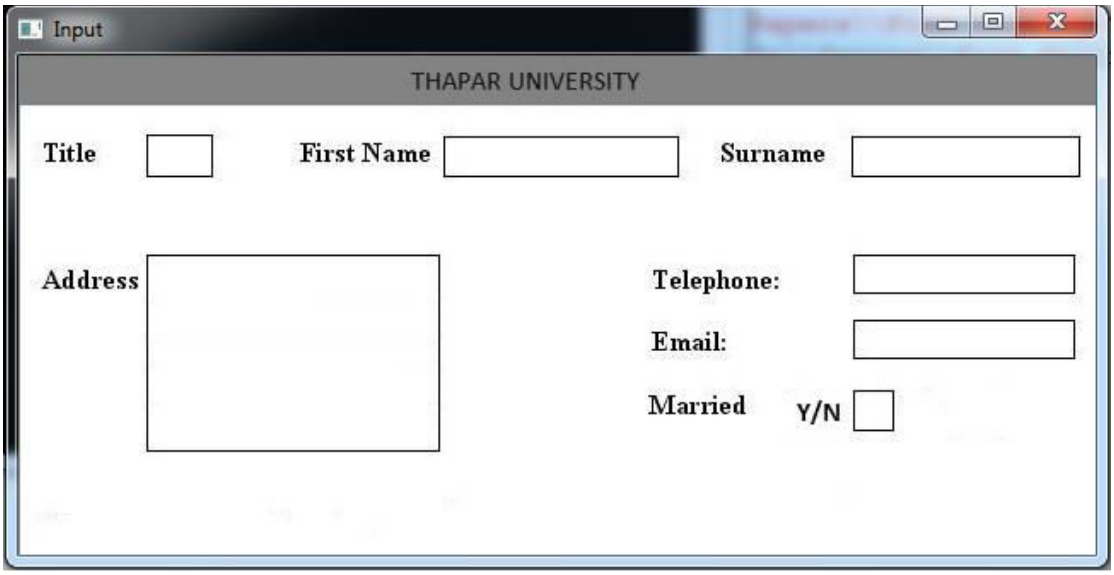


Figure 5.1: Input Image of First Form

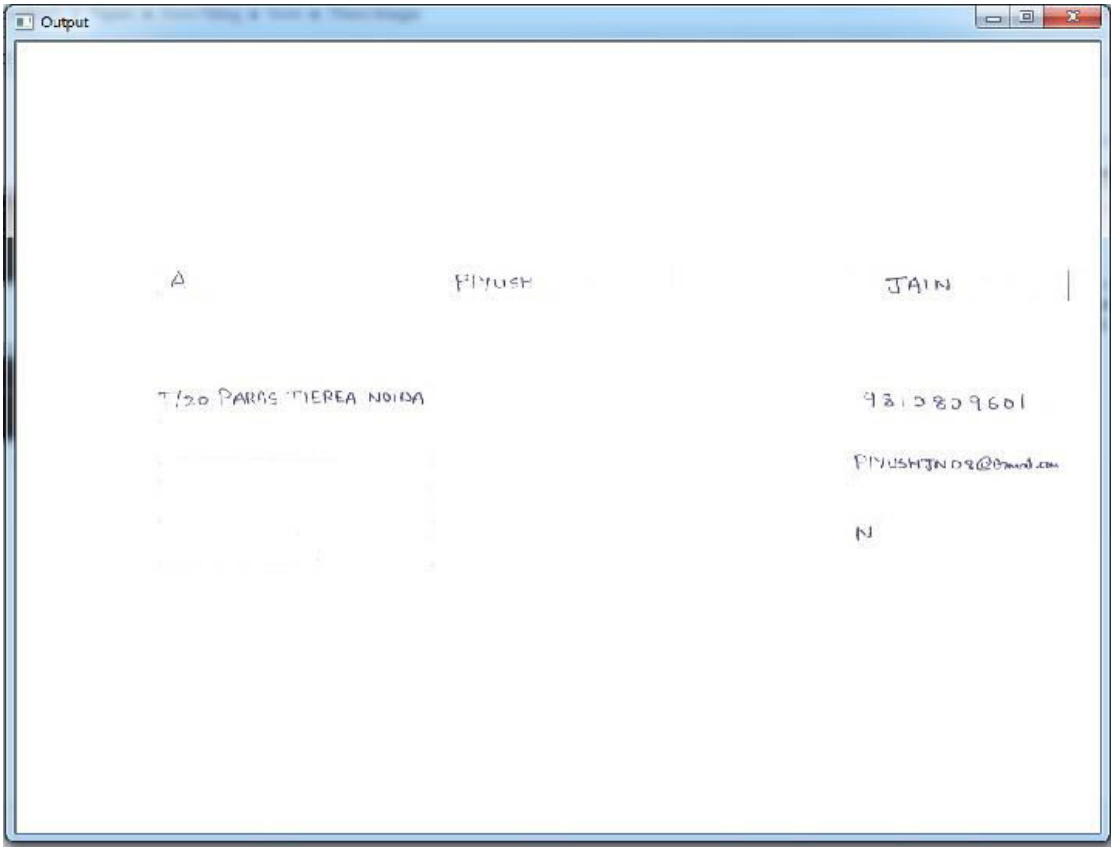


Figure 5.2: Output Image of First Form

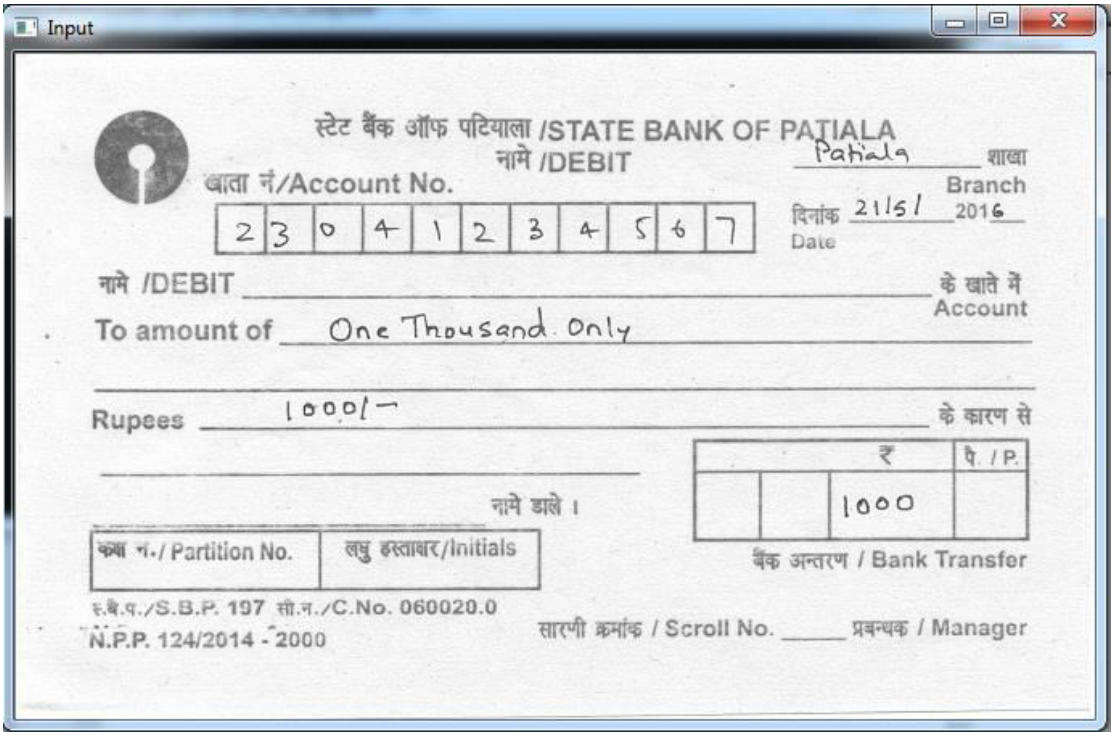


Figure 5.3: Input Image of Second Form

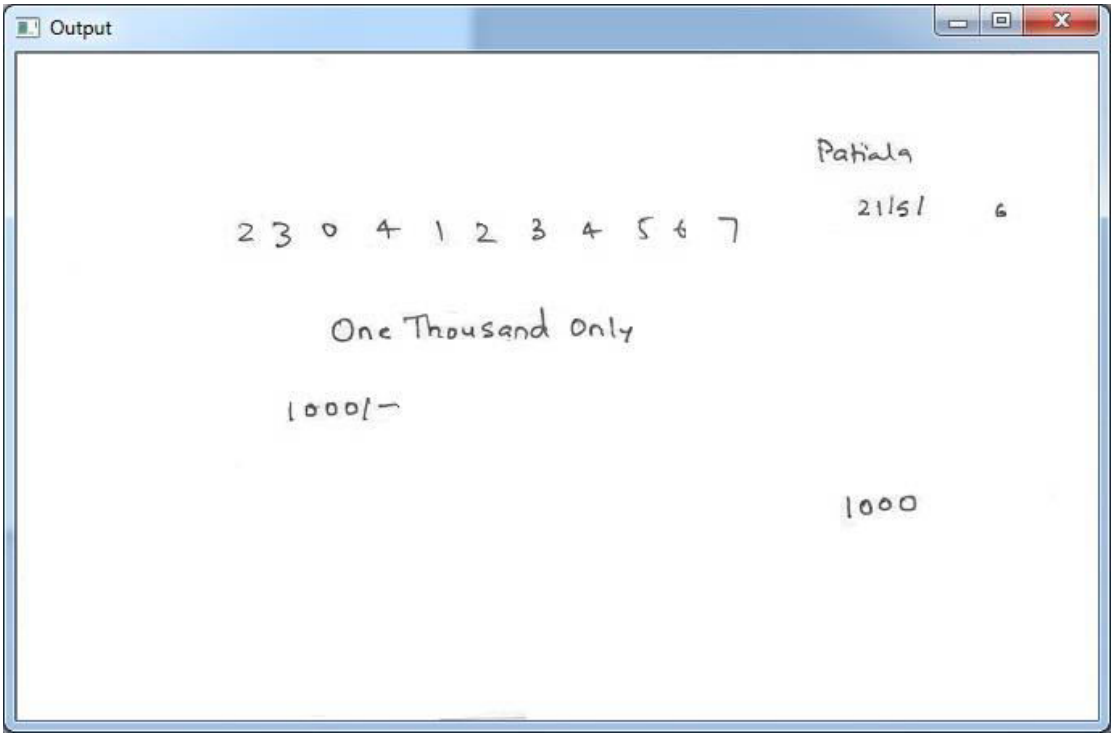


Figure 5.4: Output Image of Second Form

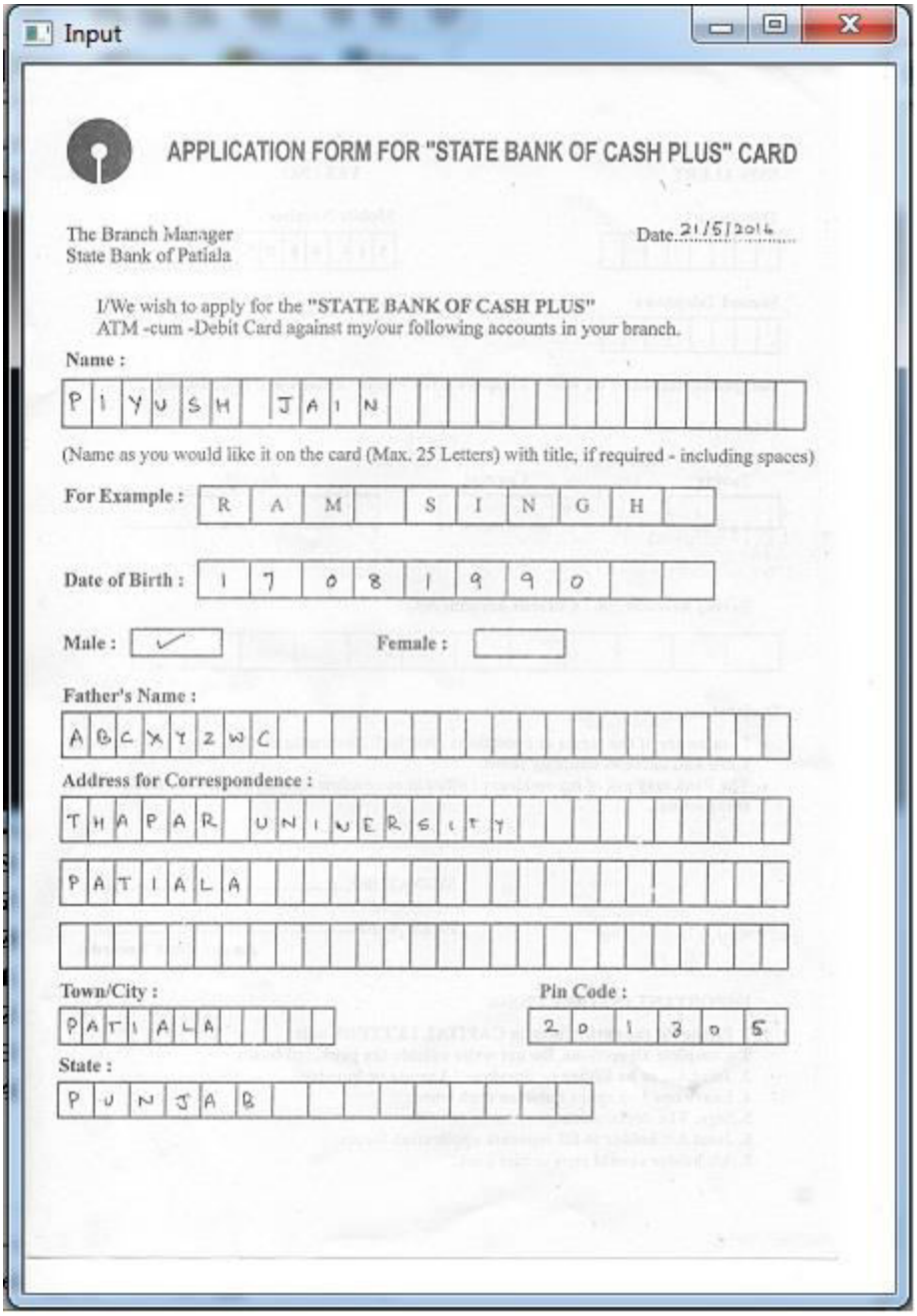


Figure 5.5: Input Image of Third Form

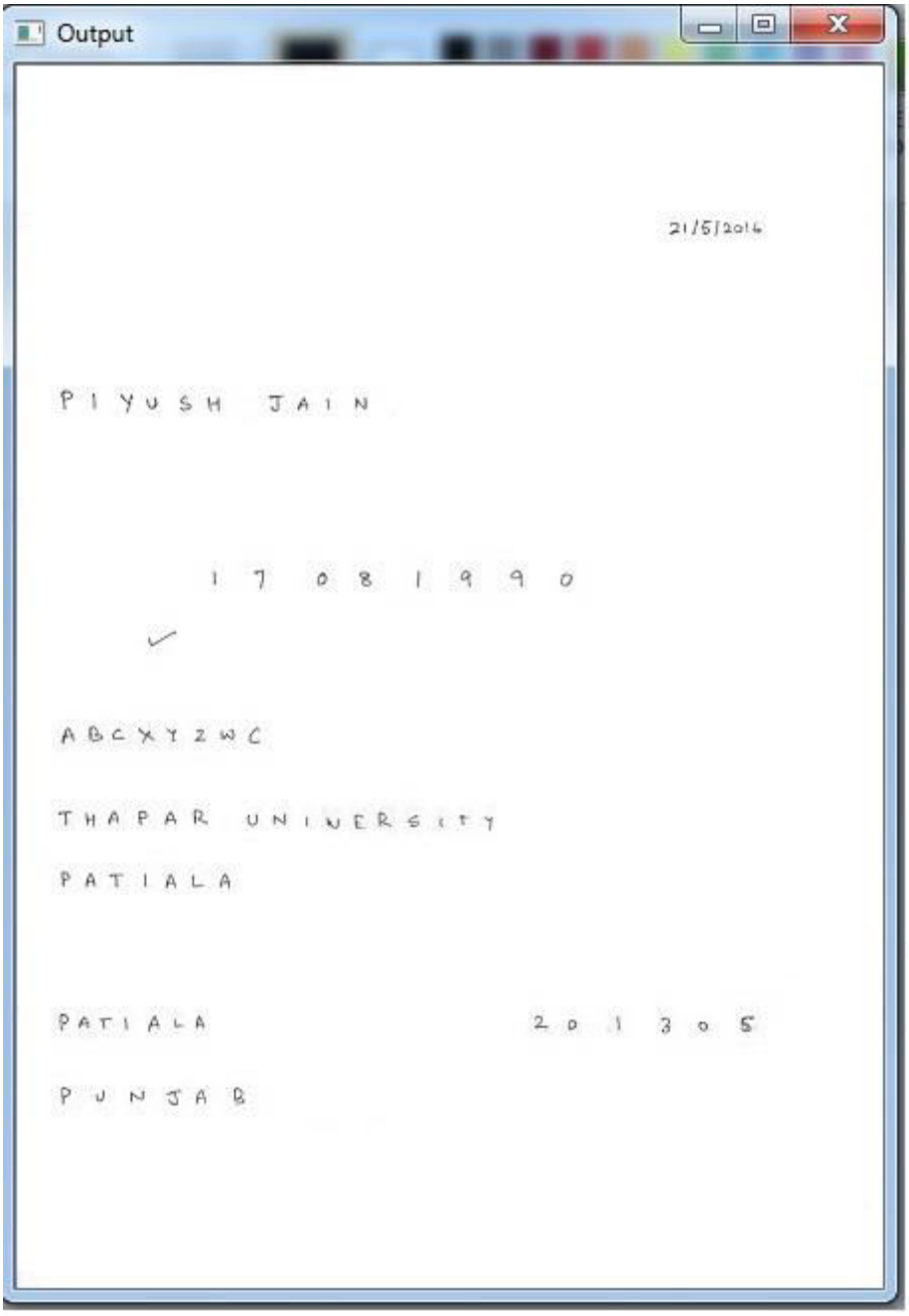


Figure 5.6: Output Image of third form

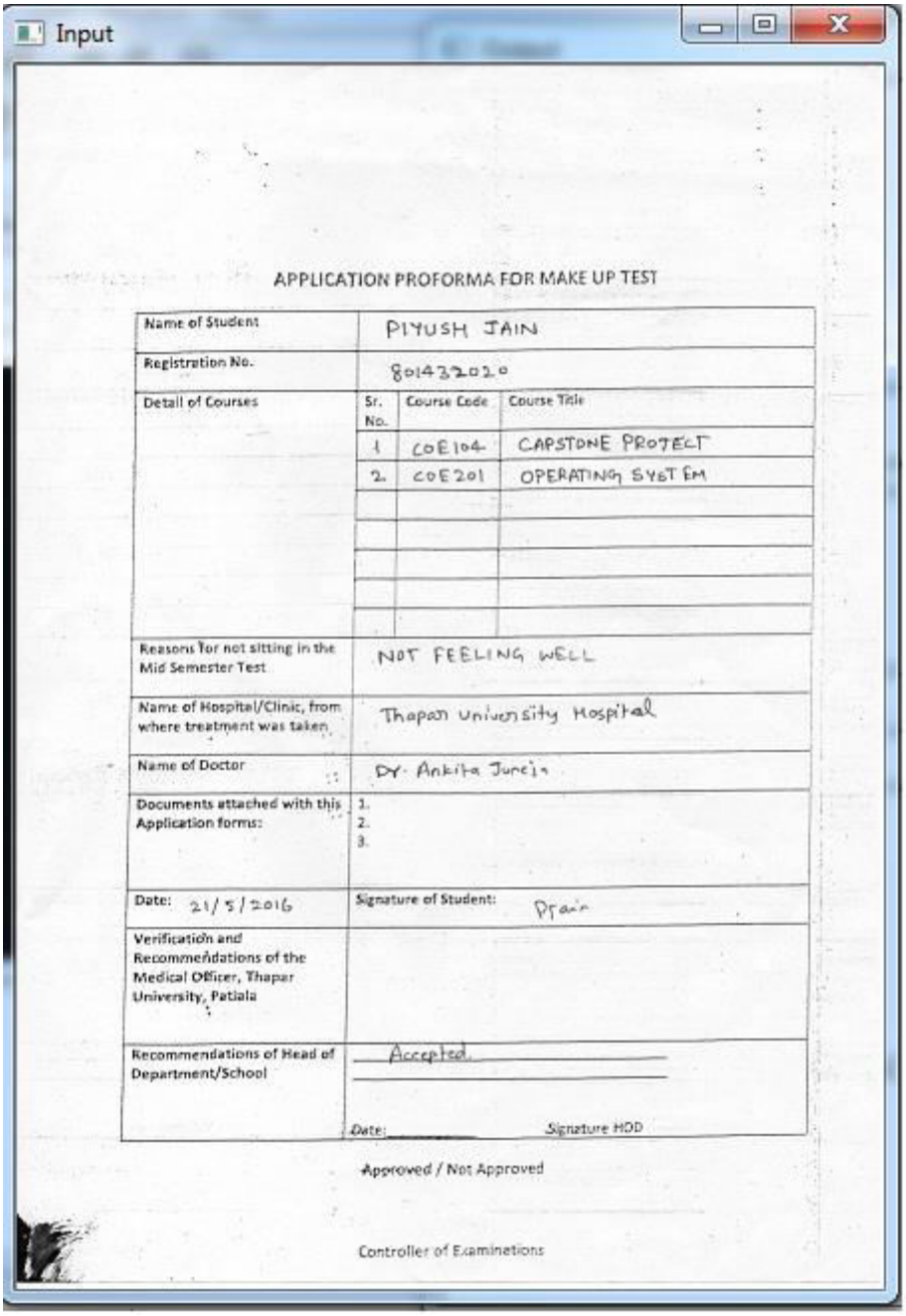


Figure 5.7: Input Image of Fourth Form

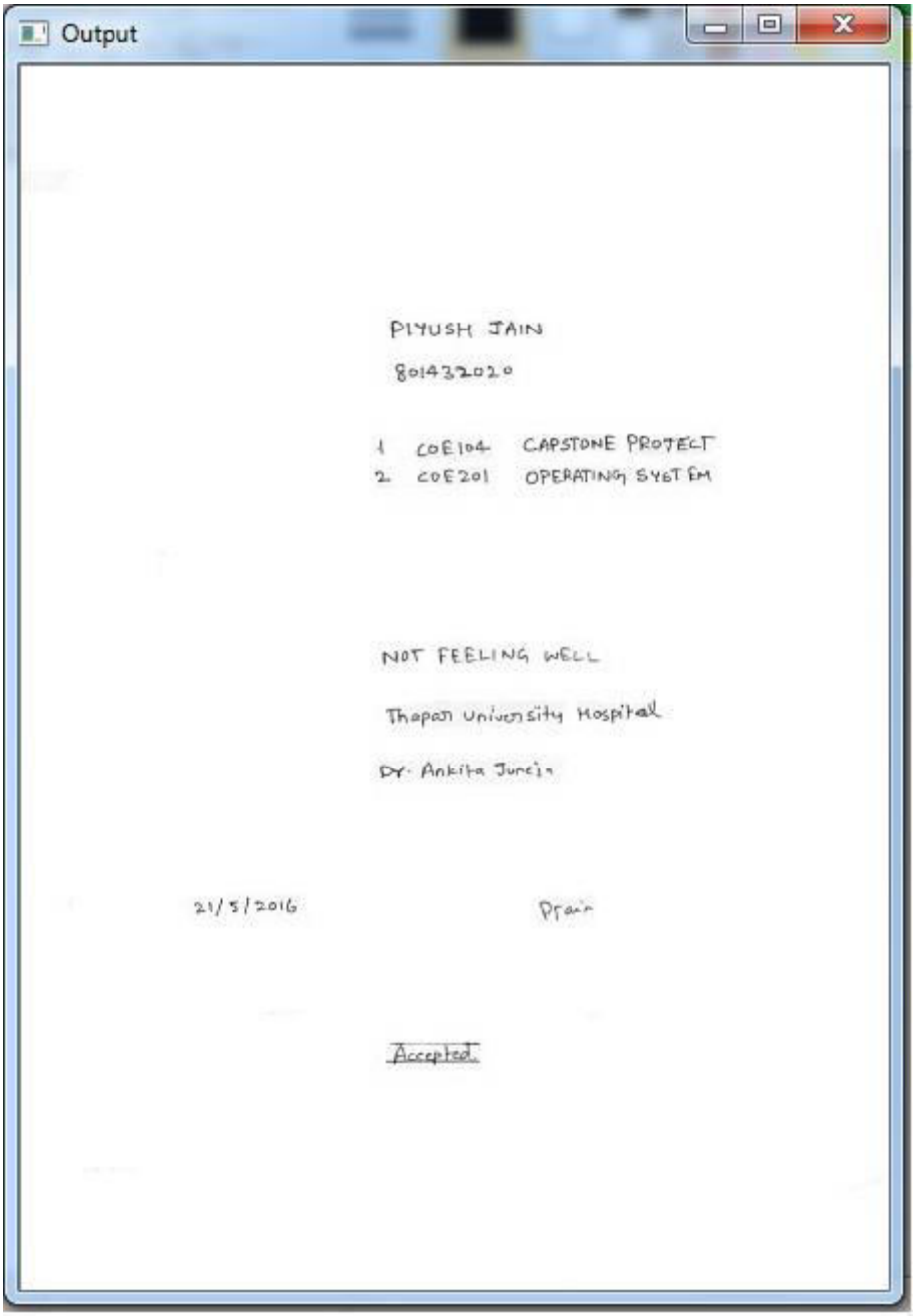


Figure 5.8: Output Image of Fourth Form

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this work, the objective of extracting handwritten text from a filled form and then processing the extracted text with the removal of redundant data has been met successfully. The proposed system can easily deal with various kinds of input field formats like checkboxes, data tables, dotted input, which most of the already known systems found it difficult to extract. The proposed system has been implemented successfully and is rigorously tested to achieve the goals. However, the proposed system could not deal with few issues like a major degree of skew and non uniform scaling within the scanned form. Also, data written outside the indicated area cannot be detected.

6.2 Future Scope

In future, the proposed system may include many new methodologies to handle cases which presently are difficult to handle. This will include a method to handle skewness, non uniform scaling and rotation and improvement in flexibility to handle handwritten text written outside the prescribed area. After handwritten data extraction this system can be equipped further with handwritten text recognition and can be distributed in offices for practical uses.

REFERENCES

[1] U. K. Ngah, N. Ashidi, M. Isa, E. Campus, S. Ampangan, and N. Tebal, "Handwritten Characters Extraction from Form Based on Line Shape Characteristics," *Journal of Computer Science*, vol. 7, no. 12, pp. 1778–1783, 2011.

[2] Z. C. Li, H. Wang, and S. S. Y. Liao, "Numerical Algorithms for Image Geometric Transformations and Applications," *IEEE Transaction Systems Man, Cybernetics Part B Cybernetics*, volume. 34, no. 1, pp. 132–149, 2004.

[3] J. Liu, W. Gui, C. Qing, T. Zhaohui, and Y. Chunhua, "An Unsupervised Method for Flotation Froth Image Segmentation Evaluation Base on Image Gray-level Distribution," *Proceeding Chinese Conference.*, pp. 4018–4022, 2013.

[4] N. Valli, M. V. Lakshmi, A. Vaishnavi, C. Kavitha, and R. S. K. Reddy, "An enhanced noise reduction model for template matching using asymmetric correlation technique," *2015 International Conference in Innovations in Information, Embedded and Communication Systems*, pp. 1–5, 2015.

[5] F. W. Chen, "Archiving and distribution of 2-D geophysical data using image formats with lossless compression," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 1, pp. 64–68, 2005.

[6] Y. Zheng, "An Improved Gray Image Representation Using Overlapping Rectangular Non-symmetry and Anti- packing and Extended Shading Approach," pp. 1863–1867, 2015.

[7] M. Nakanishi and T. Ogura, "Real-time line extraction using a highly parallel Hough transform board," *Proceedings of International Conference on Image Processing*, vol. 2, no. 1, pp. 582–585, 1997.

[8] D. R. Tobergte and S. Curtis, "Edge Following as Graph Searching and Hough Transform Algorithm for Lineament Detection," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[9] H. Fujisawa, Y. Nakano, and K. Kurino, “Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis,” *Proceedings IEEE*, vol. 80, no. 7, pp. 1079–1092, 1992.

[10] J. Llados, J. Lopez-Krahe, and E. Marti, “Hand drawn document understanding using the straight line Hough transform and graph matching” *Proceedings of International Conference Pattern Recognition*, vol. 2, pp. 497–501, 1996.

[11] R. Safari-foroushani, “Form registration : A computer vision approach”, 1998.

[12] A. Zampirolli, “Fast Multidimensional Parallel Euclidean Distance Transform based on Mathematical Morphology,” pp. 100–105, 2001.

[13] B. V. Dhandra, V. S. Malemath, H. Mallikarjun, and R. Hegadi, “Skew detection in binary image documents based on image dilation and region labeling approach,” *Proceedings - International Conference on Pattern Recognition*, vol. 2, pp. 954–957, 2006.

[14] L. Guo, G. Zhang, and J. Wu, “Infrared image area correlation matching method based on phase congruency,” *Proceedings - International Conference on Artificial Intelligence Computational Intelligence AICI 2010*, vol. 1, no. 1, pp. 488–491, 2010.

[15] I. Noda, H. Shimora, and H. Akiyama, “Flexible Framework to Maintain Multiple and Floating Coordinate Systems,” pp. 447–451, 2008.

[16] R. Roncella and E. Romeo, “Comparative Analysis of Digital Image Correlation Techniques for In-plane Displacement Measurements,” pp. 721–726, 2012.

[17] H. Torp, X. M. Lai, and K. Kristoffersen, “Comparison between cross-correlation and auto-correlation technique in color flow imaging,” *Ultrasonics Symposium 1993. Proceedings., IEEE 1993*, pp. 1039 –1042 vol.2, 1993.

[18] X. G. Wang, F. C. Wu, and Z. H. Wang, “Harris correlation descriptor (HCD): A novel descriptor for point matching,” *Proceedings. - International*

Conference on Computer Science and Software. Engineering CSSE 2008, vol. 2, no. 1, pp. 1154–1157, 2008.

[19] Y. Zhang and K. Zhou, "Study on Automotive Style Recognition with the Image Erosion Technology," pp. 4438–4441, 2011.

[20] I. Aksak, C. Feist, V. Kiiko, R. Knoefel, V. Matsello, V. Oganovskij, M. Schlesinger, D. Schlesinger, and G. Stanke, "Extraction of Filled-in Data from Colour Forms." *Computer Analysis of Images and Patterns, 7th International Conference,* 1997

[21] J. Chen, "An Efficient Algorithm for Form Structure Extraction Using Strip Projection," *Pattern Recognition*, vol. 31, no. 9, pp. 1353–1368, 1998.

[22] Casey et al,R. Ferguson, "Computer Implemeted Method for Automatic Extraction of Data from Printed Forms",1992.

[23] M. B. Hisham, S. N. Yaakob, R. a a Raof, a B. a Nazren, and N. M. Wafi, "Template Matching Using Sum of Squared Difference and Normalized Cross Correlation," pp. 100–104, 2015.

[24] K. Jain, "Generic System for Form Dropout," vol. 18, no. 11, pp. 1127–1134, 1996.

[25] I. C. Paschalidis, K. Li, and R. M. Estanjini, "An actor-critic method using Least Squares Temporal Difference learning," *Conference on Decision and Control*, pp. 2564–2569, 2009.

[26] I. Washington, J. Lilling, and W. Zitomer, "Data Processing System and Method for Field Extraction of Scanned Images of Document Forms" pp. 2–5, 1989.

[27] Y. Zhu, "Augment document image binarization by learning," *19th International Conference on Pattern Recognition*, pp. 1–4, 2008.

[28] X. Kang, W. Zhu, K. Li, and J. Jiang, "A Novel Adaptive Switching Median filter for laser image based on local salt and pepper noise density," *PEAM 2011*

- *Proceedings 2011 IEEE Power Engineering and Automation Conference*, vol. 3, pp. 38–41, 2011.

[29] R. Safari, N. Narasimhamurthi, M. Shridhar, and M. Ahmadi, "Document registration using projective geometry," *IEEE Transaction on Image Processing*, vol. 6, no. 9, pp. 1337–1341, 1997.

[30] K. Ban, J. Lee, H. Hwang, and K. Chung, "Face image registration methods using Normalized Cross Correlation," *2008 International Conference on Control Automation and Systems*, pp. 2408–2411, 2008.

[31] C. Rader, "Generating Rectangular Coordinates in Polar Coordinate Order," *Streamlining Digital Signal Processing: A Tricks Trade Guidebook Second Edition*, November, pp. 407–412, 2012.

[32] J. S. V, "Wavelet based quality enhancement for medical images," pp. 277–280, 2011.

[33] L. Yuan and J. Sun, "High quality image reconstruction from RAW and JPEG image pair," *Proceedings in IEEE International Conference Computer Vision*, pp. 2158–2165, 2011.

[34] J. Pradeep, E. Srinivasan, and S. Himavathi, "Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System using Neural Network," *Journal of Computer Science*, vol. 3, no. 1, 2011.

LIST OF PUBLICATIONS

Piyush Jain and Dr. Shivani Goel, "Filled-in Handwritten Data Extraction from Documents based on Geometrical Features" International Conference on *Micro-Electronics and Telecommunication Engineering*.(ICMETE-2016) [Communicated].

VIDEO LINK

https://www.youtube.com/channel/UCkon0R-2_Aw1wu5aOYs_bJQ