Multimed Tools Appl (2018) 77:8441–8473
DOI 10.1007/s11042-017-4745-3

Benchmark databases of handwritten *Bangla-Roman* and *Devanagari-Roman* mixed-script document images

Pawan Kumar Singh[1] · **Ram Sarkar**[1] · **Nibaran Das**[1] · **Subhadip Basu**[1] · **Mahantapas Kundu**[1] · **Mita Nasipuri**[1]

Received: 20 September 2016 / Revised: 20 February 2017 / Accepted: 21 April 2017 /
Published online: 18 May 2017
© Springer Science+Business Media New York 2017

Abstract Handwritten document image dataset is one of the basic necessities to conduct research on developing Optical Character Recognition (OCR) systems. In a multilingual country like India, handwritten documents often contain more than one script, leading to complex pattern analysis problems. In this paper, we highlight two such situations where *Devanagari* and *Bangla* scripts, two most widely used scripts in Indian sub-continent, are individually used along with *Roman* script in documents. We address three key challenges here: 1) collection, compilation and organization of benchmark databases of images of 150 *Bangla-Roman* and 150 *Devanagari-Roman* mixed-script handwritten document pages respectively, 2) script-level annotation of 18931 *Bangla* words, 15528 *Devanagari* words and 10331 *Roman* words in those 300 document pages, and 3) development of a bi-script and tri-script word-level script identification module using Modified log-Gabor filter as feature extractor. The technique is statistically validated using multiple classifiers and it is found that Multi-Layer Perceptron (MLP) classifier performs the best. Average word-level script identification accuracies of 92.32%, 95.30% and 93.78% are achieved using 3-fold cross validation for *Bangla-Roman*, *Devanagari-Roman* and *Bangla-Devanagari-Roman* databases respectively. Both the mixed-script document databases along with the script-level annotations and 44790 extracted word images of the three aforementioned scripts are available freely at https://code.google.com/p/cmaterdb/.

Keywords Script identification · Handwritten text · Mixed-script documents · Optical character recognition · Modified log-Gabor filter · Transform · Statistical significance tests

✉ Ram Sarkar
raamsarkar@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

1 Introduction

One of the important tasks of document image analysis is automatic reading of text information from the document image. This is performed using the tool Optical Character Recognition, usually abbreviated as OCR, which is referred to the electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine editable text. An OCR system enables us to take a book or a magazine article, feed it directly into an electronic computer file, and then edit the file using available word processing software. Mixed-script documents contain text words written in more than one language. As India is a multilingual country, therefore, it is obvious that a document is composed of text contents written in multiple (often two) languages. As a consequence, OCRing such a document possess a real difficulty because the language/script types of the text need to be pre-determined, before employing a particular OCR engine. This is because that every OCR system makes an imperative inherent postulation that a particular script, in which the document is written, is known in advance. Therefore, such processing of documents which heavily depends on OCR would undoubtedly necessitate human intervention to select the suitable OCR package. This criterion is certainly inefficient, undesirable and unrealistic in an automatic multilingual situation. Design of a single recognizer system which can identify a large number of scripts/languages is also perhaps close to impossible. Therefore, before allocating the input document to its corresponding OCR system, it becomes obligatory to initially recognize the language/script in which the document is written.

India is a multilingual country where 23 constitutionally recognized languages are there which are written using 12 major scripts. Besides these, hundreds of other languages are used in India, each one with a number of dialects. The officially recognized languages are: *Hindi*, *Bangla*, *Punjabi*, *Gujarati*, *Oriya*, *Sindhi*, *Assamese*, *Nepali*, *Marathi*, *Urdu*, *Sanskrit*, *Tamil*, *Telugu*, *Kannada*, *Malayalam*, *Kashmiri*, *Manipuri*, *Konkani*, *Maithali*, *Santhali*, *Bodo*, *Dogari* and *English*. Scripts used to write these languages are: *Devanagari*, *Bangla*, *Oriya*, *Gujarati*, *Gurumukhi*, *Tamil*, *Telugu*, *Kannada*, *Malayalam*, *Manipuri*, *Urdu* and *Roman*. The first 11 scripts are originated from the early *Brahmi* script (300 BC) and are also referred to as *Indic* scripts [55]. *Indic* scripts are a logical composition of individual script symbols and follow a common logical structure. This can be referred to as the "script composition grammar" which has no counterpart in any other set of scripts in the world. *Indic* scripts are written syllabically and are usually visually composed in three tiers where constituent symbols in each tier play specific roles in the interpretation of that syllable. Besides, being the official languages, *Hindi* and *Bangla* are the most popular languages (in terms of the total number of speakers) in Indian sub-continent. *Devanagari* script is used to write *Hindi, Nepali, Marathi* and *Sindhi* languages and *Bangla* script is used to write *Assamese*, *Manipuri* and *Bangla* languages. *English* is the binding language due to the colonial past in our country as well as the diversity of languages/scripts in India and other parts of the world. However, *English* written using *Roman* script is frequently used in conjunction with different *Indic* scripts while writing a text document. Their usage is frequently seen in advertisements, movies, and text messaging nowadays. A multilingual document such as railway reservation forms, question papers, language translation books and money-order forms, etc. may contain text in more than one script/language. Script identification has long been the forerunner of many OCR processes as a precursor during the preprocessing stages. Identification of scripts is also essential to extract information presented in digitized documents *namely*, articles, newspapers, magazines and e-books [55]. Document analysis systems that facilitate processing of these stored images are crucial for both efficient archival and providing access to various researchers.

Script identification is a vital footstep that arises in document image analysis particularly in a multi-script and multilingual situation. The solution of this dilemma is the development of an automatic script identification system. Script identification facilitates many important applications such as sorting and selecting appropriate script specific text understanding system and searching online archives of document images comprising of a particular script, etc. [15].

Processing of handwritten and machine printed documents require different approaches. Handwriting consists of elongated strokes, whereas the machine counterpart consists of regularly spaced blobs. Handwritten documents present three challenges for script identification. Firstly, the resemblance among different scripts is more commonly found in handwritten documents rather than in printed ones. Secondly, a single character (or word) written by different individuals possesses the catalog of different possible character (or word) shapes that can be frequently seen in case of handwritten documents. This is due to individual differences, and even differences seen in the writing styles of analogous people at different instances. Thirdly, typical problems such as ruling lines, word fragmentation due to low contrast, noise, skew, etc. are commonly found in handwritten documents. Researchers face enormous difficulties while segmenting and recognizing handwritten text due to the wide variations in handwriting styles which poses huge challenges in script identification scheme.

Script identification is generally achieved at three levels: (a) Page-level, (b) Text-line level and (c) Word-level. A detailed survey on script identification described by Singh et al. [55] shows that researches on identification of different scripts from document pages [15, 25, 26, 36, 38, 50, 56] or text-lines [29, 31, 37, 39, 42, 57] are limited in the literature. In comparison to this, script recognition at the word-level in a multi-script document is generally much more challenging but useful. It is challenging because the information available from only a few characters in a word may not be adequate for the purpose. Furthermore, the variation of different scripts in the form of text words (generally bi-script) is commonly seen rather than in text-lines or document pages. Hence, the identification of scripts at word-level is much more preferable than its other two counterparts. Some researchers have even attempted to do script identification at the character level. However, script recognition at the character level is generally not required in practice. This is because the script usually changes only from one word to the next and not from one character to another within a word. Some of the word-level script identification methodologies are discussed in [9–12, 16, 17, 24, 40, 41, 44–47, 49, 51, 53, 54].

It can be observed from the literature study that most of the existing works [9–12, 16, 17, 24, 40, 41, 45–47] are done on printed script words whereas only few works [44, 49, 51, 53, 54] are available for identification of handwritten *Indic* scripts. K. Roy et al. [49] have described a scheme for word-wise identification of handwritten *Roman* and *Oriya* scripts for Indian postal automation. In the proposed scheme, at first, the document skew is corrected. Using a piece-wise horizontal projection, the document is segmented into text lines and by vertical histogram, the text lines are segmented into words. Finally, some features based on fractal dimension, presence of small component, water reservoir, topology of a word, etc. are used for the *Oriya* and *English* script word identification by using a MLP classifier. R. Sarkar et al. [51] have proposed 8 holistic features for word-level script identification from *Bangla* and *Devanagari* handwritten texts mixed with *Roman* script by using MLP classifier. P. K. Singh et al. [53] have reported an intelligent feature based technique for word-level script identification of *Devanagari* script mixed with *Roman* script. A set of 39 distinctive features comprising of 8 topological and 31 convex hull based features had been designed. An MLP classifier with these 39 features is used to identify the said scripts. In [54], performances of

multiple classifiers are evaluated with the designed feature set (described in [53]) for selection of a suitable classifier on randomly selected multiple datasets of *Devanagari* and *Roman* script words. A set of statistical significance tests followed by its corresponding post-hoc tests has also been performed as an essential part for validating the performance of the multiple classifiers using multiple datasets. A word-level handwritten *Indic* script identification technique for 11 different major Indian scripts (including *Roman*) in bi-script and tri-script scenarios has been proposed by R. Pardeshi et al. [44]. The features are extracted based on the combination of Radon transform, Discrete wavelet transform, Statistical filters and Discrete cosine transform. The classification is done using linear discriminant analysis, Support Vector Machine (SVM) and *k*-nearest neighbor classifiers.

The main contribution of our work is the development of benchmark databases comprising of 150 *Bangla-Roman* and 150 *Devanagari-Roman* mixed-script handwritten document pages. We have also applied a robust page-to-word segmentation algorithm for segmenting the word images from the handwritten document pages. Finally, a method based on Modified log-Gabor filter approach and MLP classifier is also presented for handwritten word-level script identification. The present scheme has also been tested on the developed handwritten databases and the corresponding recognition accuracies in bi-script and tri-script scenarios are also reported here. Fig. 1 shows the block diagram of the present approach.

The organization of the paper is done in the following way: First, the need for standardization of database is described in Section 2 and some characteristics of *Devanagari* and *Bangla* scripts are described in Section 3. Section 4 deals with detailed dataset description including data collection and pre-processing. Compositions of the databases are described in Section 5. Information related to ground truth annotations and GTGen software is described in Section 6. Section 7 discusses the benchmark script separation result on the developed databases, and experimental results and discussion are provided in Section 8. Finally, conclusion and scope of future work are given in Section 9.

2 Need for standardization of experimental data

A document containing text information in more than one script is called a mixed-script document. Many of the Indian documents contain two scripts *namely*, the state's official language (local script) and *English*.

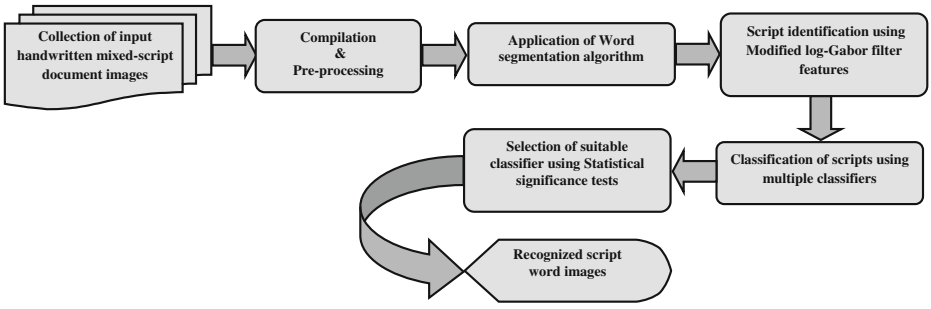


Fig. 1 Schematic diagram showing the key modules of the present approach

This is because *English* is frequently used in daily activities, along with almost all official purposes. *English* is one of the key mediums of education in our country. Even in text-books written in regional language, keywords are mentioned in *English* too. Above all, as our country-people use various languages, hence, *English* acts as the binding language for us. These are the main reasons that mixed-script documents are so pertinent in Indian sub-continent. Fig. 2 shows some samples of mixed-script printed documents used in India. All the Indian languages do not have the unique scripts. Some of them share the same script. Among these, *Devanagari* is the most widely used script; it is the script of *Hindi* language which is the fourth most popular language in the world. Being the official language, *Hindi* is a medium through which messages are communicated in multilingual and heterogeneous Indian society. As compared to other languages (international), progress in *Devanagari* and *Bangla*

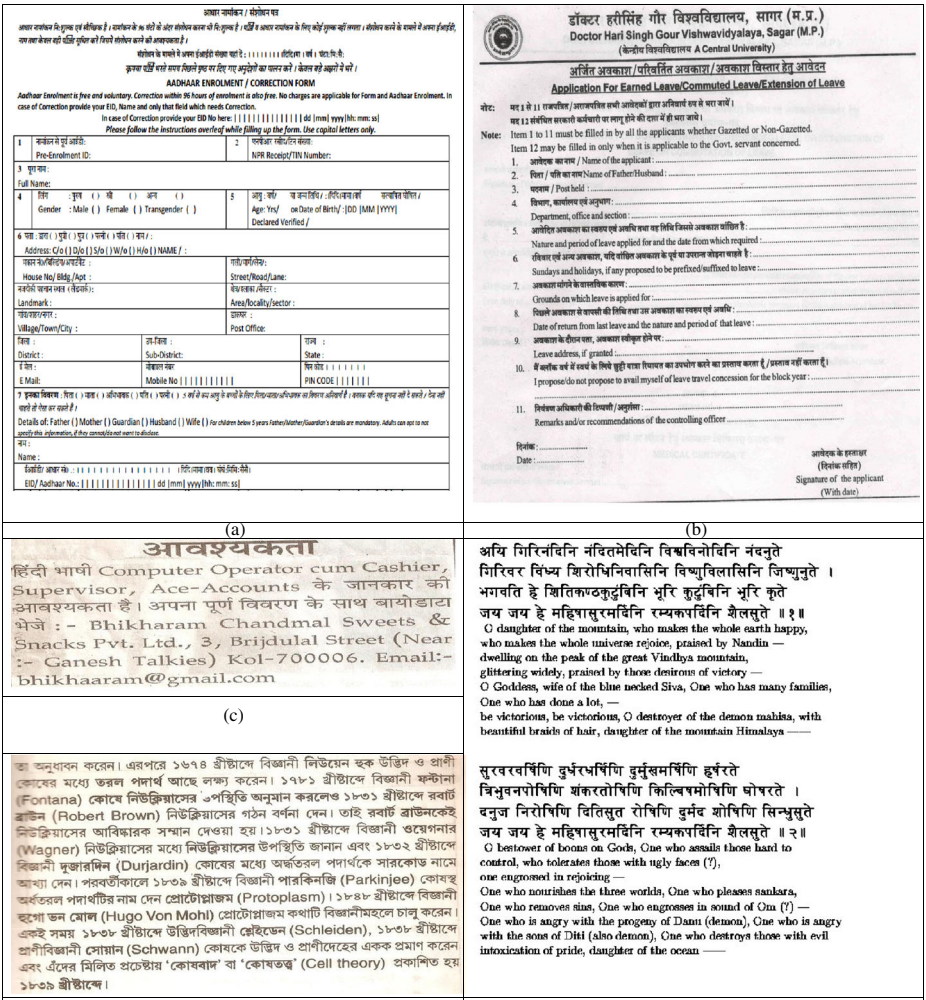


Fig. 2 Examples showing mixed-script documents used in Indian sub-continent: **a** government job application form, **b** college leave application form, **c** newspaper advertisement, **d** *Bangla* school text-book, and **e** treasure of *Stotras* in both *Sanskrit and English*

character recognition systems have not been achieved satisfactory advancement till date. Also, early researchers paid very little attention to test data collection. Invariably, many of them tested their algorithms on artificially crafted datasets. In our assessment, lack of standard dataset is one of the important reasons for the slow progress in developing the *Devanagari* and *Bangla* OCR systems. In order to build a realistic system, researchers need handwriting samples collected from different sections of society. Such samples would help in understanding the complex structure of any script, discovering features, and training and testing the system in real environment. In recent years, efforts to create dataset for Indian languages are being reported in the literature. The study on *Indic* scripts has got prime attention in last few decades. Many authors have taken the challenges and are working on several Indian languages. A brief summary of dataset available for *Indic* scripts, surveyed here, is presented in Table 1 for quick referencing. Survey shows that efforts to collect *Devanagari* or *Bangla* dataset started after 2000 (Pal et al. [43], Bhattacharya et al. [3], and Jayadevan et al. [28]).

However, to the best of our knowledge, there is no public domain database freely available till date for unconstrained handwritten document pages of mixed-script document written in *Bangla* or *Devanagari* mixed with *Roman* words. There are two main important issues related to handling the document pages in a mixed-script environment. The first approach requires a robust page-to-word segmentation technique to extract the words written in different scripts which are fed to the script identification module. Whereas the second approach is to initially perform text-line segmentation followed by word segmentation from the document images. However, the computational complexity using former approach is much less as compared to the latter one. In multi-script environment, a single document is written using a particular script and one can apply the script identification module at page-level to avoid complexities as designing an appropriate script independent text- line/word segmentation technique for handwritten documents is a very challenging task. It may be worth mentioning at this point that for *Indic*, *Arabic,* and *Chinese* scripts, special techniques are required to implement handwritten OCR algorithms. Previous researches on *Indic* script

Table 1 Summarization of datasets for *Indic* scripts available till date

Reference	Year	Language/Script	Data Type	Size of Datasets
Pal et al. [43]	2007	*Devanagari*, *Bangla*, *Telugu*, *Oriya*, *Kannada*, and *Tamil*	Isolated Numerals	22,546,14,650,2220,5638, 4820 and 2690 Numeral images respectively
Chaudhary [13]	2007	Online and offline handwritten *Bangla*	Strings of Numerals and Isolated Numerals	Online-8348 Numeral images Offline −23,392 isolated Numeral images
Bhattacharya et al. [3]	2009	*Devanagari*	Isolated Numerals	22,556 isolated Numeral Images
Nethravathi et al. [35]	2010	*Tamil* and *Kannada*	Handwritten Documents from 600 subjects	100,000 word images
Alaei et al. [1]	2011	*Kannada*	51 native speakers	4298 text line images and 26115word images
Jayadevan et al. [28]	2011	*Hindi* and *Marathi*	Legal amount words	26720word images
Sarkar et al. [52]	2012	*Bangla*, *Bangla-English*	Handwritten *Bangla* Text and *Bangla* text mixed with *Roman script* words	100 pages-purely Handwritten *Bangla* text 50 pages-*Bangla* and *English* mixed text

recognition systems were reported on the basis of databases artificially created for training and testing their developed systems. But, future research in this domain requires standard benchmark databases fulfilling certain criteria depending on the application domain. This will in turn help the researchers to test their developed techniques on a common platform and compare their recognition accuracies. To address these issues, we have been motivated to prepare two moderately large datasets which consist of handwritten document databases containing both *Bangla-Roman* and *Devanagari-Roman* words. The research on mixed-script document pages would gain popularity because due to the presence of two contrasting types of scripts inscribed in it.

3 Characteritics of scripts

3.1 *Devanagari* script

Devanagari script is a derivative of ancient *Brahmi* script which is mother of almost all *Indic* scripts. Nearly more than 300 million people from all over the world use *Devanagari* script [32]. Word formation in *Indic* scripts follows a definite script composition rule for which there is no counterpart in *Roman*. *Devanagari* script is used to write *Hindi, Nepali, Marathi, Sindhi,* etc. So, this script plays a very important role in the literature and manuscripts in India.

Devanagari has 13 vowels and 33 consonants. Besides this, other constituent symbols in *Devanagari* are set of vowel modifiers (placed to the left, right, above, or at the bottom of a character or conjunct), pure-consonant (also called half-letter) which when combined with other consonants yield conjuncts. A horizontal line called Shirorekha (a headline) runs through entire span of a word.

3.2 *Bangla* script

Bangla is the seventh most popular script in the world [32]. *Bangla* script is used to write *Bangla, Assamese* and *Manipuri* languages. There are 11 vowels and 39 consonants in modern *Bangla* alphabet. They are called basic characters. Sometimes two or more characters get combined and generate a new shape which is known as compound character. Many characters of *Bangla* alphabet have a horizontal line at the upper zone. This line is called Matra or headline.

4 Database description

4.1 Database nomenclature

Our developed database have been named as *CMATERdb*1 and *CMATERdb*2,where *CMATER* stands for ‘Center for Microprocessor Applications for Training Education and Research’, a research laboratory at Computer Science and Engineering department of Jadavpur University, India, where the current databases are prepared. Here, *db* symbolizes *database*, and the numeric values 1 and 2 represent handwritten database at page and word-level respectively. In the current work, we have developed two variations of *CMATERdb*1and three variations of *CMATERdb*2 which are enlisted in Table 2. These databases are available *freely* at https://code.google.com/p/cmaterdb/and the link is also given in our *CMATER* website (www.cmaterju.org).

Table 2 Tabular representation showing all the variations of the developed databases *namely*, *CMATERdb*1 and *CMATERdb*2

CMATERdb#	Release Version	Script(s)	Dataset
1.2	*CMATERdb*1.2.2 (Second Version)	*Bangla* and *Roman*	150 pages mixed script document pages
1.5	*CMATERdb*1.5.1 (First Version)	*Devanagari* and *Roman*	150 pages mixed Script document pages
2.1	*CMATERdb*2.1.3 (Third Version)	*Bangla*	18931words
2.2	*CMATERdb*2.2.3 (Third Version)	*Devanagari*	15528words
2.3	*CMATERdb*2.3.1 (First Version)	*Roman*	10,331 words

4.2 Data collection

Materials of the handwritten document pages for the proposed databases have been written by different persons. Document pages were collected from various individuals who were requested to write textual contents selected from newspaper articles and text-books containing both *Devanagari* (or *Bangla*) and *English* vocabularies. The writers were asked to use a black or blue ink pen and write inside the margins on all the four sides of A-4 size pages. Most of them took the content from either school text-books, or articles of popular daily *Hindi* newspaper "Sanmarg", and *Bangla* newspaper "Anandabazar Patrika". No other restrictions were imposed regarding the kind of pen they used or the style of writing chosen. Special attention was paid to ensure data collection from the writers belonging to different origins, age-groups and educational levels. Moreover, we collected the pages from different places (home, office, school etc.) in order to include different styles of writing. In total, 150 men and 150 women participated in this data collection drive. The main highlighting aspect of our developed database is the heterogeneity with respect to three important factors: *namely*, state of origin, educational background and age among the writers participated in the data collection process which is shown in Fig. 3a-c.

4.3 Digitization and pre-processing

All the document pages were scanned using a flatbed scanner with 300 dpi gray scale image resolution. Each page, meant for the databases *CMATERdb*1.5.1and *CMATERdb*1.2.2, is stored in 24-bitmap file format with the naming convention HE###.bmp and BE###.bmp

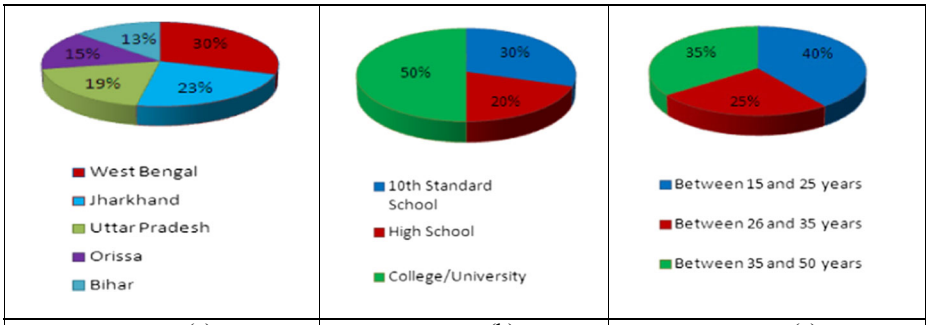
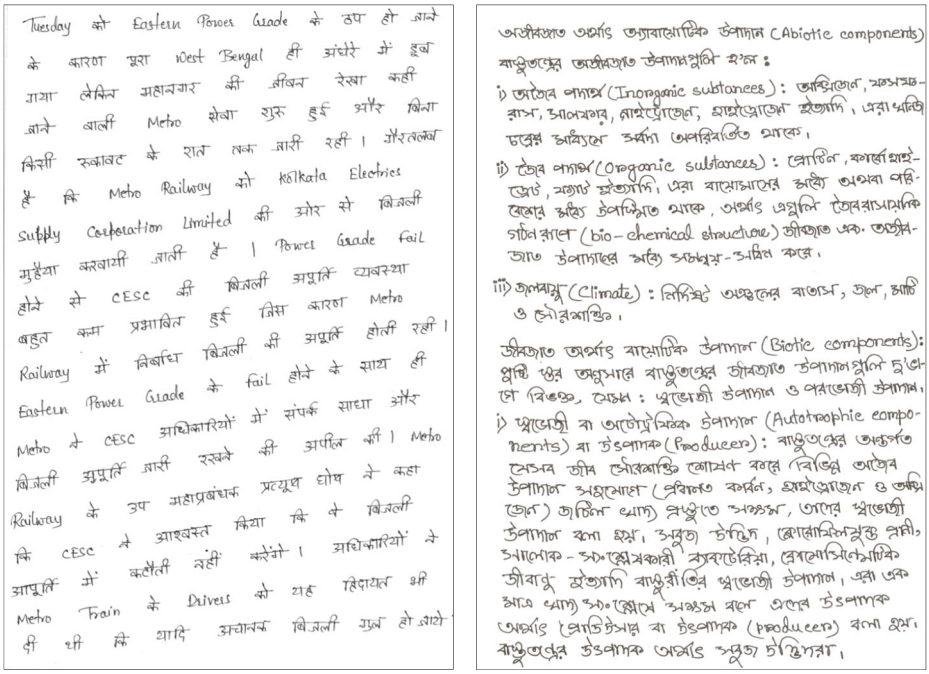


Fig. 3 Graphical representation highlighting the writer's information such as: **a** state of origin, **b** educational level, and **c** age group

respectively. ### is a unique integer given to the file name to maintain sequence, and HE or BE refers to the document type, i.e., *Devanagari–Roman* or *Bangla-Roman*, respectively. One sample image from each of these databases is shown in Fig. 4a-b. On the other hand, the remaining three databases *namely*, *CMATERdb*2.1.3, *CMATERdb*2.2.3 and *CMATERdb*2.3.1, are also stored as 24-bitmap file format with the same naming convention data#####.bmp. After scanning, the documents are binarized by simple adaptive *thresholding* technique, where the threshold was chosen as the *average* of *maximum* and *minimum* gray level values in each document image. All the binarized images were archived in DAT format, where foreground and background pixels are represented as '0' and '1', respectively. Then, the documents are preprocessed in order to remove all the remaining noisy artifacts like long lines present along the margins on the collection sheet. All the binarized images are finally labeled with the ground truth annotations for the purpose of script recognition.

5 Develpoed database

*CMATERdb*1.5.1, the *Devanagari* mixed with *Roman* scripts handwritten document database contains 150 pages in its first version whereas *CMATERdb*1.2.2 contains 150 handwritten document pages in its second version comprising of *Bangla* mixed with *Roman* script. Each of the document pages of these databases are described with the help of some auxiliary information like height, width and aspect ratio, total number of text lines and *Devanagari/Bangla* script words, and statistical estimations of the average horizontal and vertical stroke widths.



(a) (b)

Fig. 4 **a-b** Sample document images from: **a** *CMATERdb*1.5.1, and **b** *CMATERdb*1.2.2

Detailed descriptions regarding the averages and standard deviations of all the attributes of document pages of the two databases are uploaded as supplementary files in the database website [https://code.google.com/p/cmaterdb/]. Document attributes related to page dimensions are actually based on the scanned region of the images. In most cases, we have attempted to preserve the original/physical page dimensions, but in some cases, they may get compromised because of misalignment due to scanning or cropping of torn out page boundaries. Counting of total number of text lines as well as number of words written in different scripts in the document images are done manually at the *CMATER* research laboratory. These attributes are necessary for evaluating effective page segmentation and script identification algorithms. Stroke width in any binarized document image is estimated as the run of black pixels in any given direction (horizontal/vertical) which is shown in Fig. 5. Unlike other features, these two features are computed programmatically and are particularly useful in estimating an important characteristic of the writers, i.e., the connectedness in writing style. These writers' characteristics play key roles in designing different features for character/word recognition.

Popularly used run length-based features are specifically sensitive to stroke width of any unconstrained handwriting. Run-length based horizontalness and verticalness attributes in document/word images are widely used for script separation from document images. Average horizontal stroke width has been calculated as the mean of all the continuous run of black pixels along the rows. Likewise, average vertical stroke width is also computed over the mean of column-wise runs of black pixels. In order to estimate the variation of density of text words present in the handwritten document pages, counts of the number of *Roman* words written in each document page taken from the database *CMATERdb*1.5.1 and *CMATERdb*1.2.2 are also shown in Fig. 6a-b respectively.

6 Ground truth of the databases

Generation of appropriate ground truth data has always been a challenging and tiresome task for the kind of problem under consideration. Availability of ground truth information, however, makes any database more useful, enabling proper evaluation of one's technique by comparing their output with the ground truth. In this work, we have prepared ground truth images for all the pages of our databases, viz., *CMATERdb*1.5.1 and *CMATERdb*1.2.2 for script identification application. For each of the two handwritten databases, we have generated the ground truth information, which has been archived as *CMATERgt*1.5.1 and *CMATERgt*1.2.2, respectively. These ground truth images of the databases are prepared in a *semi-automatic* way. We have applied a two-password identification approach, as described in [59], for identifying individual word images from any document image containing *Bangla/Devanagari* script words mixed with *Roman* script words. In the first pass, key points are

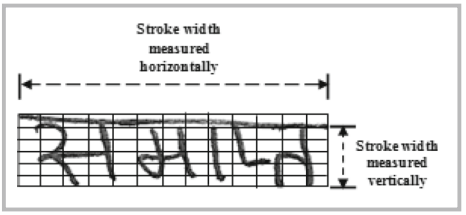


Fig. 5 Illustration of horizontal and vertical stroke widths on a sample *Devanagari* script word

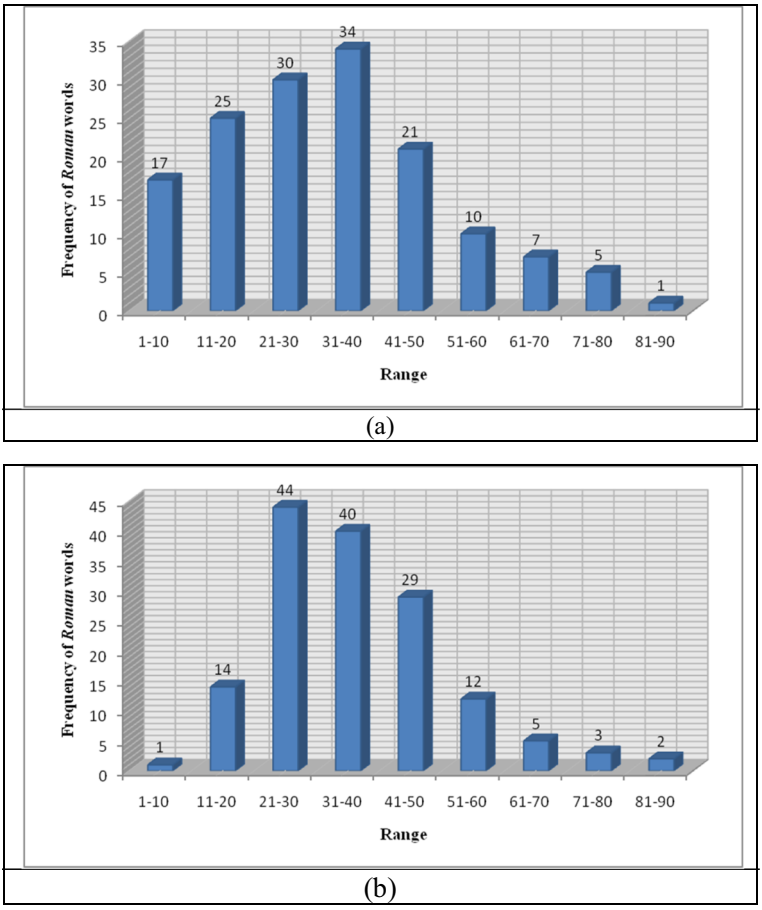


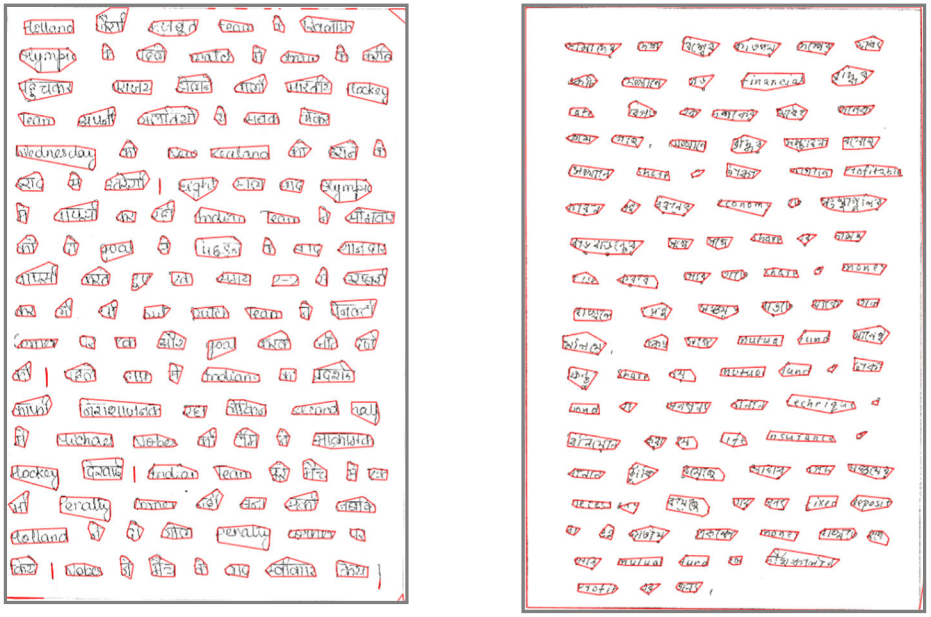
Fig. 6 Graphical analysis showing the histogram of the frequency of occurrence of *Roman* script words written in each document page taken from the database: **a** *CMATERdb*1.2.2, and **b** *CMATERdb*1.5.1 respectively

initially estimated from the handwritten document images using Harris corner point detection algorithm. Harris corner detector [23] is based on the local auto-correlation function of an image which measures local changes of an image with patches shifted by a small amount in different directions. It is based on the Moravec Operator which is used to compare the error between shifted patches with the original image using sum of squared differences [33]. The feature points generated from Harris corner point detection are passed on to Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [19]. Given a set of points in some space, it groups together points that are closely packed together, marking as outlier points that lie unaccompanied in low-density regions. DBSCAN requires two parameters: 1) distance up to which points are to be checked whether it belongs to a particular cluster or not i.e., ε and the minimum number of points required toform a dense region (*minPts*). These points' neighborhood up to distance ε is retrieved, and if it contains significant number of points, a cluster is initiated. Values of ε and *minPts* for the present work are set on trial-and-error basis while executing the DBSCAN algorithm. Finally, the boundary of the text words present in a document image are estimated based on the convex hull [22] drawn for each of the clustered key points. In the second pass, a simple post-processing technique has been applied

for handling the two major error cases: over-segmentation and under-segmentation of the words. If a single word component is erroneously broken down into two/more parts, then it is considered as an over-segmentation error. Whereas if two/more words are recognized as a single word, then it is considered as an under-segmentation error. Possible causes of these errors are either wrongly detection of Harris corner points or improper clustering of the corner points around the word images. To combine over-segmented components, spatial distance between two neighbouring convex hulls is measured to verify their closeness and those two convex hulls are merged if they are close by. For under-segmented components, vertical histogram of the word image is considered and the minima valley is calculated which considers the gap in between two or more consecutive words. This gap is taken into consideration to separate the word images. Examples of successful word extraction algorithm on document pages taken from the two databases *CMATERdb*1.5.1 and*CMATERdb*1.2.2 are shown in Figs. 7a and b respectively.

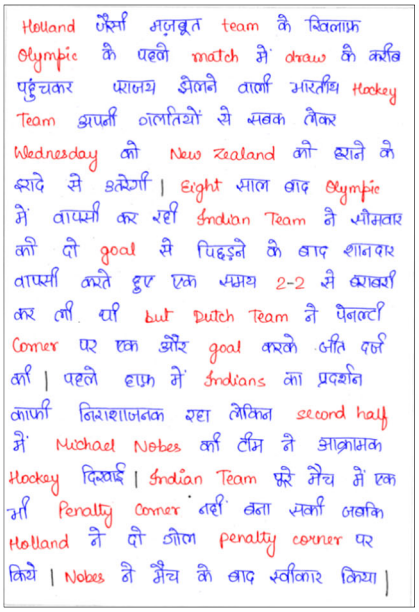
Then, a software tool called GTGen version 1.1, developed in *CMATER* research laboratory, is used for correcting the possible errors that might have been generated in script separation algorithm. In addition, we have also used GTGen to recolor those words or part of the words which have been labeled erroneously by our script separation technique. It may be noted that all the ground truth images are stored in bitmap file format, where the background is labeled in white and individual scripts are marked in different colors. All the files in *CMATERgt*1.5.1 and *CMATERgt*1.2.2 are named as GTHE###.bmp and GTBE###.bmp respectively. Figs. 8a-b shows sample ground truth images from the two databases respectively, prepared for the script separation application.

GTGen version 1.1 is a software tool, developed in Visual Basic dot net technology at the *CMATER* research laboratory that can label text in any chosen color. GTGen reads images

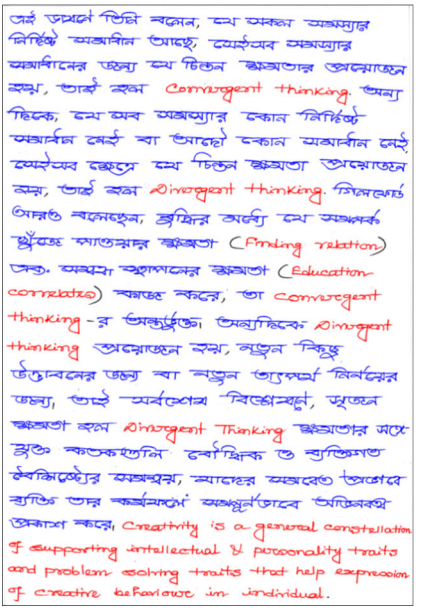


(a) (b)

Fig. 7 **a-b** Sample document pages after the application of word identification algorithm on: **a** *CMATERdb*1.5.1, and **b** *CMATERdb*1.2.2



(a)



(b)

Fig. 8 a-b Sample ground truth images taken from **a** *CMATERgt*1.5.1 and, **b** *CMATERgt*1.2.2. (where *Devanagari* and *Bangla* scripts are shown in *blue color*, *Roman* script shown is in *red color* and non-text components are shown in *black color*)

having white background. One can select any color from a color panel and use that to recolor the text by selecting the intended region with a mouse. Using this technique, we can easily correct errors in our script identification algorithm to generate ground truth data. We can even use this tool to label words written in different scripts for mixed-script document pages or even generate ground truths for text-line and word segmentation algorithms. This software setup is also made available *freely* at https://code.google.com/p/cmaterdb/.

7 Benchmark script identification result on the developed databases

For any successful pattern classification system, it is very challenging but essential to design features which are strong enough to categorize an input pattern to the actual class to which it belongs to. Proposed scheme is inspired from the observation that the humans are capable of distinguishing unknown scripts just based on visual inspection. We assume the script identification as a process of the texture classification. In general, a texture is a complex visual pattern composed of sub-patterns (http://www.csse.uwa.edu.au/~pk/research/matlabfns/PhaseCongruency/Docs/convexpl.html). However, theses sub-patterns lack a sound mathematical model. Thus, we have hired a Modified log-Gabor filter approach (already described in [58]) based on Gabor filter for handwritten script identification.

Gabor filters are local and linear band-pass filters in which a sinusoidal plane at a certain orientation and frequency is modulated by a Gaussian envelop. Impulse response of these filters is generated by multiplying a complex oscillation with Gaussian envelope function. 2D Gabor filter function can be written as [20]:

$$\varphi(x,y) = \frac{f^2}{\pi\gamma\omega} e^{-\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\gamma^2}y'^2\right)} e^{j2\pi f x'} \tag{1}$$

where,

$$\begin{aligned} x' &= \quad x\cos\theta + y\sin\theta \\ y' &= -x\cos\theta + y\sin\theta \end{aligned}$$

In spatial domain (Eq. (1)), Gabor filter is the product of a complex plane wave (a 2D Fourier basis function) and an origin-centered Gaussian. Here, f is the central frequency of the filter, θ is the rotation angle, $\gamma$is sharpness (bandwidth) along the Gaussian major axis, and ω is sharpness along the minor axis (perpendicular to the wave). In the given form, the aspect ratio of the Gaussian which is denoted by $1/\gamma$. This function, in frequency domain, takes the following analytical form (http://www.csse.uwa.edu.au/~pk/research/matlabfns/PhaseCongruency/Docs/convexpl.html):

$$\varphi(u,v) = e^{\frac{-\pi^2}{f^2}\left(\gamma^2\left(u'-f\right)^2 + v'^2\omega^2\right)} \tag{2}$$

where,

$$\begin{aligned} u' &= \quad u\cos\theta + v\sin\theta \\ v' &= -u\cos\theta + v\cos\theta \end{aligned}$$

Gabor filters possess excellent joint localization characteristics in both the spatial and the frequency domains and its convolution kernel is obtained by multiplying a Gaussian and a cosine function. However, most applications that employ Gabor filters require a large bank of filters leading to high computational cost. Additionally, they have two main limitations:-

- Maximum bandwidth of a Gabor filter is limited to approximately one octave.
- Gabor filters are not optimal if one is seeking broad spectral information with maximal spatial localization.

To overcome the above limitations, log-Gabor filter was constructed with arbitrary bandwidth and the bandwidth can be utilized to build a filter with minimal spatial extent. Feature extraction procedure based on our Modified log-Gabor methodology is detailed below:

Consider there are n_s scales and n_o number of orientations, resulting in $n_s \times n_o$ different filters. Let J denotes the Fourier transform of the input word image, $G_{s,o}$ is the Gabor filter at scale s and orientation o, and $V_{s,o}$ is the output of the convolution of $G_{s,o}$ and J.

$$V_{s,o} = J^* G_{s,o} \tag{3}$$

Local responses of each of the Gabor filters can also be represented in terms of amplitude $A_{s,o}(x, y)$ and energy $E_{s,o}(x, y)$ as defined below,

$$A_{s,o} = |V_{s,o}(x,y)| \tag{4}$$

and

$$E_{s,o}(x,y) = |\text{Real}\{V_{s,o}(x,y)\}| - |\ Img\ \{V_{s,o}(x,y)\}| \tag{5}$$

where (x, y) denotes 2D coordinates of a pixel, and *Real* and *Img* denote the real and imaginary parts of the filter responses respectively. Next, we define the median over all orientations for a fixed scale s for $A_{s,o}$ and $E_{s,o}$ as follows:

and

$$A_s(x,y) = \text{median}\{o = 1, 2\ldots, n_o\}\ A_{s,o}(x,y) \tag{6}$$

$$E_s(x,y) = \text{median}\{o = 1, 2\ldots, n_o\}\ E_{s,o}(x,y) \tag{7}$$

Finally, the phase symmetry measure, denoted by η(x,y) is defined as follows:

$$\eta(x,y) = \frac{\sum_{s=1}^{n_s} E_s(x,y)}{\sum_{s=1}^{n_s} A_s(x,y)} \tag{8}$$

For the present work, features based on Modified log-Gabor filter have been extracted for 5 scales (n_s = 1, 2, 3, 4, and 5) and 6 orientations ($n_o = 0^0$, 30^0, 60^0, 90^0, 120^0, and 150^0), where each filter is convolved with the input image to obtain 30(5*6) different representations (response matrices) for a given input image. These response matrices are then converted to feature vectors. Each input image provides us with one feature vector consisting of 30 elements. Application of the Modified log-Gabor filter based approach on a sample handwritten *Devanagari* script word for 5 scales and 6 orientations is shown in Fig. 9.

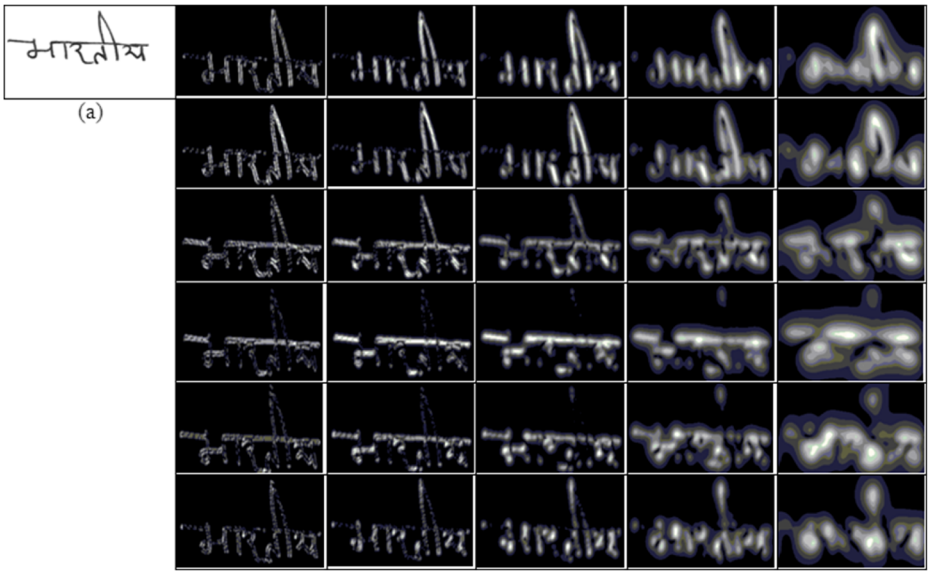


Fig. 9 Illustration of output images after performing Modified log-Gabor filter based approach on a sample handwritten *Devanagari* script word **a** for 5 scales and 6 orientations (The first row shows the output for $n_o = 0^0$ and five scales, the second row shows the output for $n_o = 30^0$ and five scales, and so on)

8 Experimental analysis and discussion

Evaluation of the script separation technique, as discussed above, has been applied on the set of 150 handwritten documents of *CMATERdb*1.2.2 and150 handwritten documents of *CMATERdb*1.5.1. In our experiments, all the schemes are executed in the same environment, i.e., on a PC with an Intel Dual Core processor (2.13 GHz) and 2 GB RAM. In this experiment, *CMATERdb*1.2.2 is named as Dataset#1 and *CMATERdb*1.5.1 is named as Dataset#2. The first part of the experiment involves the extraction of the text words form Datasets#1 and #2 using the technique already described in [59]. For evaluating the performance of the word segmentation algorithm (shown in Table 3), we have considered two types of errors: (a) Over-segmentation (O) and (b) Under-segmentation (U). Denoting the number of actual text words present in a given document page as T, the success rate (SR) of the present technique can be calculated as follows:

$$SR = \left[\frac{(T-(O+U))^{*}100}{T}\right] \tag{9}$$

It is noted from Table 3 that the word extraction algorithm attains segmentation accuracies of 89.65% and 91.27% on Datasets#1 and #2 respectively.

The second part of the experiment focuses on the selection of a suitable classifier for our script recognition algorithm using Modified log-Gabor filter based approach. A 3-fold cross validation scheme has been used for this purpose. For bi-script scenario, a total of 19,507 words (12,620 *Bangla* and 6887 *Roman* words) have been randomly selected from *CMATERdb*2.1.3 and *CMATERdb*2.3.1 for the training purpose whereas the remaining 9755 words (6311 *Bangla* and 3444 *Roman* words) have been chosen for the testing purpose which is named as Dataset#3. A total of 17,239 words (10,352 *Devanagari* and 6887 *Roman* words) have been randomly selected from *CMATERdb*2.2.3 and *CMATERdb*2.3.1for the training purpose whereas the remaining 8620 words (5176 *Devanagari* and 3444 *Roman* words) have been chosen for the testing purpose which is named as Dataset#4. Similarly, for tri-script scenario, a total of 29,859 words (12,620 *Bangla*, 10,352 *Devanagari* and 6887 *Roman* words) have been randomly selected from all the three word databases for the training purpose whereas the remaining 14,931 words (6311 *Bangla*, 5176 *Devanagari* and 3444 *Roman* words) have been taken for the testing purpose, named as Dataset#5. The designed feature set has been individually applied to eight well-known classifiers *namely*, Naïve Bayes, Bayes Net, MLP, SVM, Random

Table 3 Performance evaluation of the word segmentation algorithm on Datasets#1 and #2

Script	Dataset#1		Dataset#2	
	Actual number of words present	Number of words found experimentally	Actual number of words present	Number of words found experimentally
Bangla/Devanagari	18,931	17,360	15,528	14,582
Roman	4878	3985	5453	4567
TOTAL	23,809	21,345	20,981	19,149
SR (%)	89.65		91.27	

Forest, Bagging, MultiClass Classifier and Logistic. A brief description of these classifiers is discussed below:

- **Naïve Bayes:** Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It is called "naive" because it incorporates a simple assumption that attribute values are conditionally independent, given the classification of the instance. Naive Bayes classifiers [48] are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.
- **Bayes Net:** This classifier is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies by means of a directed acyclic graph. Popular Bayesian classifier [30] uses Bayes network learning using different search algorithms and quality parameters. The base class of this classifier provides data structures such as conditional probability distributions, network structure etc. and facilities common Bayes network learning algorithms like K2 and B.
- **MLP:** MLP [2], special kind of Artificial Neural Network (ANN), is a feed-forward layered network of *artificial neurons*. Each artificial neuron in the MLP computes a *sigmoid function* of the weighted sum of all its inputs. An MLP consists of one *input layer*, one *output layer* and a number of *hidden* or intermediate *layers*. Numbers of neurons in the input and the output layers of MLP are mainly chosen as the number of features extracted for the given problem and the number of output classes respectively. Number of neurons in other layers and the number of layers in the MLP are all determined by a trial and error method at the time of its *training*.
- **SVM:** SVM classifier [7] effectively maps pattern vectors to a high dimensional feature space where a 'best' separating hyperplane (the *maximal margin* hyperplane) is constructed. *Maximal margin* results in better generalization and a global solution for the problem, which is a highly desirable property for a classifier to perform well on a novel dataset. Support vector machines are less complex (smaller VC dimension) and perform better (lower actual error) with limited training data. SVM classifier is found to be suitable for most pattern recognition problems having large number of classes and high dimensional input data due to its effective training and testing algorithms and natural extension to the kernel methods. There are number of kernels that can be used in SVM models such as linear, polynomial, radial basis function (RBF) and sigmoid. For the present work, we have implemented RBF based SVM.
- **Random Forest:** A collection or ensemble of simple tree predictors constitute a Random Forest, each capable of producing a response when presented with a set of predictor values. For classification problems, these responses acquire the type of a class membership, which relates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Response of each tree depends on a set of predictor values selected independently (with replacement) and with the similar distribution for all trees in the forest, which is a subset of the predictor values of the original data set. Optimal size of the subset of predictor variables is given by log_2^{M+1}, where M is the number of inputs. Given a set of simple trees and a set of random predictor variables, Random Forest classifier defines a margin function that computes the extent to which average number of

votes for the correct class surpasses the average vote for any other class present in the dependent variable. For more detail refer to [6].

- **Bagging:** Bagging (**B**ootstrap **agg**regat**ing**) classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. For more detail, please refer to [5].
- **MultiClass Classifier:** This classifier [4] is a meta-classifier for handling multi-class datasets with 2-class classifiers. It is also capable of applying error correcting output codes for increased accuracy.
- **Logistic:** It is a classifier for building linear logistic regression model [8]. Here, LogitBoost is used with simple regression function as base learner for fitting logistic model. Optimal number of LogitBoost iterations to be performed, is cross-validated which, in turn, helps in selecting automatic attributes.

Script identification performances of the present technique using each of these classifiers and their corresponding success rates achieved on Datasets#3, #4 and #5are shown in Fig. 10. It can be seen from the figure that the highest script identification accuracies achieved by the present technique are found to be 92.32%, 95.30% and 93.78% on Dataset#3, Dataset#4 and Dataset#5 respectively. The performance analysis involves two parameters *namely*, Model Building Time (MBT) and Recognition Time (RT). MBT is defined as a parameter which is measured based on the time required to train the system on the given training samples and RT is defined as a parameter which is measured based on the time required to recognize (test) the test set samples. MBT and RT required by the above mentioned classifiers for all three datasets are shown in Figs. 11(a-b). Recognition accuracy of the method is estimated by the following equation:

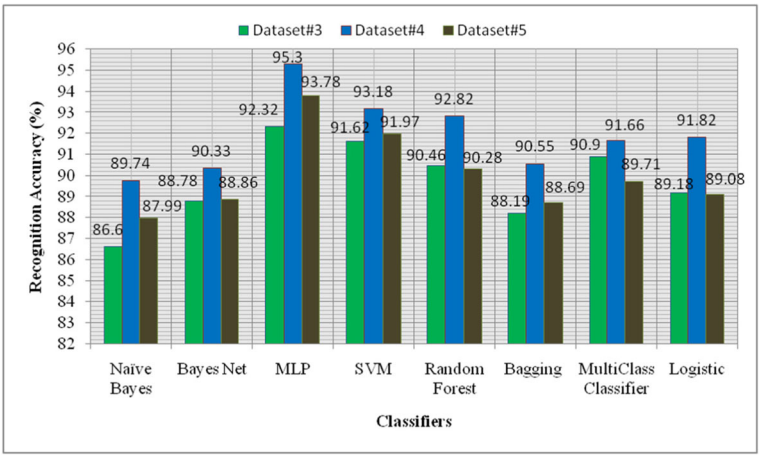
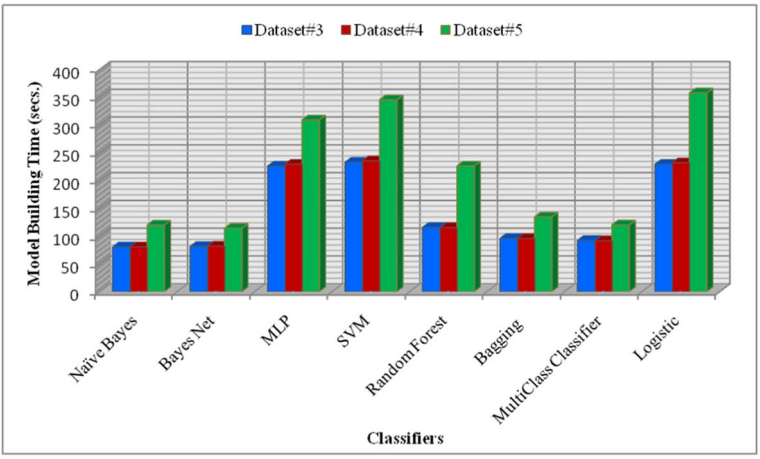
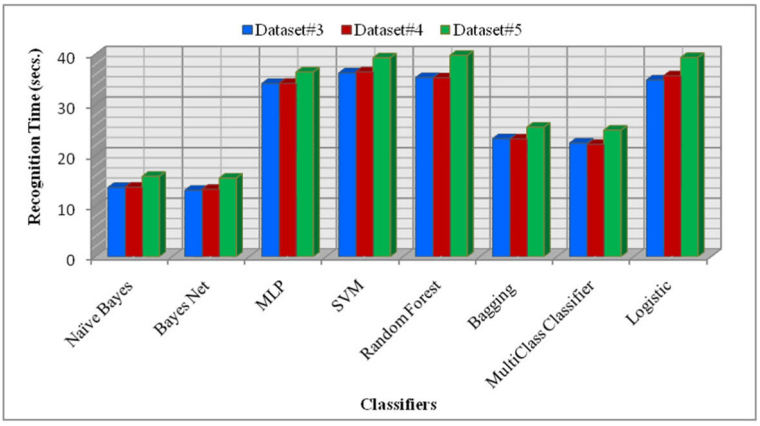


Fig. 10 Graph showing the recognition accuracies of the proposed script identification technique using eight classifiers in bi-script scenario on Dataset#3, Dataset#4 and Dataset#5



(a)



(b)

Fig. 11 Graphical comparison of: **a** Model Building Time (MBT), and **b** Recognition Time (RT) required by eight different classifiers on Dataset#3, Dataset#4 and Dataset#5

$$\text{Recognition Accuracy} = \frac{\#\text{Correctly classified words}}{\#\text{Total words}} \times 100\% \tag{10}$$

8.1 Statistical Significance tests

Statistical significance of the present experimental setup has also been measured as an essential part for validating the performance of multiple classifiers using multiple datasets. For statistical comparison of multiple classifiers, two or more classifiers are first trained and tested on a suitable set of datasets and then their classification accuracies are evaluated. A large dataset is randomly divided to create small datasets with different *sample sizes*. Performances of different classifiers are then carried out for each randomly created dataset. The only requirement for performing non-parametric tests is that the compiled results provide reliable estimates

of the classification algorithms' performances on each dataset [54]. In the usual experimental setups, these numbers come from cross-validation or from repeated stratified random splits onto training and testing datasets. The term "*sample size*" refers to the number of datasets used, not the number of training/testing samples taken from each individual set. *Sample size* can therefore lies between 5 and 30.

To do so, we have performed a safe and robust non-parametric Friedman test [21] with corresponding post-hoc tests. For the experimentation on Dataset#3, number of randomly selected datasets (N) and number of classifiers (k) are set as 12 and 8 respectively. Performances of the classifiers on different datasets are shown in Table 4. On the basis of these performances, classifiers are then ranked for each dataset separately, and the best performing algorithm gets rank 1, second best gets rank 2, and so on (see Table 4). Average ranks are assigned in case of ties.

Let r_j^i be the rank of j^{th} classifier on i^{th} dataset. Then, the mean of ranks of the j^{th} classifier over all the N datasets will be computed as:

$$R_j = \frac{1}{N}\sum_{i=1}^{N} r_j^i \tag{11}$$

The null hypothesis states that all the classifiers are equivalent and so their ranks R_j should be equal. To justify it, Friedman statistic [21] is computed as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{12}$$

Under the current experimentation, this statistic is distributed according to χ_F^2 with k-1(=7) degrees of freedom. Using Eq. (12), value of χ_F^2 is calculated as 26.075. From the table of critical values [available in any standard statistical book], value of χ_F^2 with 7 degrees of

Table 4 Recognition accuracies of 8 classifiers and their corresponding ranks using 12 different datasets (ranks in parentheses are used for performing Friedman test)

	Recognition Accuracy (%)							
	Classifiers							
Set	Naïve Bayes	Bayes Net	MLP	SVM	Random Forest	Bagging	MultiClass Classifier	Logistic
#1	86(8)	92(7)	100(1)	99(2.5)	97(6)	99(2.5)	98(4.5)	98(4.5)
#2	99(3)	98(5.5)	100(1)	99(3)	96(7.5)	96(7.5)	97(5.5)	99(3)
#3	98(5)	94(7)	100(1)	93(8)	99(2.5)	98(5)	99(2.5)	98(5)
#4	93(8)	94(7)	99(1.5)	98(3.5)	98(3.5)	97(5)	96(6)	99(1.5)
#5	91(8)	92(7)	100(1.5)	99(3.5)	99(3.5)	97(6)	98(5)	100(1.5)
#6	99(2.5)	98(4.5)	100(1)	96(7)	97(6)	98(4.5)	99(2.5)	95(8)
#7	91(8)	94(7)	100(1)	99(3)	99(3)	99(3)	98(5.5)	98(5.5)
#8	91(8)	96(7)	100(1)	98(4.5)	98(4.5)	97(6)	99(2.5)	99(2.5)
#9	92(8)	94(7)	100(1.5)	100(1.5)	97(6)	99(4)	99(4)	99(4)
#10	99(2.5)	97(6.5)	100(1)	99(2.5)	97(6.5)	97(6.5)	98(4)	97(6.5)
#11	94(8)	96(7)	100(1)	98(5)	99(3)	98(5)	99(3)	99(3)
#12	98(4)	98(4)	100(1)	97(7)	98(4)	99(2)	97(7)	97(7)
Mean Rank	$R_1 = 6.08$	$R_2 = 6.37$	R_3=1.125	R_4=4.25	R_5=4.67	R_6=4.75	R_7=4.33	R_8=4.33

freedom is 14.0671 for $\alpha = 0.05$ (where α is known as level ofsignificance). It can be seen that the computed χ_F^2 differs significantly from the standard χ_F^2. So the null hypothesis is rejected.

Iman et al. [27] derived a better statistic using the following formula:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{13}$$

F_F is distributed according to F-distribution with k– 1 (=7) and $(k-1)$(N − 1) (=77) degrees of freedom. Using Eq. (13), value of F_F is calculated as 4.952.Critical value of F (7, 77) for α =0.05 is 2.147 [see any standard statistical book] which shows a significant difference between the standard and calculated values of F_F. Thus, both Friedman and Iman et al. statistics reject the null hypothesis.

As the null hypothesis is rejected, Nemenyi test [34], a post-hoc test, is carried out for pair-wise comparisons of the best and worst performing classifiers. Performances of two classifiers significantly differ if the corresponding average ranks differ by at least the critical difference (*CD*) which is expressed as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{14}$$

For Nemenyi's test, value of $q_{0.05}$ for eight classifiers is 3.031 (see Table 5a of [14]). So, *CD* is calculated as $3.031\sqrt{\frac{8\times9}{6\times12}}$ i.e. 3.031 using Eq. (14). Since, the difference between mean

Table 5 Detailed results of the present script recognition technique using MLP classifier on: **a** Dataset#3, **b** Dataset#4 and **c** Dataset#5

Script	Fold#1		Fold#2		Fold#3	
	Number of words trained	Number of words tested	Number of words trained	Number of words tested	Number of words trained	Number of words tested
(a)						
Bangla	13,157	5774	13,106	5825	11,599	7332
Roman	3508	1370	2917	1961	3351	1547
TOTAL	16,665	7144	16,023	7786	14,950	8879
Recognition Accuracy for test case (%)	91.12		91.35		**92.32**	
(b)						
Devanagari	10,790	4738	9673	5855	10,593	4935
Roman	3738	1715	3693	1760	3475	1978
TOTAL	14,528	6453	13,366	7615	14,068	6913
Recognition Accuracy for test case (%)	94.2		**95.30**		93.86	
(c)						
Bangla	13,157	5774	13,106	5825	11,599	7332
Devnagari	10,790	4738	9673	5855	10,593	4935
Roman	7246	3085	6610	3721	6826	3525
TOTAL	31,193	13,597	29,389	15,401	29,018	15,792
Recognition Accuracy for test case (%)	92.95		92.56		**93.78**	

Best accuracy for each case is highlighted in bold

ranks of the best and worst classifiers is much greater than the *CD*, we can conclude that there is a significant difference between the performing ability of the classifiers. For comparing all classifiers with a *control classifier* (say MLP), we have applied Bonferroni-Dunn test [18]. For this test, *CD* is calculated using the same Eq. (14). But here, the value of $q_{0.05}$ for 8 classifiers is 2.690 (see Table 5(b) of [14]). So, *CD* for Bonferroni-Dunn test is calculated as $2.690\sqrt{\frac{8\times9}{6\times12}}$ i.e. 2.690. As the difference between the mean ranks of any classifier and MLP is always greater than *CD*, so the chosen control classifier performs significantly better than other classifiers for Dataset#3.A graphical representation of the above mentioned post-hoc tests for comparison of seven different classifiers on Dataset#3 is shown in Fig. 12. Similarly, it can also be shown for Dataset#4 and Dataset#5, that the chosen classifier (MLP) performs significantly better than the other seven classifiers.

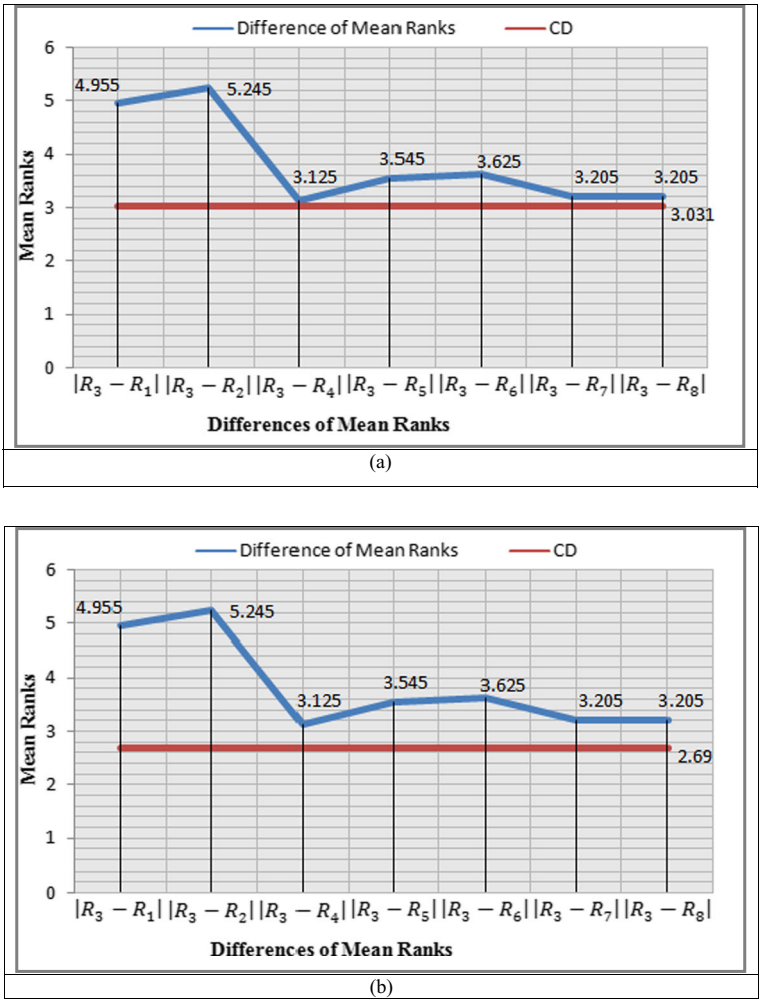


Fig. 12 Graphical representation of comparison of multiple classifiers for: **a** Nemenyi's Test and **b** Bonferroni-Dunn's Test

8.2 Detailed evaluation of MLP classifier

After performing above mentioned statistical significance tests over the 12 datasets and eight classifiers, we can conclude that MLP outperforms all other classifiers for all the three datasets. So, MLP classifier has been chosen for exhaustive testing by tuning its different parameters. For designing the requisite model for each of the MLP based classifiers, several runs of Back Propagation learning algorithm with learning rate (η) = 0.6 and momentum term (α) = 0.7 are executed for different number of neurons in its hidden layer.

The model is trained for 1000 iterations. For the experiment, each dataset (i.e., *CMATERdb*1.2.2 and *CMATERdb*1.5.1) is divided into 3 subsets and testing is done on each subset using rest of the subsets for learning. That is, for the first subset, the training is done with the text words extracted from the document pages 1 to 100 and testing is done with the remaining pages 101 to 150. The second subset of the experiment involves the selection of text words from the document pages 1 to 50 and 101 to 150 while testing is done with the remaining pages 51 to 100. Finally, for the third subset of the experiment, the selection of text words is done from the document pages 51 to 150 while testing is done with the remaining pages 1 to 50. The accuracies of three different runs of script identification scheme on test sets of Datasets #3, #4 and #5 are detailed in Tables 5a, b and c respectively. In the present work, detailed error analysis with respect to different parameters *namely*, Kappa statistics, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, F-measure, Matthews Correlation Coefficient (MCC) and Area Under ROC (AUC) are computed. Table 6 provides a statistical performance analysis with respect to ten parameters for each of the above mentioned datasets.

The overall performances of the technique applied on both the databases are also shown in Table 7. Fig. 13 shows some samples of successful script identification of different scripts taken from both the databases.

In concluding part of the experiment, the handwritten text words present in the said databases have been classified into three types depending on the number of the characters present in a word image. These are: (a) *small-sized words*, (b) *middle-sized words*, and (c) *large-sized words*. If the number of characters present in any word image is less than 3, then it is termed as *small-sized words* whereas if the number of characters lies between 3 and 5, it is called *middle-sized words*. Again, if the number of characters is found to be more than 5, it is labeled as *large-sized words*. Counting of the characters present in each word image has been performed manually in our laboratory. Based on this counting, the text words present in the databases are grouped into the above-said three classes for each of the three above mentioned databases. For each of these databases, the number of words in each class is shown in Table 8. Same script identification algorithm is again applied to them individually and the recognition is done with cross-validation scheme using MLP classifier. Recognition accuracies recorded for each of the word classes are detailed in Table 9.

8.3 Comparison with other state-of-the-art works

For comparison of the present work with some recent works, proposed feature sets as described in [12, 25, 49, 51, 54, 57] have been implemented and evaluated on the developed databases. We have also measured the computational time of the feature extraction. In the experiments, all the schemes are executed in the same environment, i.e., using MATLAB R2009a on a PC with an Intel Dual Core processor (2.13 GHz) and 2 GB memory. From the

Table 6 Statistical performance measures achieved by the proposed technique on Dataset#3, Dataset#4 and Dataset#5

		Kappa Statistics	MAE	RMSE	TPR	FPR	Precision	Recall	F-measure	MCC	AUC
Dataset#3	*Bangla*	0.7341	0.1604	0.2438	0.990	0.337	0.920	0.990	0.954	0.751	0.993
	Roman				0.663	0.010	0.944	0.663	0.779	0.751	0.924
Dataset#4	*Devanagari*	0.8599	0.0835	0.1934	0.986	0.159	0.955	0.986	0.970	0.862	0.989
	Roman				0.841	0.014	0.944	0.841	0.890	0.862	0.944
Dataset#5	*Bangla*	0.9033	0.1069	0.1898	0.960	0.068	0.911	0.960	0.935	0.886	0.984
	Devanagari				0.976	0.017	0.968	0.976	0.972	0.957	0.997
	Roman				0.839	0.015	0.945	0.839	0.889	0.860	0.967

Table 7 Overall best case performances of MLP classifier on all the three Datasets

Combination of scripts	Dataset	Language/scriptsused	Number of words Trained	Number of words Tested	Success Rates (%)
Bi-Script Scenario	#3	*Bangla-English*	14,950	8879	92.32
	#4	*Devanagari-English*	13,366	7615	95.30
Tri-script Scenario	#5	*Bangla-Devanagari-English*	29,018	15,792	93.78

outcome (see Table 10), it is noted that the current feature set not only gives higher identification accuracies but it is also very fast compared to other methods. So, it may be concluded from the result that the proposed technique outperforms the previous ones.

8.4 Error analysis

It is evident from Table 7 that only few words from Dataset#4 are misclassified during testing. This can be due to discontinuities in Matra and poor quality of documents due to presence of noise. Sample word images written in *Devanagari* script are shown in Figs. 14c-d. On the other hand, comparatively low accuracy has been observed for the word images present in Dataset#3. Errors are also observed when there is overwriting. Due to structural similarity in some words, high rate of error is observed in these words. For example, the word seen in Fig. 14f is actually a *Roman* script word "*or*" but it is very much similar to *Bangla* script word "*Baa*". This is why, the said *Roman* script word image is misclassified as *Bangla* script word. Some words are written in structurally different ways**,** depending on the educational and regional background of the writer. For example, no Matra like component is found in word images of Figs. 14a and c, written in *Bangla* script whereas the same is found in the word image of Fig. 14e, written in *Roman* script and these are misclassified among each other. Thus sample word images written in *Bangla* script, shown in Figs. 14a-b, are misclassified as *Roman* script. Few *Roman* script words are also

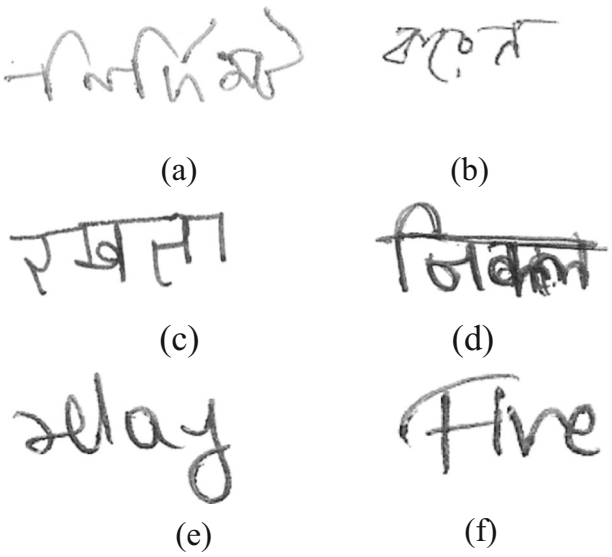


Fig. 13 Sample images of successful script identification of **a-b** *Bangla* script, **c-d** *Devanagari* script, **e-f** *Roman* script

Table 8 Number of *small-sized*, *middle-sized* and *large-sized words* present in the databases *CMATERdb*2.1.3, *CMATERdb*2.2.3 and *CMATERdb*2.3.1

Database	Number of *small-sized words*	Number of *middle-sized words*	Number of *large-sized words*
*CMATERdb*2.1.3	3137	9291	6503
*CMATERdb*2.2.3	3225	7556	5137
*CMATERdb*2.3.1	1995	6062	2604

misclassified (see Figs. 14e-f). This is due to existence of Matra like component in the word for which the extracted feature values are almost similar to those for the words written in *Devanagari*/*Bangla* scripts. In addition, presence of some small components found in the upper part of *Devanagari*/*Bangla* script misclassifies them into *Roman* script or vice-versa. Apart from this, misclassification is mostly seen in the categories of *small-sized* and *middle-sized* words rather than in *large-sized* words. This may be due to the fact that the feature values extracted from such classes of words may not be sufficient enough for the script identification purpose.

9 Conclusion

In this paper, development of benchmark databases for unconstrained handwritten document pages containing both *Bangla-Roman* (Dataset#1) and *Devanagari-Roman* (Dataset#2) mixed-script words are reported. Dataset#2 is first of its kind in this domain of application, i.e., OCR of handwritten *Devanagari* script mixed with *Roman* script. In addition, the second version of Dataset#1 containing 150 handwritten document pages containing *Bangla* mixed with *Roman* script words has been provided. Each document contains characters, text, digits, and other symbols written by different persons. Despite many research efforts in this domain, availability of standard benchmark dataset is limited for *Devanagari*/*Bangla* script. The current work also assessed our word segmentation algorithm on mixed-script document pages written in *Bangla*/*Devanagari* mixed with *Roman* script and we have attained reasonable segmentation accuracies of 89.65% and 91.27% on both developed datasets respectively. We have also evaluated Modified log-Gabor filter based feature extractor for identifying the scripts in mixed-script text documents using MLP based classifier and the script identification accuracies on these handwritten document pages in bi-script and tri-script scenarios are also reported here. Apart from this, we have also provided the word-level ground truth annotations of both the databases

Table 9 Recognition performances of MLP classifier for bi-script and tri-scenarios on all the three datasets (best case for each class is styled in bold and shaded in grey)

Dataset	Success Rates (%)								
	Small-sized words			Middle-sized words			Large-sized words		
	3-Fold	5-Fold	7-Fold	3-Fold	5-Fold	7-Fold	3-Fold	5-Fold	7-Fold
#3	88.57	**89.78**	89.05	90.21	**91.75**	90.65	**94.52**	94.34	93.86
#4	91.2	**92.59**	91.36	93.68	**94.01**	93.25	95.89	**95.95**	95.24
#5	**91.98**	91.37	90.75	**94.95**	94.55	94.2	**93.67**	93.5	93.09

Table 10 Performance comparison of the proposed script identification technique with some state-of-the-art techniques

Researchers	Feature Set	Feature Dimension	Recognition Accuracy (%)			Average Computational Time (sec.)
			Dataset#3	Dataset#4	Dataset#5	
Hiremath et al. [25]	Wavelet based co-occurrence histogram	32	83.47	86.233	85.36	801.244
Singh et al. [57]	Gray Level Co-occurrence Matrix (GLCM) features	80	85.05	88.9	86.29	1265.046
Chaudhari et al. [12]	Gabor filter based features	30	82.24	86.95	84.8	849.478
Sarkar et al. [49]	Holistic features	8	75.5	79.02	78.69	861.207
Singh et al. [51]	Topological and Convex hull based features	39	80.64	84.85	82.55	1076.575
Pardeshi et al. [54]	Radon transform, Discrete wavelet transform, Statistical filters and Discrete cosine transform	46	84.49	89.66	87.75	955.803
Proposed Method	Modified log-Gabor filter based features	30	92.32	95.30	93.78	759.860

which are available *freely* in public domain. Improvement of the ground truth generation software by including the text line extraction routine and performance evaluation metrics are also in our future plans of research. Moreover, some additional techniques must also be devised which will be integrated with the existing scheme in order to recognize the misclassified small and middle-sized script word images.

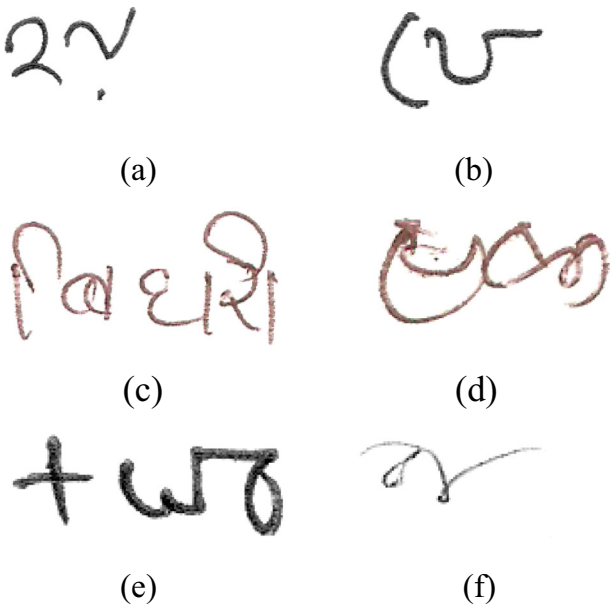


Fig. 14 Sample images of unsuccessful script identification of **a-b** *Bangla* script (misclassified as *Roman* script), **c-d** *Devanagari* script (misclassified as *Roman* script), and, **e-f** *Roman* script (misclassified as *Bangla* script)

In future releases of the database, our aim is to increase the database quantity consisting of document pages written in purely *Devanagari* script and may also include other Matra-based scripts like *Gurumukhi, Gujarati, Oriya* etc. and collect other possible mixed-script document pages. In short, we have attempted to provide databases for the researchers interested in a challenging problem domain, related to mixed-script OCR systems of unconstrained hand-written document pages containing *Devanagari/Bangla* texts mixed with *Roman* script words.

Acknowledgements The authors are thankful to the *CMATER* and Project on Storage Retrieval and Understanding of Video for Multimedia (SRUVM) of Computer Science and Engineering Department, Jadavpur University, for providing infrastructure facilities during progress of the work. The current work, reported here, has been partially funded by University with Potential for Excellence (UPE), Phase-II, UGC, Government of India. Also a lot of people helped us to make the database worthy to use. Authors are grateful to everyone who contributed with data to make this project successful.

References

1. Alaei A, Nagabhushan P, Pal U (2011) A benchmark Kannada handwritten document dataset and its segmentation. In: Proc. of 12th IEEE International Conference on Document Analysis and Recognition (ICDAR), pp 141–145
2. Basu S, Das N, Sarkar R, Kundu M, Nasipuri M, Basu DK (2005) An MLP based approach for recognition of handwritten Bangla numerals. In: Proc. of 2nd International Conference on Artificial Intelligence, pp 407–417
3. Bhattacharya U, Chaudhuri BB (2009) Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE Trans Pattern Anal Mach Intell 3(3):444–457
4. Bishop CM (2006) Pattern recognition and machine learning. In: Information Science and Statistics. Springer Publishers, New York
5. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
6. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
7. C-Chang C, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3) article no. 27
8. le Cessie S, van Houwelingen JC (1992) Ridge estimators in logistic regression. Appl Stat 41(1):191–201
9. Chanda S, Pal U (2005) English, Devnagari and Urdu text identification. In: Proc. of International Conference on Cognition and Recognition, pp 538–545
10. Chanda S, Pal S, Pal U (2008) Word-wise Sinhala, Tamil and English script identification using Gaussian kernel SVM. In: Proc. of 19th IEEE International Conference on Pattern Recognition, pp 1–4
11. Chanda S, Pal S, Franke K, Pal U (2009) Two-stage approach for word-wise script identification. In: Proc. of 10th international Conference on document analysis and recognition (ICDAR), pp 926–930
12. Chaudhari S, Gulati RM (2016) Script identification using Gabor feature and SVM classifier. In: Proc. of International Conference on Communication, Computing and Virtualization, Procedia Computer Science, vol 79, pp 85–92
13. Chaudhuri BB (2006) A complete handwritten numeral database of Bangla—a major Indic script. In: Proc. of 10th International Workshop on Frontiers of Handwriting Recognition, La Baule, France, pp 379–384
14. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
15. Dhandra BV, Nagabhushan P, Hangarge M, Hegadi R, Malemath VS (2006) Script identification based on morphological reconstruction in document images. In: Proc. of IEEE International Conference of Pattern Recognition, Hong Kong, vol 2, pp 950–953
16. Dhandra BV, Mallikarjun H, Hegadi R, Malemath VS (2006) Word-wise script identification from bilingual documents based on morphological reconstruction. In: Proc. of 1st IEEE International Conference on Digital Information Management, pp 389–394
17. Dhanya D, Ramakrishnan AG, Pati PB (2002) Script identification in printed bilingual documents. Sadhana 27(1):73–82
18. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56:52–64
19. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of 2nd International Conference on Knowledge Discovery and Data Mining, vol 96, pp 226–231

20. Field DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. J Opt Soc Am A 4:2379–2394
21. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32:675–701
22. Gonzalez RC, Woods RE (1992) Digital Image Processing, 1st Edn. Prentice-Hall, India
23. Harris C, Stephens M (1988) A combined corner and edge detector. In: Alvey vision Conference, vol 15
24. Hassan E, Garg R, Chaudhury S, Gopal M (2011) Script based Text Identification: A Multi-level Architecture. In: Proc. of the 2011 Joint Workshop on multilingual OCR and analytics for noisy unstructured text data. Beijing, China
25. Hiremath PS, Shivashankar S (2008) Wavelet based co-occurrence histogram features for texture classification with an application to script identification in document image. Pattern Recogn Lett 29(9):1182–1189
26. Hiremath PS, Shivshankar S, Pujari JD, Mouneswara V (2010) Script identification in a handwritten document image using texture features. In: Proc. of 2nd IEEE International Conference on Advance Computing, pp 110–114
27. Iman RL, Davenport JM (1980) Approximations of the critical region of the Friedman statistic. Commun Stat 9(6):571–595
28. Jayadevan R, Kohle SR, Patil PM (2011) Database development and recognition of handwritten Devanagari legal amount words. In: Proc. of 12th IEEE International Conference on Document Analysis and Recognition, pp 304–308
29. Jindal M, Hemrajani N (2013) Script identification for printed document images at text-line level using DCT and PCA. IOSR J Comput Eng 12(5):97–102
30. John GH, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Proc. of 11th Conference on Uncertainty in Artificial Intelligence, San Mateo, pp 338–345
31. Joshi GD, Garg S, Sivaswamy J (2006) Script identification from Indian documents. In: Lecture Notes in Computer Science: International Workshop Document Analysis Systems, Nelson, LNCS-3872, pp 255–267
32. Languages spoken by more than 10 million people. Encarta Encyclopedia (2007) Retrieved 3 Aug 2016
33. Moravec H (1980) Obstacle avoidance and navigation in the real world by a seeing robot rover. In: Tech report CMU-RI-TR-3 Carnegie-Mellon University, robotics institute
34. Nemenyi PB (1963) Distribution-free multiple comparisons. PhD thesis, Princeton University
35. Nethravathi B, Archana CP, Shashikiran K, Ramakrishnan AG, Kumar V (2010) Creation of a huge annotated database for Tamil and Kannada OHR. In: Proc. of International Conference on Frontiers in Handwriting Recognition (ICFHR), pp 415–420
36. Obaidullah SM, Kundu SK, Roy K (2013) A system for handwritten script identification from Indian document. J Pattern Recognit Res 8:1–12
37. Padma MC, Vijaya PA (2009) Identification of Telugu, Devnagari and English scripts using discriminating features. Int J Comp Sci Inf Technol 1(2):64–78
38. Padma MC, Vijaya PA (2010) Global approach for script identification using wavelet packet based features. Int J Sig Process, Image Process Pattern Recognit 3(3):29–40
39. Padma MC, Vijaya PA (2010) Script identification from trilingual documents using profile based features. Int J Comput Sci Appl (IJCSA) 7(4):16–33
40. Padma MC, Vijaya PA (2010) Script identification of text words from a tri lingual document using voting technique. Int J Image Process 4(1):35–52
41. Pal U, Chaudhuri BB (1997) Automatic separation of words in multi lingual multi script Indian documents. In: Proc. of 4th IEEE International Conference on Document Analysis and Recognition (ICDAR), pp 576–579
42. Pal U, Sinha S, Chaudhuri BB (2003) Multi-script line identification from Indian documents. In: Proc. of 7th IEEE International Conference on Document Analysis and Recognition (ICDAR), pp 880–884
43. Pal U, Sharma N, Wakabayashi T, Kimura F (2007) Handwritten numeral recognition of six popular Indian scripts. In: Proc. of 9th IEEE International Conference on Document Analysis and Recognition (ICDAR), pp 749–753
44. Pardeshi R, Chaudhuri BB, Hangarge M, Santosh KC (2014) Automatic handwritten Indian scripts identification. In: Proc. of 14th International Conference on Frontiers in Handwriting Recognition, pp 375-380
45. Pati PB, Ramakrishnan AG (2006) HVS inspired system for script identification in Indian multi-script documents. In: Lecture Notes in Computer Science: International Workshop Document Analysis Systems, Nelson, LNCS-3872, pp 380–389
46. Pati PB, Ramakrishnan AG (2008) Word level multi-script identification. Pattern Recogn Lett 29(9):1218–1229
47. Patil SB, Subbareddy NV (2002) Neural network based system for script identification in Indian documents. Sadhana 27(1):83–97

48. Rish I (2001) An empirical study of the naive Bayes classifier. In: IJCAI Workshop on Empirical Methods in AI
49. Roy K, Pal U (2006) Word-wise handwritten script separation for Indian postal automation. In: Proc. of 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, pp 521–526
50. Roy K, Das SK, Obaidullah Sk Md (2011) Script identification from handwritten documents. In: Proc. of 3rd IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Hubli, Karnataka, pp. 66–69
51. Sarkar R, Das N, Basu S, Kundu M, Nasipuri M, Basu DK (2010) Word level script identification from Bangla and Devnagari handwritten texts mixed with Roman scripts. J Comput 2(2):103–108
52. Sarkar R, Das N, Basu S, Kundu M, Nasipuri M, Basu DK (2012) *CMATERdb1*:a database of unconstrained handwritten Bangla and Bangla-English mixed script document image. Int J Doc Anal Recognit 15(1):71–83
53. Singh PK, Sarkar R, Das N, Basu S, Nasipuri M (2013) Identification of Devnagari and Roman scripts from multi-script handwritten documents. In: Proc. of 5th International Conference on pattern recognition and machine Intelligence (PReMI). LNCS 8251:509–514
54. Singh PK, Sarkar R, Das N, Basu S, Nasipuri M (2014) Statistical comparison of classifiers for script identification from multi-script handwritten documents. Int J Appl Pattern Recognit 1(2):152–172
55. Singh PK, Sarkar R, Nasipuri M (2015) Offline script identification from multilingual Indic-script documents: a state-of-the-art. In: Computer Science Review, Elsevier 15–16:1–28
56. Singh PK, Dalal SK, Sarkar R, Nasipuri M (2015) Page-level Script identification from Multi-script handwritten documents. In: Proc. of 3rd IEEE International Conference on Computer, Communication, Control and Information Technology (C3IT), pp 1–6
57. Singh PK, Sarkar R, Nasipuri M (2015) Line-level script identification for six handwritten scripts using texture based features. In: Proc. of 2nd Information Systems Design and Intelligent Applications. Adv Intell Syst Comput 340:285–293
58. Singh PK, Chatterjee I, Sarkar R (2015) Page-level handwritten script identification using Modified log-Gabor filter based features. In: Proc. of 2nd IEEE International Conference on Recent Trends in Information Systems (ReTIS), pp 225–230
59. Singh PK, Chowdhury SP, Sinha S, Eum S, Sarkar R (2017) Page-to-word extraction from unconstrained handwritten document images. In: Proc. of 1st International Conference on Intelligent Computing and Communication ($ICIC^2$), AISC 458, pp. 517-524.

Pawan Kumar Singh received his B. Tech degree in Information Technology from West Bengal University of Technology in 2010. He received his M. Tech in Computer Technology degree from Jadavpur University in 2013. Currently he is pursuing PhD degree in Jadavpur University. His areas of current research interest are Handwriting Script Recognition, Document Image Processing etc.

Ram Sarkar received his B. Tech degree in Computer Science and Engineering from University of Calcutta in 2003. He received his M.C.S.E and PhD (Engg.) degrees from Jadavpur University in 2005 and 2012 respectively. He joined Jadavpur University as an Assistant Professor in 2008. He received Fulbright-Nehru Fellowship (USIEF) for post-doctoral research in University of Maryland, College Park, USA in 2014-15. His areas of current research interest are Document Image Processing, Offline and Online handwriting recognition, Machine Learning, and Soft Computing etc. He is a member of the IEEE, USA

Nibaran Das received his B. Tech degree in Computer Science and Engineering from Kalyani University in 2003. He received his M.C.S.E and PhD (Engg.) degrees from Jadavpur University in 2005 and 2012 respectively. He joined Jadavpur University as an Assistant Professor in 2006. He received EMMA Fellowship for post-doctoral research in University of Evora, Portugal in 2013-14. His areas of current research interest are Document Image Processing, Offline Character Recognition, Machine Learning, and Artificial Intelligence etc. He is a member of the IEEE, USA

Subhadip Basu received his B.E. degree in Computer Science and Engineering from Kuvempu University, Karnataka, India, in 1999. He received his Ph.D. (Engg.) degree thereafter from Jadavpur University in 2006. He joined Jadavpur University in 2006 and currently, his designation is Associate Professor. He received Boyscast Fellowship (Govt. of India) for post-doctoral research in University of Iowa, USA in 2009-10. He also received European Union EMMA-West Post-doctoral fellowship in 2012 for visiting University of Warsaw, Poland. His areas of current research interest are OCR of handwritten text, Gesture recognition, Real-time image processing, Bioinformatics, Machine Learning, Artificial Intelligence etc. He is a senior member of the IEEE, USA

Mahantapas Kundu received his B.E.E, M.E.Tel.E and Ph.D. (Engg.) degrees from Jadavpur University in 1983, 1985 and 1995 respectively. Prof. Kundu has been a faculty member of Jadavpur University since 1988. His areas of current research interest include Pattern Recognition, Image Processing, Multimedia database, and Artificial Intelligence etc.

Mita Nasipuri received her B.E.Tel.E., M.E.Tel.E., and Ph.D. (Engg.) degrees from Jadavpur University in 1979, 1981 and 1990 respectively. Prof. Nasipuri has been a faculty member of Jadavpur University since 1987. Her current research interest includes Image Processing, Pattern Recognition, Machine Learning, Soft Computing and Multimedia Systems etc. She is a senior member of the IEEE, U.S.A., Fellow of I.E (India) and W.B.A.S.T, Kolkata, India.