Automatic Processing of Structured Handwritten Documents: An Application for Indian Railway Reservation System

Sandip Rakshit
Techno India College of Technology,
Kolkata, India

Soumya Sona Das
Superia Info Solutions Pvt. Limited,
Kolkata, India

Kalyan S Sengupta
ICFAI Business School,
Kolkata, India

Subhadip Basu
CSE Department,
Jadavpur University
Kolkata, India

ABSTRACT

An effective document processing system must be able to recognize structured and semi structured forms that is written by different persons' handwriting. In this work we have developed a method and system that can process structured form document layout and recognize its contents. Our approach has been applied here in the context of Indian railway reservation/cancellation requisition system with encouraging results. In reality, handwritten data usually touch or cross the preprinted form frames and texts, creating complex problems for the recognition routines. In this paper, we address these issues and attempted to solve the problem for Indian Railway Reservation system using our custom built form processing software and Tesseract open source character recognition engine.

General Terms

Pattern Analysis, Machine Learning, Character Recognition.

Keywords

Optical Character Recognition, Handwritten Document Analysis, Form Processing, Tesseract OCR.

1. INTRODUCTION

The automation of handwritten form processing (from pre-formatted documents) is attracting intensive research interests due to its wide application and reduction of the tiresome manual workload. In Indian context most of the people, especially the senior people and the people from rural areas, are not yet familiar with personal computer/Internet system. Therefore online forms will not be able to completely replace physical forms (written and processed manually) in near future. One may argue that computer literacy will eventually lead to paperless scenario in many large scale information processing systems, but systems that largely include partially literate common people of rural India, such goal may not be achieved in near future. Indian Railway Reservation System (IRRS) is an example of one such application. It may be worth mentioning in this context that IRRS is one of the largest in its category that caters around 14 million passengers a day. Although IRRS has an online reservation system [1] but still most of the people fills up their railway reservation form manually. This work mainly focuses on automatic handwritten document processing technology and its applications to IRRS. Our main objective in this paper is to design a standard form layout for IRRS, extract the handwritten characters therein and subsequently recognize them for automatic processing.

Documents can generally be classified into two types, *viz.*, printed documents and handwritten documents. Further we can divide handwritten documents in two categories, *viz.*, unconstrained or unstructured documents and structured or constrained documents. The current work focuses on the second category of documents, i.e., processing of hand filled or handwritten form documents having predefined structures or layouts. Following are the steps in structured form document processing:

- Design the form layout
- Image acquisition(scanning at a specific resolution)
- Image registration (alignment/skew correction)
- Image pre-processing(noise elimination)
- Localization of handwritten content
- Recognition of isolated handwritten characters and digits (for specific scripts).
- Post-processing (systematic error correction, manual intervention etc.)

Handwritten form document processing involves many important technical issues. This is an active research topic and is being researched widely across the globe. In one of the recent research efforts, Navon et.al.[2] describe a method and system for a generic form processing approach, similar to the human visual system, which differentiates between form templates via features such as logos ,key-words, geographical shapes while ignoring minor details and variations. When the system finds the appropriate template, it then decodes the contents of the form. Among other relevant works, Yu et.al. [3], describe a generic system for form dropout, i.e. recovering contents when the filled in characters and symbols are either touching or crossing the form frames/grid. Belaid et,al. [4] describe a robust method to locate the items whose boundaries are lines without using a prior information about the form. In other related works [5-8] the authors primarily concentrated on different methodologies for feature based identification of form document (grid) structures, character segmentation and subsequent recognition of the form-data. Ye et.al. developed a technique [9] for pre-processing and enhancing form document contents. Commercial form document processing software is popular and one such example is available at [10].

Despite contemporary research efforts, novelties of the current work are the design of a structured *Railway Reservation/Cancellation Requisition Form* in Indian context and recognize its contents using a custom-build classifier, derived from the open-source Tesseract project [11-12]. There are unexplored opportunities in automation of Indian official documents. Those efforts are not only confined in mere application of any general purpose form document processing software, but involve intense domain knowledge and through implementation of a custom-designed system. In the next section we first briefly introduce the structure of the existing IRRS. In subsequent sections we explain different steps involved in pre-processing, segmentation and recognition of handwritten contents of our designed *Railway Reservation/Cancellation Requisition Form*.

2. INDIAN RAILWAY RESERVATION SYSTEM

Indian Railway is the largest rail network in Asia and the world's second largest under one management. It covers 64000 of route kilometer along length and width of country [1]. It runs 12000 trains every day, it also carry 14 million passengers and 16 lakh tones of goods every day. Our main concentration of the current research is to re-design the current railway reservation or cancellation requisition form and recognize the contents of the specified forms in various aspects. Currently there are two different options to book or cancel tickets in Indian railway, i.e., through on line or web based interface and using manual ticket booking system. Despite increasing popularity of the web based (online) ticket reservation or cancellation system, vast majority of the Indian population still uses paper based reservation requisition forms for this purpose. Fig 1(a) shows an example of such form, now in use, both in English and in different Indian languages. Presently Indian Railway uses unstructured forms (as shown in Fig 1(a)) that are entered manually into the system by the on-duty counter clerk, at the Indian Railway Reservation counters. It is time consuming both for the clerk and for the customers waiting in long queues in such counters. To address this problem we have designed a structured reservation/cancellation requisition form for IRRS, as shown in Fig 1(b),that can be processed automatically by our designed system.

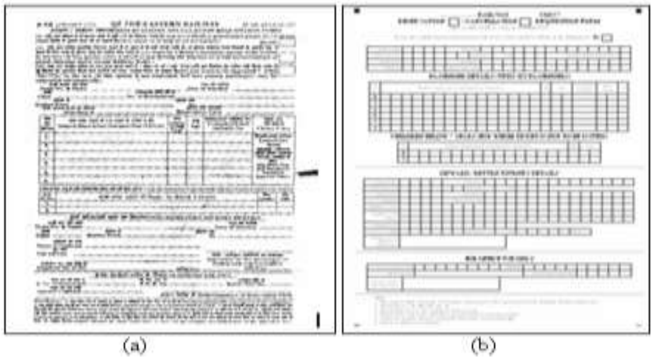


Figure 1. Scanned images of the IRRS forms are shown. (a) Original Railway reservation/cancellation form, currently in use in India. (b) The structured form designed for our proposed IRRS system.

3. FORM PROCESSING METHODOLOGY

In our current work we have first designed the structured form layout and then fill up the forms manually from random users. We then scan the filled-up forms using a flatbed scanner and analyze its contents using the custom-build form-recognition system. This section presents the main steps of our system. The key issues are discussed below:

- Design Issues: We have used structured blocks/boxes/grids for input of every character in the newly designed IRRS form. There are two defined entities in this form, viz., characters and data fields. Multiple characters of one data part are grouped together to constitute one data field (like, six digits of any date field or a signature field). In this way, we have 2510 boxed regions, that constitute 35 data fields. Out of these data fields, 27 box fields are in machine readable forms, where each such field contains isolated box-formatted characters. The remaining 8 non box fields contain 2 signature fields and 6 concession fields. These parts are the form are written in a continuous way and entered into the system manually (or as an image).

- Another important design issue in form document processing is image registration. In this work, we have put square block markers in four corners of the form for registration (skew correction and alignment) of the input images. Input images are scanned by a flatbed document scanner at a resolution of 300 dpi. Once the registration blocks/markers on the input/scanned images are identified by the designed system/program, different morphological filtering is done to extract the embedded data. These steps are discussed below:

Step#1: We first localize the markers/rectangular-blocks, present at the four corners of the scanned image. For that, we crop a square area of pre-fixed dimension from the upper left corner of the original image. This square region is supposed to content the upper left marker. After that we do blob-filtering (connected component labeling) on this cropped square image and the marker is separated as the single largest blob. Now we calculate the starting co-ordinate of that blob as (x_1,y_1), with respect to the original image. In a similar way, we crop a square area from the upper right corner of the original image which is supposed to contain the upper-right marker. Then applying blob filtering in a similar manner we separate the upper-right marker as the single largest blob from the cropped square area and get the starting co-ordinate of the upper-right marker with respect to the original image as (x2,y2). The skew angle (α) of the document image is then calculated as $\alpha=\tan^{-1}((y_2-y_1)/(x_2-x_1))$.

Step#2: As the input image is being skew corrected/aligned now, we can start it for further processing and we use simple morphological operations to extract the data. To detect horizontal lines of the whole forms we used 9x9 horizontal masks, similarly for vertical lines we use 9x9 vertical masks. Then we merge (union) the vertical and horizontal masks' outputs. Now we get inverted image of the form .Then by the using connected component analysis we got each block of the form. In this process, we reject all blocks larger than the standard box size, eliminating the background and the 8 non-box data fields.

Step#3: Here we use one pre-defined coordinate structure to identify the data field position of each block. We compare each block's coordinates with pre-defined coordinate structure and mark its appropriate classification label, *i.e,*. Alpha-Numeric (AN), Alphabetic (A) or Digit (D). Then we combine each groups character images for subsequent processing by the classifiers.

Step#4: Finally, the images of the characters and digits are recognized using the custom-trained Tesseract optical character recognition (OCR) engine (to be discussed to the subsequent section) to generate the result of a particular IRRS form image.

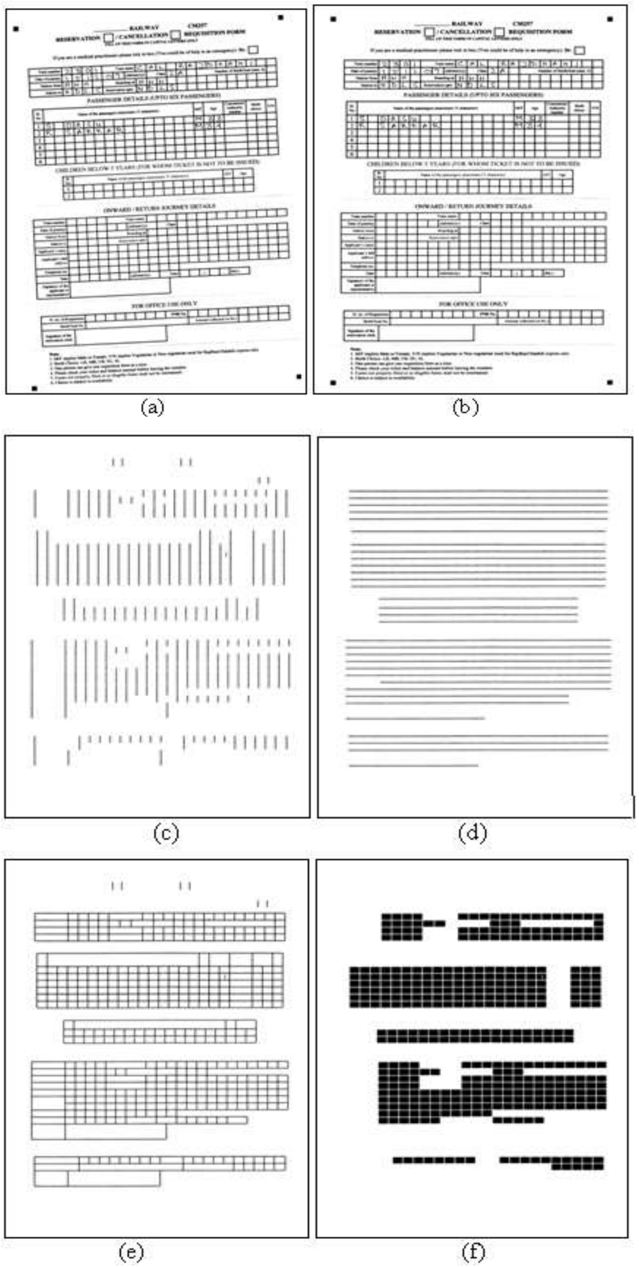


Figure 2. Different pre-processing steps involved in the form-image segmentation process are shown. (a) a sample skewed scanned image of the filled up form (b) the input image is shown again after skew correction by our method (c) output of the horizontal mark on the skew-corrected image (d) output of vertical mark (e) union of the vertical and horizontal mark is shown (f) output of connected component analysis is shown with the identified blocks in black colour.

4. RECOGNITION ENGINE

Here we use the Tesseract open source OCR engine for pattern classification and recognition. Tesseract is an open source (under Apache License 2.0) offline optical character recognition engine, originally developed at Hewlett Packard from 1984 to 1994. Tesseract is now partially funded by Google [16-17] and released under the Apache license, version 2.0.. In the current work, we have used Tesseract 2.03, released in April, 2008.

Like any standard OCR engine, Tesseract is developed on top of the key functional modules like, line and word finder, word recognizer, static character classifier, linguistic analyzer and an adaptive classifier. However, it does not support document layout analysis, output formatting and graphical user interface. To train Tesseract in any language 8 data files are required in tessdata sub directory for the specific language. For example, the 8 files used for English digits (tessdata/eng.*) are generated as follows:

tessdata/eng.freq-dawg

tessdata/eng.word-dawg

tessdata/eng.user-words

tessdata/eng.inttemp

tessdata/eng.normproto

tessdata/eng.pffmtable

tessdata/eng.unicharset

tessdata/eng.DangAmbigs

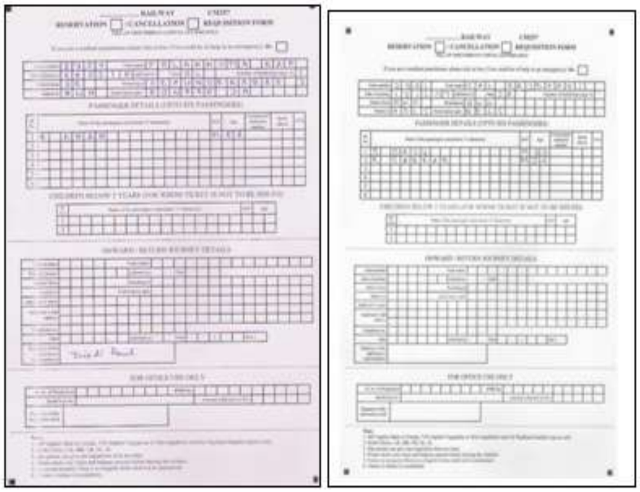
For training a new language set for any user, we have to put in the effort to get one good box file (manual labeling information of input training images) for a handwritten document page, run the rest of the training process, discussed below, to create a new language set. Then use Tesseract again using the newly created language set to label the rest of the box files corresponding to the remaining training images using the process discussed in [13-17], generating the inttemp, normproto, pffmtable and unicharset files.

Tesseract uses 3 dictionary files for each language. Two of the files (freq-dawg and word-dawg) are coded as a Directed Acyclic Word Graph (DAWG), and the other (user-words) is a plain UTF-8 text file. The final data file of Tesseract is DangAmbigs file. This file contains possible ambiguities involved in similar looking character shapes, but cannot be used to translate characters from one set to another. The DangAmbigs file may be empty also.

In this work, we have designed three different classifiers T_{AN}, T_A and T_D using Tesseract for recognizing alphanumeric, alphabetic and digit characters respectively. With the help of these classifiers we have tested our current data set. The experimental results for these classifiers on the IRRS forms are discussed in the following section.

5. EXPERIMENTAL RESULTS

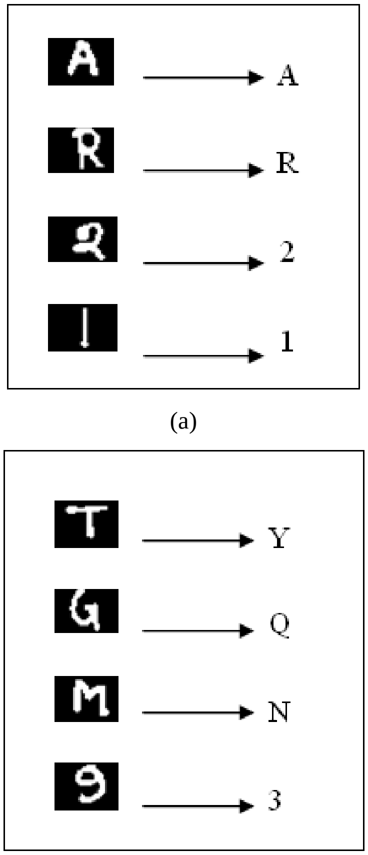
In the current work we have separately trained Tesseract to recognize three different character sets, *viz.*, English capital letters (26 classes), English digits (10 classes) and English alpha-numeric characters (36 classes). Three different classifiers T_A, T_D and T_{AN} are respectively designed for this purpose. We have prepared three different language sets for the three different classifiers by collecting sample isolated training samples from number of different users, chosen randomly. As discussed in our earlier works [13-17], labeled datasets are prepared using bbTesserat tool [16] and the respective training processes are completed following the standard Tesseract steps. The performance of the system is then finally evaluated on filled up IRRS data forms, collected from a newer set of users/writers. In this work we have collected and tested the system on 45 filled up IRRS forms that consist of 2153 characters and 632 digits, totaling 2785 handwritten test samples. Two sample handwritten test data sheets are shown in Fig. 3(a-b).



(a) (b)

Figure-3(a-b). Sample handwritten IRRS forms, used for testing the performance of the designed system, are shown

Out of the total number of character samples, 1937 characters (89.96%) and 573 digits (90.66%) are segmented correctly, leading to an overall segmentation accuracy of 90.12%. The remaining 275 characters (9.88%) are rejected by the present system. This high rejection rate is due to the poor quality of scanning and misalignment in the collected data forms. Among the test samples, 1627 character (75.56% of total number of characters) and 538 digits (85.12% of total number of digit samples) are recognized by the current system. The overall recognition accuracy of 77.73% is achieved by the current system, when we have no rejection parameter. However, when we exclude rejected samples (and raise flags for manual intervention for them) we have improved recognition percentage. Among the correctly segmented data samples, 84% characters and 93.89% digits are recognized correctly by our current Tesseract based recognition engine. Fig. 4(a) shows sample images of characters/digits where the present system performs satisfactorily. However, in Fig. 4(b) we show some misclassified images, where the Tesseract based recognizer fails in identifying the true character/digit classes.



(a)

(b)

Figure 4. (a) Sample images where the current recognizer successfully recognizes the data samples. (b) Misclassified data samples with the corresponding erroneous data labels are shown.

6. CONCLUSION

The current work is an attempt to simplify the existing manual data-processing task involved in the Indian Railway Reservation/Cancellation process, by proposing a smart document management system. Contribution of the work is in design of the new IRRS form layout and development of a Tesseract based form processing system. The segmentation accuracy is low in the current system. In future research, we will attempt to develop an improved form segmentation algorithm and expand the training and test dataset sizes. We will also conduct a software acceptance survey on the developed software with the potential users of the proposed system. Another future direction of the present work is to incorporate just-in time information retrieval system [18], i.e, including features for indexing and searching handwritten contents in the document archives.

7. ACKNOWLEDGMENTS

The first author is thankful to the R&D Cell of Techno India College of Technology, India, for providing necessary supports during the research work. The last author is thankful to the Center for Microprocessor Application for Training Education and Research (CMATER), SRUVM project of Computer Science and Engineering Department, Jadavpur University, for providing infrastructure facilities during of the progress of the work. The last author also acknowledges the Department of Science and Technology, Govt. of India, for providing him the BOYSCAST fellowship to visit The University Iowa, USA, where part of this manuscript was written.

8. REFERENCES

[1] http://www.indianrail.gov.in/..

[2] Yaakov Navon, Ella Barkan, Boaz Ophir, "A Generic Form Processing Approach for Large Variant Templates," icdar, pp.311-315, 2009 10th International Conference on Document Analysis and Recognition, 2009.

[3] B. Yu and A. K. Jain, "A Generic System for Form Dropout", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 18, No. 11, Nov. 1996, pp. 1127-1134.

[4] Y. Belaid, et al., “Item Searching in Forms: Application to French Tax Form”, Int. Conf. on Document Analysis and Recognition, Aug. 1995, pp. 744-747.

[5] C.D. Yan, Y.Y. Tang, and C.Y. Suen, “Form Understanding System Based on Form Description Language”, Int. Conf. on Document Analysis andRecognition, Oct. 1991, pp. 283-293.

[6] K. Fan and M. Chang, “Form document identification using line structure based features”, Proc. Int. Conf. on Pattern Recognition, Vol. 2, Aug. 1998, pp. 1098 – 1100

[7] H. Fujisawa, Y. Nakano, and K. Kurino, “Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis”, Proc. of the IEEE, Vol. 80, No. 7, 1992, pp. 1079-1092.

[8] Hiroshi Sako et al., “Form Reading based on Form-type Identification and Form-data Recognition”, Int. Conf. on Doc. Ana. and Recognition, Aug. 2003, Vol. 2, pp. 926-930.

[9] X. Ye, M. Cheriet and C.Y. Suen, “A generic method of cleaning and enhancing handwritten data from business forms,” International Journal on Document Analysis and Recognition, vol. 4, pp. 84-96, 2001.

[10] http://www.smartform.com/.

[11] http://code.google.com/p/tesseract-ocr

[12] R. Smith. “An overview of the Tesseract OCR engine”. In ICDAR’2007, International Conference on Document Analysis and Recognition, Curitiba, Brazil, Sept. 2007

[13] S.Rakshit, A. Kundu, M. Maity,S. Mandal, S. Sarkar, S. Basu, “Recognition of handwritten Roman Numerals using Tesseract open source OCR engine” Second International Conference on Advances in Computer Vision and Information Technology (ACVIT 2009) pp572-577.

[14] S. Rakshit,S. Basu, Hisashi Ikeda “ Recognition of Handwritten Textual Annotations Using Tesseract Open Source OCR Engine FOR information Just in Time (iJIT) In Proc.of International Conference on Information Technology and business Intelligence(ITBI-09).

[15] S.Rakshit, S. Basu, “Development of a Multiuser Handwritten Recognition System Using Tesseract Open source OCR” in proc. of C3IT-2009 An International conference, pp.240-247 .Proceedings published by Macmillan advanced Research Series, ISBN NO: 023-063-759-0

[16] S. Rakshit, S. Basu, “Recognition of Handwritten Roman Script Using Tesseract Open source OCR Engine,” in proc. of National Conference on NAQC-2008, pp. 141-145, Kolkata.

[17] S. Rakshit, D. Ghosal, T. Das, S. Dutta, S. Basu, “Development Of A Multi-User Recognition Engine For Handwritten Bangla Basic Characters And Digits” In Proc.(CD)of International Conference on Information Technology and business Intelligence(ITBI-09).

[18] S. Basu, K. Konishi, N. Furukawa, H, Ikeda, “A novel scheme for retrieval of handwritten textual annotations for information Just In Time (iJIT)”, proceedings (CD) of IEEE Region 10 Conference (TENCON) -2008