# Data Extraction from Hand-filled Form using Form Template

Rohit Sachdeva
Department of Computer Science
Multani Mal Modi College,
Patiala, India
*rsachu.147@gmail.com*

Dharam Veer Sharma
Department of Computer Science
Punjabi University,
Patiala, India
*dveer72@hotmail.com*

***Abstract-*** Database is very vital for taking the day to day decision and in the long run it helps in formulation of policies, strategies of an organization. Numerous efforts, time and money are spent to get, store and process the data. To get the data from a user, an interface is designed which is known as form. The forms may vary from paper based to online. Manually processing paper based form is prone to errors. Therefore, it will be useful to deploy automated systems for reading data from paper based forms and storing it in the database. Further, this data can be modified, processed and analyzed. In this paper, we have proposed a method to extract data from hand-filled pre-designed form based on form templates.

***Keywords-****Data Extraction, Hand-filled form, Form Tempalte, Color Drop*

*****

## I. INTRODUCTION

In daily routine, forms are used to get data from users. Form is a user interface for data collection. In many offices various types of form documents are processed for collecting data. The data obtained through the forms is a static representation of handwriting. Scanning of these forms will only produce an image copy of the document. That copy may be displayed on the screen and printed, but that cannot be changed or re-formatted. These forms are manually entered in the system, which is time consuming and arduous task. Moreover, it also puts a lot of strain on the financial resources of the organization. Further, it gives rise to many non-sampling errors.

Requirement of the hour is to devise an automated solution for extraction and recognition of data from paper based forms. Taking due consideration of the facts an algorithm has been developed, extracts the handwritten data from paper based form. Extracted data may be further converted into an editable form. As compared to manual data feeding system, proposed system reduces the number of human resources, work at maximum speed, less error-prone, gives better performance and more reliable. The application areas include: railway reservation, census data collection, banking system, tax payments, postal systems, educational institutes, office automation etc.

## II. REVIEW OF LITERATURE

A general system for extraction and cleaning of data from handwritten forms has been proposed by Ye et al.[1]. The items of interest are located from the form for which a model template is generated from a blank form, which is used to remove the form frame from the actual forms to be used for recognition. To clean the handwriting touching the pre-printed text morphological operations based on statistical features are used. Authors reported 95.5% of recognition rate.

Sako et al.[2] have proposed a form reading technology based on form type identification and form-data recognition. A recognition rate of 97 % has been reported by the author.

An algorithm for removal of the field frame boundary of the hand filled forms in Gurmukhi Script proposed by Sharma et al.[3]. In their paper authors discussed about the characteristics of Gurmukhi script such as use of headline and varied writing styles and also discussed problems related to it, such as filled data may overlap or get merged with the field frame boundaries. A novel approach has been proposed to remove the form field frame boundary, while preserving the data contained therein.

While writing in the form, a person may write within a box or outside the prescribed box. The form field line removal methods extract the data only from the prescribed box. There is need to overcome this problem of extracting the words, even outside the boundary area (box) of the form. In form recognition, frame line detection is a vital and difficult step. Existing methods of form field line removal in roman script are given in[4-13]. Two most common methods used for line detection are Hough transform discussed by Illingworth et al.[4] and vectorization by Liu et al.[5].

Simoncini et al.[6] developed a system in which a set of regions of each edge are extracted and a standard line fitting method is used to parameterize them. The deletion of the lines is carried out, leading to an excessive erosion of the crossing strokes. At the end, they must be repaired and the crossing characters re-constructed.

Hough transforms method has great advantage that it can detect dashed or broken lines. However, it is not applied in form recognition as it is too slow. Generally in forms, the frame lines are horizontal and vertical. Liu et al. [7] and Chen et al.[8] are used, fast modified Hough transforms method which are just projection methods. These methods have the drawback that they are unable to detect diagonal lines and frame lines with large skew angles. The projections of frame lines are overwhelmed in the projection of characters and cannot be correctly detected, when the characters are merged or overlapped with the frame lines. Vectorization method, which is bottom up approach used by Liu et al.[5] can solve the problems of projection methods. Firstly, vectors are extracted from images. The whole objects are detected by merging all the extracted vectors.

Yoo et al. [9] tested and analyzed their proposed system on a large number of real handwritten form samples and concise out 13 types, 34 subtypes overlapping modes formed by Korean characters overlapping with frame lines. Different techniques are used to deal with every overlapped mode whenever it is detected. This method is very tedious, and cannot cover all

 ISSN: 2321-8169
5311 - 5317

overlapping modes. Moreover, it is more difficult to detect overlap mode in form document which contains noise.

In most of the literature, frame line detection procedures depend on a critical threshold representing the character size. This threshold value is a constant value as per Liu et at.[7] and Chen et al.[8] and it is input by users as assumed by Pan[10].

Zheng et al.[11] used vectorization method by using novel image structure element named "Directional Single-Connected Chain (DSCC)", as the element vector. Most of the frame lines are detected correctly by merging DSCCs under some constraints and it can also solve most types of character-line crossing problems but the pseudo lines and the broken lines cannot detect correctly. Vectorization methods are used the large number of vectors so these methods are much slower than projection methods. Shimamura et al.[12] used the erosion and dilation method of removing field frame lines. This method is not possibly practical due to variation of thickness in handwritten data and in some cases it may be thinner than the frame boundaries.

These methods are not suitable for Indic scripts such as Devanagari, Gurumukhi scripts where characters are connected with the headline. Since, if the headline is merged with the field frame boundaries, then removal of field frame boundaries will inadvertently remove the headline from the text and produce wrong recognition results.

## III. PROPOSED METHOD OF DATA EXTRACTION

### A. *Form Designing*

Before designing forms, determines which types of data and how many various parts of data required. Then the sequence from one field to the next field, for example, in personal information first field should be name and the second field must be father's name. That data must be captured in the fields of the form in the logical arrangement. These fields must have their captions on a form which helps in entering data in the appropriate fields. Two blocks of square shape are used for reference points for marking the starting and ending of form. For the starting reference point, on the left-top corner square is placed and for ending reference point on the right bottom corner square is placed as shown in figure 1. Once these two reference points are identified, then the relative distances of other fields from the starting reference point are used in measuring the absolute coordinates of the fields on the forms.

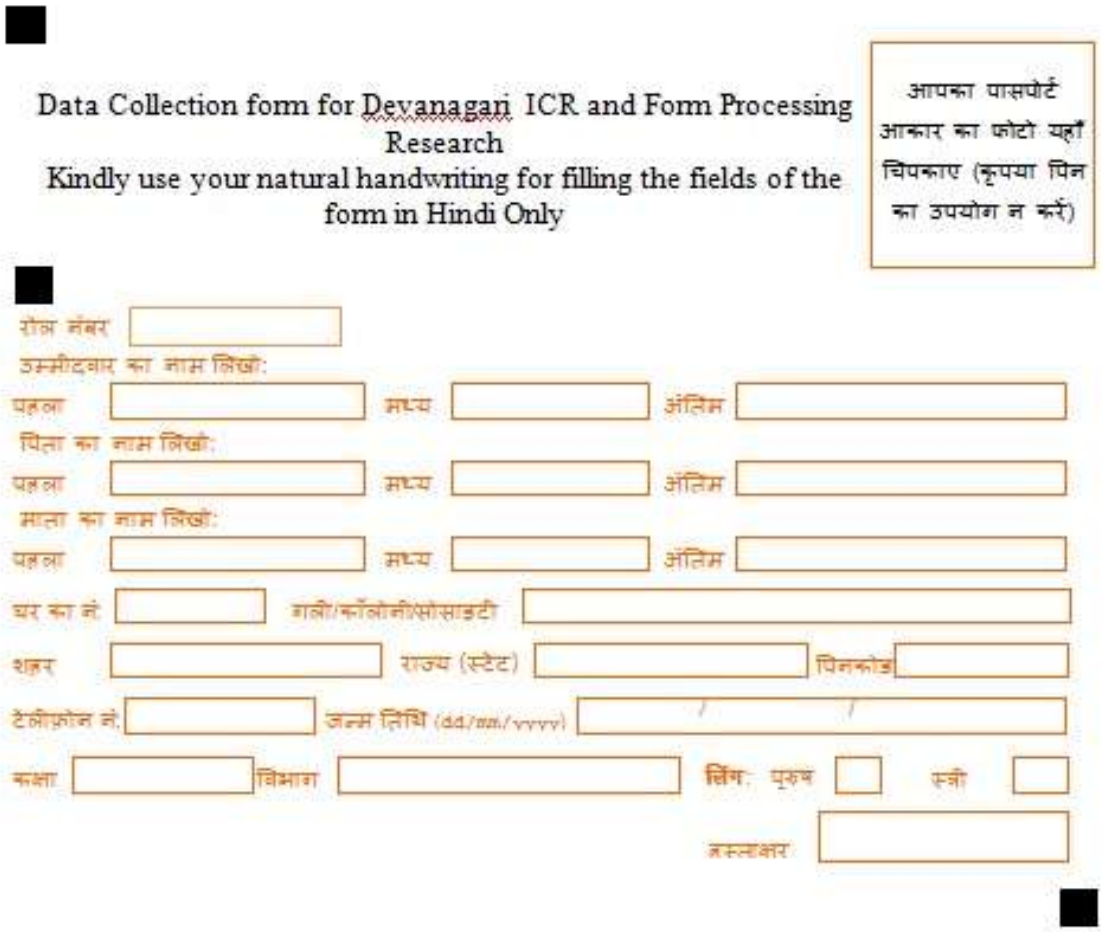



Figure 1-1: A sample of designed form

### B. *Collection of data*

The sample data collection form collects data about student and covers different types of data to be supported by the system. A total of 235 forms of the same type have been used. Each form consisted of 22 fields (figure 1.1). The forms were filled by different students with their natural handwriting one of the sample as shown in figure 1.2.

 ISSN: 2321-8169
5311 - 5317

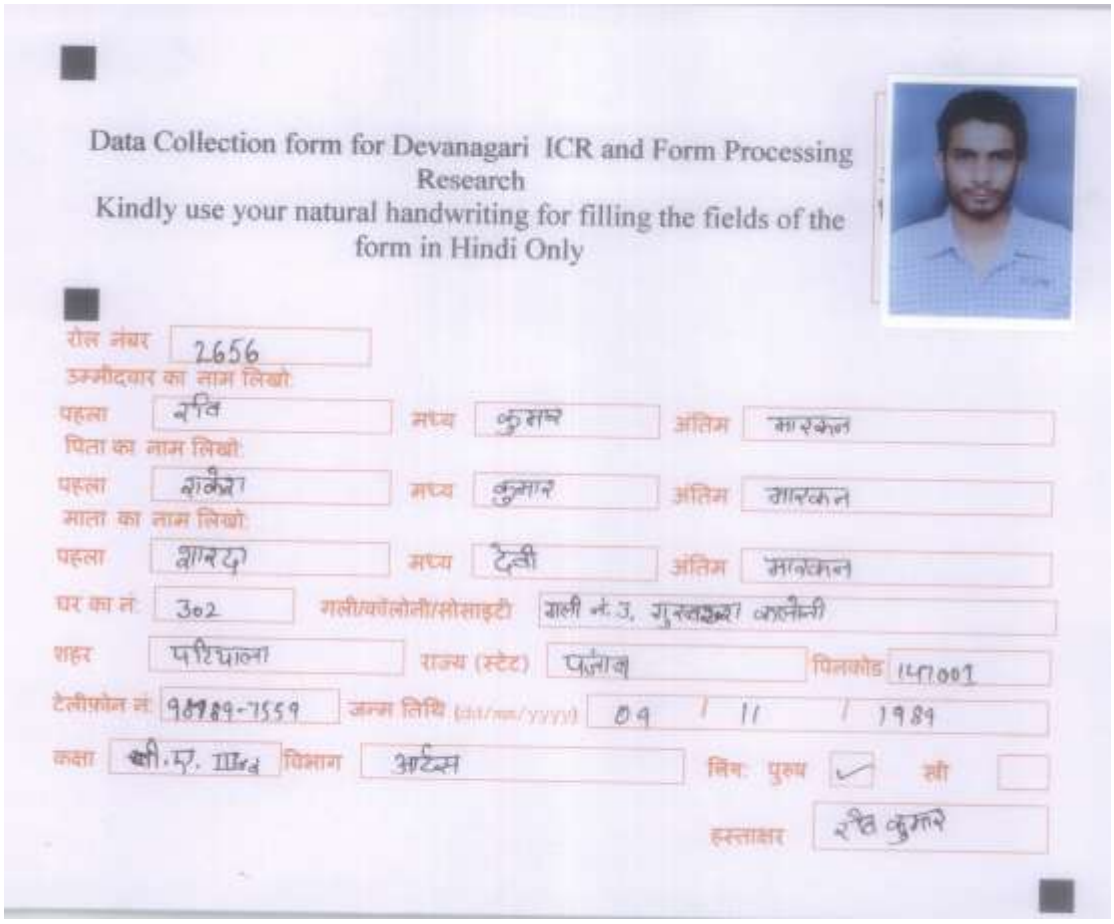



Figure 1-2: A sample of filled form

### C. *Template Generation*

Generating a form definition is backbone of the present form processing systems. The form contains the field captions which are printed on the form and data areas where the user writes the data in their handwriting. From a blank form a template is created which is used to separate the pre-printed matters i.e. captions and data areas. This template helps to remove the form frame from the actual forms to get the required printed data, such as barcode and handwritten data used for recognition.

For customized form processing a new system has to be developed. The paper presents a method to extract data from hand-filled pre-designed form (Figure1.1) based on the form template.

The recognizable fields of the form are highlighted to generate a form template as shown in figure 1.3. A definition of the form is generated by using the form template. It contains the information about starting and ending reference points for skew assessment, number of fields, fields' data types their sequences and locations, for validation of numeric field (where applicable) its domain registration, for post processing contextual dictionaries applicable on text fields. The data types, considered for the system, are given in Table 1.1 and have been marked accordingly in the Figure 1.3.

Table 1-1: Table of data types used in the system

| Sno | Data Type Name | Purpose |
|---|---|---|
| 1. | Numeric | For storing digits only as in case of age, pin code, phone number etc. |
| 2. | Alpha | For storing data consisting of alphabets only like parts of name, city, state etc. |
| 3. | Alphanum | For storing data consisting of alphabets and numbers like house number, class etc. |
| 4. | Date | For storing date type data e.g. date of birth, date of joining, date of purchase etc. |
| 5. | Choice | It is used where multiple options are available as in case of gender which can be (1) male or (2) female. |
| 6. | Picture | For sub-images in the form document, like photographs, signatures, barcodes etc. |

In Figure 1.3, starting and ending reference points (refereed as service fields) is marked as 0 fields, which are also used for detection of form level skewness and form's top and bottom. For every new type of form, a new form definition is generated. It is a one-time process and stored in a file. It can be reused for further modification purpose. At the time of form definition, all the validation sources are provided such as a domain for numeric fields, dictionaries for alphabetic fields etc. During the post processing these sources are used to validate data. The system is flexible that provides the facility of creating and dynamically adding new dictionaries.

 ISSN: 2321-8169
5311 - 5317

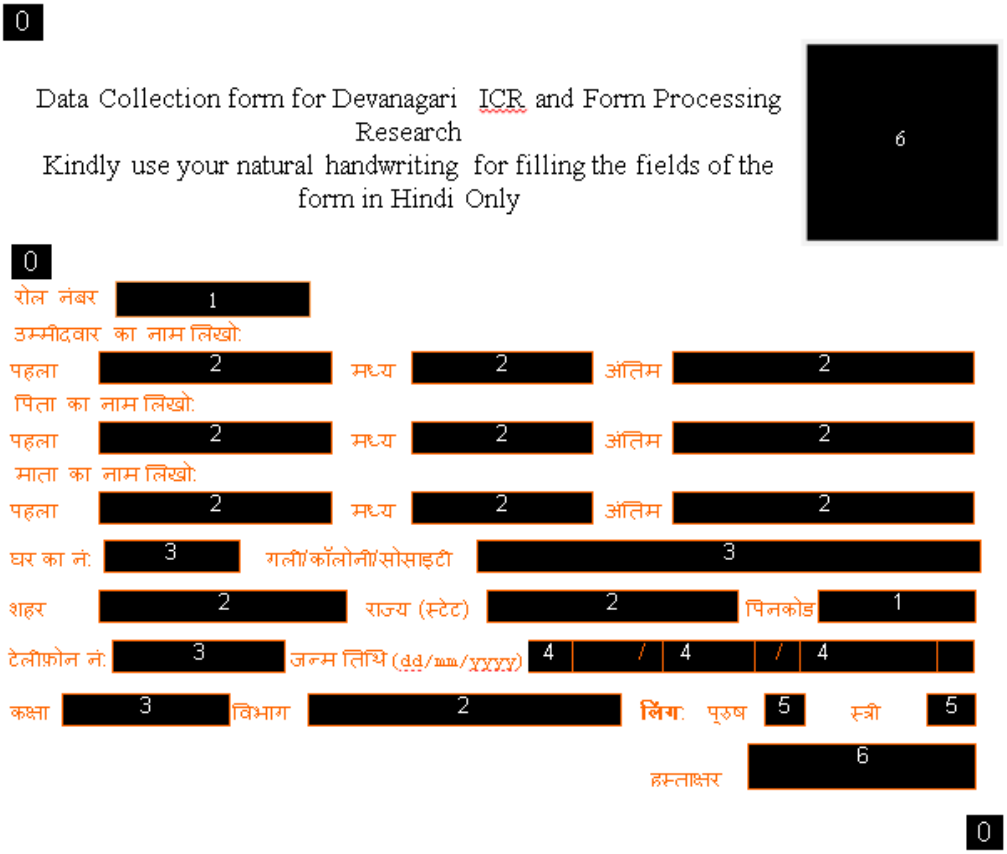



Figure 1-3: Template of a sample form with field types marked on it

A module has been developed, to detect the fields on the form by using a form template image which interactively generates the form definition (Figure 1-4).

Positions of the fields on the form are calculated and user is required to supply information regarding type of field, name of the field, dictionary to be used for post-processing in case of alphabetic fields, range of values for numeric fields. Picture fields (photograph and signature) are the sub-images which are extracted and stored separately.


Figure 1-4: Screen short of module while genrating form defination

*D. Digitization*

Digitization means scanning the original paper based form document and storing it as a digital image. Brightness, contrast and scanning resolution measured in terms of dots per inches are the key factors while digitization of paper based form. In the present system, the forms have been scanned with 300 dpi resolution, the 100 threshold for brightness and 100 for contrast (these values have been

 ISSN: 2321-8169
5311 - 5317

computed after experimentation) which is without any distortions.

*E. Form Level Skew Correction*

The form image may get skewed during scanning of form documents. This may happen because of improper alignment of paper on the scanner. It results in the wrong alignment of text on the form document image. Therefore, before the data extracted from the form, skewness of the forms are checked and removed by using skew correction.

As discussed earlier in form designing, reference points for marking the starting and ending of forms were placed. These points are located and their distance is calculated and compared with the template form's starting and ending reference point's distance. If any deviation is found between them, then it indicates skew in the form image. This is used for the calculation of skew angle. To save the time, the skew is only detected and correction is deferred till the time of field data extraction.

The skew parameters XFactor and YFactor, calculated by using a technique given by Sharma et al.[13], are used for filtering the actual location. By using the following formulae we get the new coordinates of the rectangle of the field.

newrect.left = rect.left + rect.top/XFactor
newrect.top = rect.top + rect.left/YFactor
newrect.right = rect.right + rect.bottom/XFactor
newrect.bottom = rect.bottom + rect.right/YFactor

However, form images having large skew, which is detected based on the values of XFactor and YFactor, are not corrected but only detected. If the value of XFactor and YFactor are lesser than 12% (chosen after experimentation) of the value of the image height and image width respectively, the form image is highly skewed and is not processed.

*F. Field data extraction*

In this step, from the form, fields are traced and their boundaries are removed and data are extracted from it. The boundaries of the fields are removed by using various techniques such color dropout, form template, removing boundary lines, etc. In monochrome forms, by continuously removing the boundary lines, boundary of the field is removed. While in colored form, the boundaries are removed by dropping the color, which is used while designing the form fields.

In the proposed method, to extract the hand filled data from the form, location of fields are located by using the coordinates of designed form fields stored during the form template generation as shown in Figure 1.3.

Except the picture fields this is applied to a rectangular area of each field. The first step is to store picture fields as an image in the database, it is directly extracted and stored. The second step is to find the correct location and the field size on the form. The size of the actual bounding field rectangle may be more than identified in the form definition phase because of overlapping of filled data with the field boundaries (Figure 1-5). The actual rectangular of the field is identified by finding the bounding rectangle of the field.

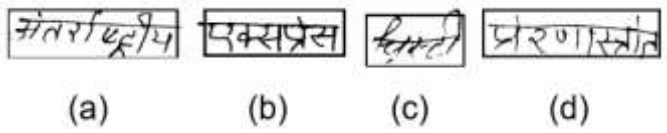



Figure 1-5: Examples of overlapping on all four sides

With the physical location of field of template form, we are able to locate the fields on the form. To locate the field, the starting and ending reference point plays a key role. To rectify the value of field location of the source form, the difference between the value of starting reference of template form and source form calculated and processed. After finding the location of the field next step is to extract data from source form.

Sharma et al.[3] proposed a method for form field frame boundary removal for form processing system in Gurmukhi script by using some assumptions. This method removes the form field frame boundary, while preserving the data contained therein. The drawback of the technique is that if the word of the field contains a character without a headline, like ਖ, ਪ, ਮ etc. then the headline is added to such characters and these can be wrongly recognized. Another problem is overlapping of filled data with the field caption of the form.

By using the color drop method we can eliminate these problems in color forms. Color drop is a better alternative than the form field frame boundary removal in color forms. In this method, forms field captions and boundaries are printed using lighter tones of any some specific color, usually Red as per the convention. The user can fill data using a dark color pen (Blue or Black), other than Red. By dropping the caption color, which in this case is Red, the form field boundary is removed. The main advantage of using color dropout is that the field boundary does not get mixed up with the data which results accurate data extraction. This method is a little costly as compare to form filed boundary removal because in this case the form must be colored printed. Though, its cost is more, but the advantage derived over weighs the cost involved.

## IV. Results

A colored form is designed which contain 22 fields as shown in figure 1.1. A total of 235 students filled forms with their natural handwriting (one of the samples is shown in figure 1.2). A template of the form is generated as shown figure 1.3. Then these forms are scanned by using resolution at 300 dpi, threshold at 100 and contrast at 100. By using the color drop method, red color is dropped and the output of that is shown in figure 1.6.

 ISSN: 2321-8169
5311 - 5317

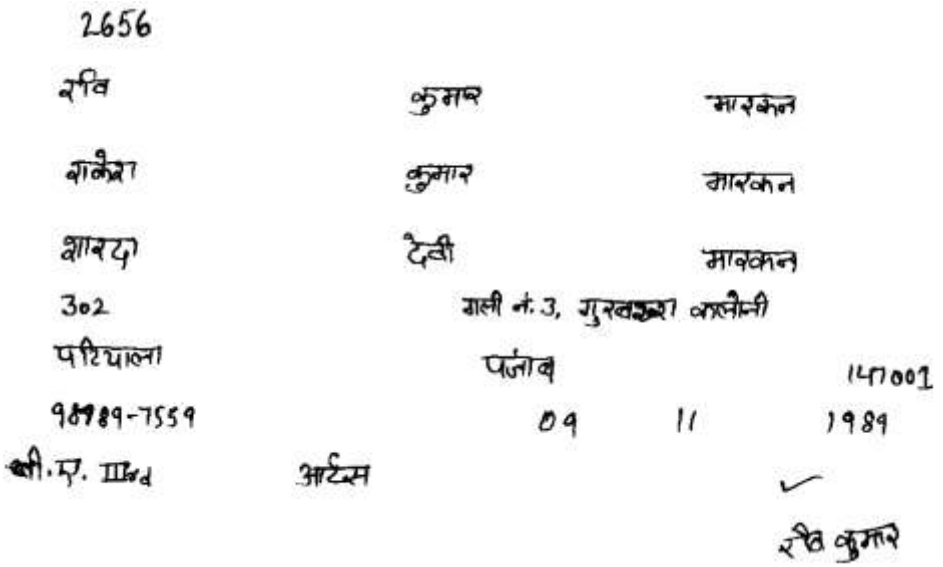
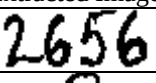
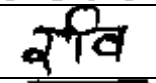
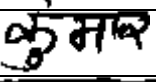
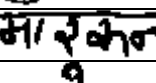
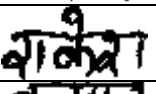
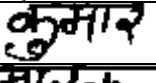
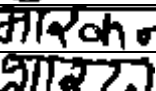
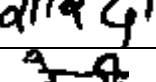
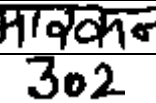
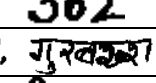
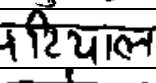
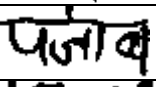
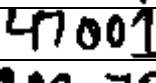
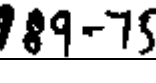



Figure 1-6: Form after color drop

Then by using template matching method data is extracted. Picture fields are the sub-images which are extracted and stored as images. For other data types fields, field positions are located by using template matching method and fields are extracted and stored as images as shown in Table 1.2.

| Field name | Extracted Image |
|---|---|
| Roll No |  |
| Student's First Name |  |
| Student's Middle Name |  |
| Student's Last Name |  |
| Father's First Name |  |
| Father's Middle Name |  |
| Father's Last Name |  |
| Mother's First Name |  |
| Mother's Middle Name |  |
| Mother's Last Name |  |
| House No |  |
| Address |  |
| City |  |
| State |  |
| Pin code |  |
| Telephone |  |

| Field name | Extracted Image |
|---|---|
| Date of Birth Date |  |
| Month |  |
| Year |  |
| Class |  |
| Department |  |

Table 1-2: Results of data extraction from hand-filled form

## V. Conclusion

This paper proposes a method for data extraction from the hand-filled form by using color drop and template matching methods. For this purpose colored form is designed with caption written in red color. We obtained encouraging results and extracted data fields are further used as the input for the segmentation process. .

## References

[1] X. Ye, M. Cheriet, C. Y. Suen, "A Generic System to Extract and Clean Handwritten Data From Business Forms", in the Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition (IWFHR), pp 63-72, 2000.

[2] H. Sako, M. Seki, N. Furukawa, H. Ikeda, A. Imaizumi, "Form Reading Based on Form-type Identification and Form-data Recognition", in the Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), pp. 926–930, 2003.

[3] D.V. Sharma, G. S. Lehal, "Form Field Frame Boundary Removal for Form Processing System in Gurmukhi Script", in the Proceedings Of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 256-260,2009.

[4] J. Illingworth, J. Kittler, "A Survey of the Hough Transform", Computer Vision, Graphics & Image Processing, vol.44, pp.87-116, 1988.

[5] W. Liu, D. Dori, "From Raster to Vectors: Extracting Visual Information from Line Drawings", Pattern Analysis & Application, No.2, pp.10-21, 1999.

[6] L. Simoncini, V. Kovacs, M. Zs, "A System for Reading USA Census '90 Hand-Written Fields", in the Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 86-90, 1995.

[7] J. Liu, X. Ding, Y. Wu, "Description and Recognition of Form and Automated Form Data Entry", in the Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR), pp. 579-582, 1995

[8] J. L. Chen, H. J. Lee, "An Efficient Algorithm for Form Structure Extraction Using Strip Projection", Pattern Recognition, Vol.31, No.9, pp.1353-1368, 1998.

[9] J. Y. Yoo, M. K. Kim, S. Y. Han, Y. B. Kwon, "Line Removal and Restoration of Handwritten Characters on the Form Documents", in the Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 128-131, 1997.

[10] S. Pan, "Research and Realization of a General Form Recognition System", Master Thesis of Tsinghua University, 1999

[11] Y. Zheng, C. Liu, X. Ding, S. Pan, "Form Frame Line Detection with Directional Single-Connected Chain", in the Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR), pp. 699-704, 2001

[12] T. Shimamura, B. Zhu, A. Masuda, M. Onuma, T. Sakurada, M. Nakagawa, "A Prototype of an Active Form System", in the Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 921-926, 2003.

[13] D. V. Sharma, G. S. Lehal, "A Fast Skew Detection and Correction Algorithm for Machine Printed Words in Gurmukhi Script', in the proceedings of the International Workshop on Multilingual OCR, Article No.15, ACM, NY, 2009.