Pattern Recognition 43 (2010) 3507–3521

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A novel framework for automatic sorting of postal documents with multi-script address blocks

Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri *, Dipak Kumar Basu [1]

Computer Science & Engineering Department, Jadavpur University, Kolkata 700032, India.

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form
10 April 2010
Accepted 13 May 2010

Keywords:
Automatic mail sorting
Multi-script postal address block
Script identification for numerals
Quad-tree based feature extraction
Handwritten numeral recognition
Support vector machine

ABSTRACT

Recognition of numeric postal codes in a multi-script environment is a classical problem in any postal automation system. In such postal documents, determination of the script of the handwritten postal codes is crucial for subsequent invocation of the digit recognizers for respective scripts. The current framework attempts to infer about the script of the numeric postal code without having any bias from the script of the textual address part of the rest of the address block, as they might differ in a potential multi-script environment. Scope of the current work is to recognize the postal codes written in any of the four popular scripts, viz., *Latin*, *Devanagari*, *Bangla* and *Urdu*. For this purpose, we first implement a Hough transformation based technique to localize the postal-code blocks from structured postal documents with defined address block region. Isolated handwritten digit patterns are then extracted from the localized postal-code region. In the next stage of the developed framework, similar shaped digit patterns of the said four scripts are grouped in 25 clusters. A script independent unified pattern classifier is then designed to classify the numeric postal codes into one of these 25 clusters. Based on these classification decisions a rule-based script inference engine is designed to infer about the script of the numeric postal code. One of the four script specific classifiers is subsequently invoked to recognize the digit patterns of the corresponding script. A novel quad-tree based image partitioning technique is also developed in this work for effective feature extraction from the numeric digit patterns. The average recognition accuracy over ten-fold cross validation of results for the support vector machine (SVM) based 25-class unified pattern classifier is obtained as 92.03%. With randomly selected six-digit numeric strings of four different scripts; an average of 96.72% script inference accuracy is achieved. The average of tenfold cross-validation recognition accuracies of the individual SVM classifiers for the *Latin*, *Devanagari*, *Bangla* and *Urdu* numerals are observed as 95.55%, 95.63%, 97.15% and 96.20%, respectively.

1. Introduction

Optical character recognition (OCR) based systems ease the process of automatic translation of document images into equivalent character codes with huge savings of human energy and cost. Such systems may be implemented in variety of applications such as automatic sorting of postal documents, reading handwritten digits from bank cheques, text extraction and recognition of hand filled structured form documents, reading registration numbers from vehicle license plate images, etc. In many of these applications, the problem is compounded with the presence of handwritten text written in different scripts. Automatic mail sorting based on postal identification numbers in multi-script environment is a typical example of one such application.

Postal documents are primarily sorted on the basis of a numeric string, popularly known as PIN (postal identification number) code or ZIP (zone improvement plan) code. For development of an automated mail sorting system, a key challenge is to interpret the handwritten/printed postal code written in different scripts. In a multilingual country like India with 22 official languages [1,2], the postal codes are written in different regional scripts along with the Latin script numerals, used for the English language. Due to India's long colonial past, Latin script has been popular throughout the Indian sub-continent and arguably the most accepted script in India for official, juridical and administrative purposes. Among the other scripts, Devanagari is the most popular script in India and is used to write texts in Hindi, Marathi and Nepali languages. Bangla, the second most popular script in India and the sixth most popular script in the world [3], is widely used in the states of West Bengal, Assam, Tripura and Manipur. It may also be noted that Bangla is the National script of Bangladesh with more than 170 million users worldwide [3]. Urdu is an official language and script in many states of India including Jammu and Kashmir, Andhra Pradesh,

* Corresponding author.
E-mail address: nasipuri@vsnl.com (M. Nasipuri).
[1] A.I.C.T.E. Emeritus Fellow

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.018

Delhi, Bihar, Uttar Pradesh and Uttarakhand. The Urdu script, used by more than 52 million people in India [2], is the right-to-left alphabet used for the Urdu language. It is a modification of the Persian alphabet, which itself is a derivative of the Arabic alphabet. These four scripts, combined together, cover most of the Indian states' population.

In the present work, we have attempted to address the problem related to the interpretation of handwritten pin codes, written in any of the aforementioned four scripts, viz., *Latin*, *Devanagari*, *Bangla* and *Urdu* (*LDBU*). We first identify the specific script in which the numeric postal code is written and then focus on recognition of individual digits of that postal code, using a pre-trained classifier meant for that script. Apart from its relevance in most of the states in India, the designed framework may be applicable for automatic interpretation of multi-script postal codes in many other Asian countries like, Bangladesh, Pakistan, Nepal and Bhutan. Related research contributions in this area can broadly be classified into three categories, as described below.

1.1. First category of solutions

In this category of solutions, a single document page is assumed to be written using a single script. Chaudhury and Sheth [4] had developed a Gabor filter based feature extraction scheme for identification of the script of any digitized text-book page. The technique could successfully separate the printed *Latin, Devanagari* and *Telegu* document pages. Singhal et al. [5] had developed a similar technique using the Gabor filters to classify the handwritten document pages of *Latin, Devanagari, Bangla* and *Telegu* scripts. In two of their recent works, Joshi et al. [6,7] had reported a generalized framework using log-Gabor filters to identify the script of the printed document pages. The technique was evaluated on a diversified dataset of 10 Indian and 13 world scripts with good accuracies.

However, in a single digitized document page multiple scripts may also appear simultaneously in many cases and none of the aforementioned techniques addressed this important issue explicitly.

1.2. Second category of solutions

To address the problem of line-wise script identification in multi-script document pages, Pal et al. [8–12] had developed several projection based script separation techniques for printed text lines of *Latin* and different Indian scripts. In their most current work [12], Pal et al. had developed a feature based approach for automatic identification of *Latin* and different Indian scripts from printed text lines. Horizontal projection profile, water reservoir based features and other structural features were computed for each text line in the document image. A rule-based binary tree classifier was finally designed to separate text lines belonging to 11 different scripts. However, in any practical scenario, multiple scripts may also appear in a single text line. For this purpose, Sinha et al. [13] had developed a word-wise script identification scheme for printed *Latin, Bangla, Devanagari, Malayalam, Telegu* and *Gujrati* scripts using different topological and structural features. All these techniques are also silent over appearances of numeric strings in any text line. As most of these techniques depend heavily on the *headline* or *baseline* profiles of individual text lines or words or other character-level attributes of any script, the appearances of numeric strings in such lines or words are likely to produce erroneous script-separation results.

In any postal address block, multiple scripts may appear simultaneously in a single line, or in a single document. Roy et al. [14] had addressed this issue and extended the work of Sinha et al. [13] to develop a technique for word-wise identification of *Latin, Devanagari* and *Bangla* scripts in handwritten textual postal addresses using similar topological and structural features. However, they had not shown any result on identification of the scripts for the numeric postal codes. In another work, Zhou et al. [15] had developed a connected component profile analysis technique for separation of *Latin* and *Bangla* scripts based postal documents. Other works, reported in the literature, related to postal automations [16,17] did not explicitly address the issue of multiple script identification.

Despite these research contributions, the true issue of multi-script address block interpretation still remains an unsolved problem. This is so because in all these works [14–17], the authors had either assumed that the address blocks, including the numeric postal codes, are written using the same script of the textual address block, or remained silent on the script of the postal codes.

1.3. Third category of solutions

Research contributions on recognition of numerals from digitized handwritten documents written in a single script are plenty in the literature [16–26]. Most of these works focus on feature based recognition of isolated handwritten digit samples of a given script using standard classifiers. In one of our earlier works [21], a two-pass feature based approach was designed for recognition of handwritten numerals of *Bangla* script. In another work [22], a classifier combination scheme was proposed to infer over the decisions taken by two different classifiers on each digit pattern. In one of the recent works, Pal et al. [23], had used contour based directional features to recognize handwritten numerals of six popular Indian scripts, viz., *Devanagari, Bangla, Telugu, Oriya, Kannada* and *Tamil*. They had used six different quadratic classifiers, each for the six different scripts, and obtained good recognition accuracy. In another recent work, Wen et al. [26] had developed a handwritten *Bangla* numeral recognition system for automatic sorting of mails for the Bangladesh Post. Using principal component analysis technique and support vector machine (SVM) classifier, they had achieved high reliability in recognition of handwritten numerals of *Bangla* script. Despite addressing the recognition challenges of a large subset of digit patterns of different popular Indic scripts, the authors remained silent over the script identification technique for the said pattern classes.

1.4. Motivation behind the current work

In general, popular postal stationeries are printed by the Department of Post having specific sizes, formats and structures. The current work deals with only such postal documents having pre-determined address block structures, and is referred as structured postal documents throughout this paper. In most cases, the postal addresses contain two parts; viz., the alphanumeric address block with the name of the addressee, street name, province, country, etc. and a block-structured region to specify the postal code. Postal documents are primarily sorted on the basis of the postal code written on the postal-code block, written in one specific script. In a multi-lingual environment like India, the script(s) of the rest of the alphanumeric address part may or may not match with that of the numeric postal codes. More specifically, people often write postal address in (at most) two scripts, e.g., the textual parts in regional scripts like *Devanagari*, *Bangla* or *Urdu* and the numeric part including the postal code in the *Latin* script. Fig. 1(a–d) shows some sample Indian postal document images with postal codes in *Latin* script, but the rest of

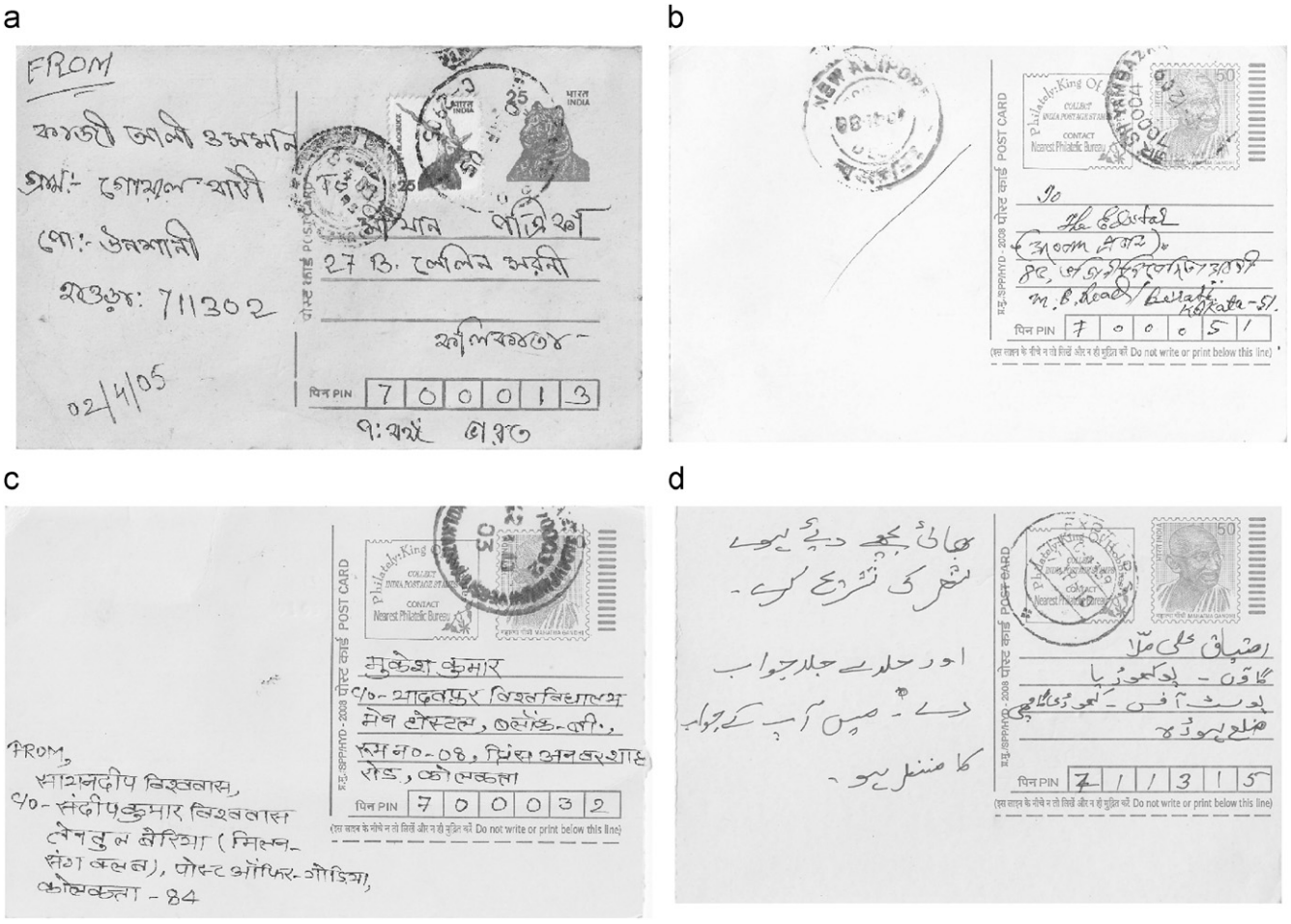

Fig. 1. (a–d) Sample Indian postal document images with multi-script address block, where most of the address block is written in different regional scripts, but the postal codes are written in the *Latin* script.

the address block in one regional script. Fig. 2(a–c) shows some more postal document images where the complete address block including the postal codes are written in three regional scripts under consideration. This leads to a complex and potentially ambiguous situation in any automatic mail sorting application. Available methodologies in the literature are mostly silent on this typical situation that arises in a multi-lingual scenario. Therefore, a special framework needs to be developed to infer about the script of the postal code, without having any bias from the script of the rest of the postal address block. This has been one of our key motivations behind the current work, discussed in this paper.

2. The present work

The objective of the current work is to develop a novel framework for recognition of postal codes written in *Latin*, *Devanagari*, *Bangla* and *Urdu* scripts from multi-script postal address blocks. In this work we first implement a Hough-transformation based pre-processing technique to localize the postal code blocks from digitized images of structured postal documents. Isolated handwritten numeric characters are then extracted from localized postal-code blocks. Fig. 3(a–d) shows sample isolated handwritten digit patterns of the four aforementioned LDBU scripts. In the next stage of the developed framework, similar shaped digit patterns of the said four scripts are grouped in to 25 clusters. Representative digit patterns of four scripts that form these 25 unique clusters are shown in Fig. 4. A script independent unified pattern classifier is then designed to classify any digit pattern of the *LDBU* scripts into one of these 25 clusters. Based on these classification decisions a rule-based script inference engine is designed to infer about the script of the numeric postal code. One of the four script specific pre-trained classifiers is subsequently invoked to recognize the digit patterns of the corresponding script. In the designed multi-stage framework, a novel quad-tree based feature extraction technique is also developed and used with SVM based classifiers in different stages of the recognition process.

A schematic block diagram of the overall system is shown in Fig. 5. Layout analysis of postal documents and segmentation of postal code into isolated numerals is a complex research problem, attempted by many researchers [16,17,19,25]. In the current work, we have developed a pre-processing technique using Hough transformation and modified run-length-smearing-algorithm (RLSA) to localize postal-code blocks from digitized images of structured Indian postal documents. This technique is described in details in the following sub-section. Rest of the paper discusses the novel quad-tree based feature set developed under the current work, the design of the 25-class unified pattern classifier, the rule-based script inference engine and the script-specific numeral recognizers for the LDBU scripts.

2.1. Pre-processing of digitized postal documents

As discussed before, the current experiment focuses on structured postal documents printed by the Department of Post,

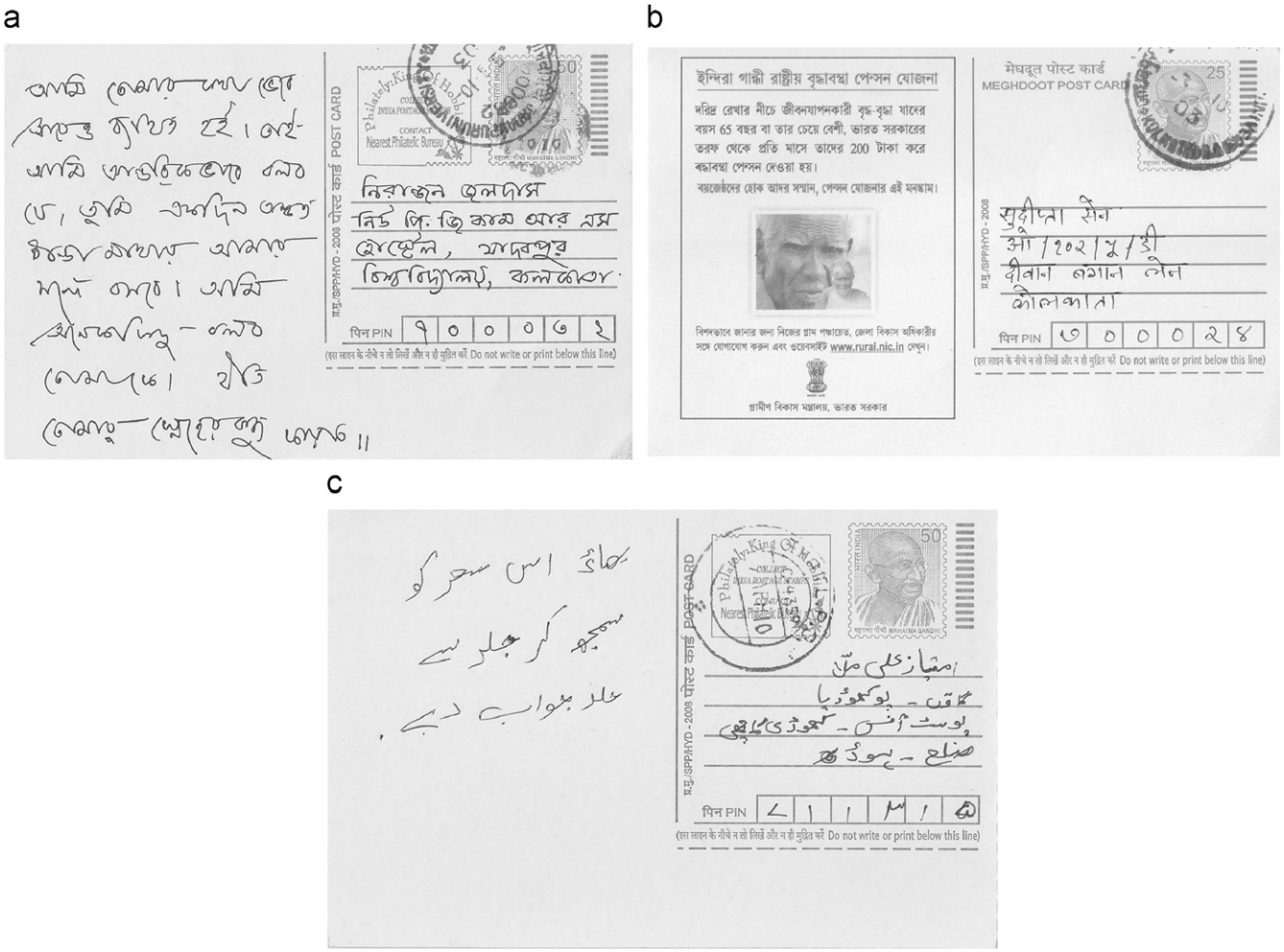

Fig. 2. Sample Indian postal document images where the complete address blocks are written in different regional scripts, (a) Bangla, (b) Devanagari and (c) Urdu script.

Government of India. More specifically, we have experimented with digitized images of postcards, envelops and inland-letters. Our technique is however flexible enough to be implemented on other standard postal document layouts, not considered in this experiment.

Physical postal documents are collected at the 'Center for Microprocessor Applications for Training Education and Research' (CMATER) laboratory, Jadavpur University, from different researchers' personal collections and the editorial offices of three Bengali magazines like 'Computer Jagat', 'Sangbad Prabaha' and 'Meezan'. The obverse sides of all such structured postal documents with filled-in address blocks are digitized as grey scale (8 bits per pixel) bitmap images using a flatbed scanner at a resolution of 300 dpi. Most postal documents under consideration are made of thick papers and align perfectly with the edges of the scanner bed. Therefore, the chances of misalignment/skew were minimal and required no special consideration in our current experimental setup. A sample subset of 50 digitized Indian postal document images is available for download at the CMATER database repository (http://code.google.com/p/cmaterdb/downloads/list) as the CMATER database series 5, version 1 (CMATERdb 5.1).

To localize the postal-code blocks in any digitized postal document image, we analyse the grey scale pixel intensity values of the potential postal-code region. We then suppress the background shading and binarize the card image to retain handwritten text and postal codes. We use a simple adaptive technique to calculate the binarization threshold T, where $T=T_v+T_b$, T_v is the average of maximum and minimum grey scale intensities in the potential postal-code region and T_b is an experimentally chosen fixed *bias*. The threshold is estimated conservatively to retain handwritten texts and the box formatted postal-code regions. We then use Hough transformation technique [27] to identify continuous vertical lines with height within the range $(H_{\min}, H_{\max})$. Such vertical lines appear only in the box-formatted postal-code regions. Localizing such vertical stripes in the postal document images make the identification process for handwritten numerals easier. We then employ a customized RLSA algorithm [27] to horizontally smear such vertical stripes, spaced apart within a range of $(W_{\min}, W_{\max})$ pixels. We then look for six successively smeared regions, each containing one handwritten numeric postal code. Fig. 6(a) shows the grey scale image of an Indian postal document and Fig. 6(b–d) illustrates corresponding results after implementation of the binarization, Hough transformation and run-length smearing algorithms on it. From the illustration, it is evident that the developed technique is robust enough to localize postal-code regions from potentially noisy and cluttered background. The primary reason behind that is the prudent choice of the background elimination thresholds and the Hough transformation parameters.

From each of the six isolated smeared regions, individual digit patterns are extracted using the connected component labeling

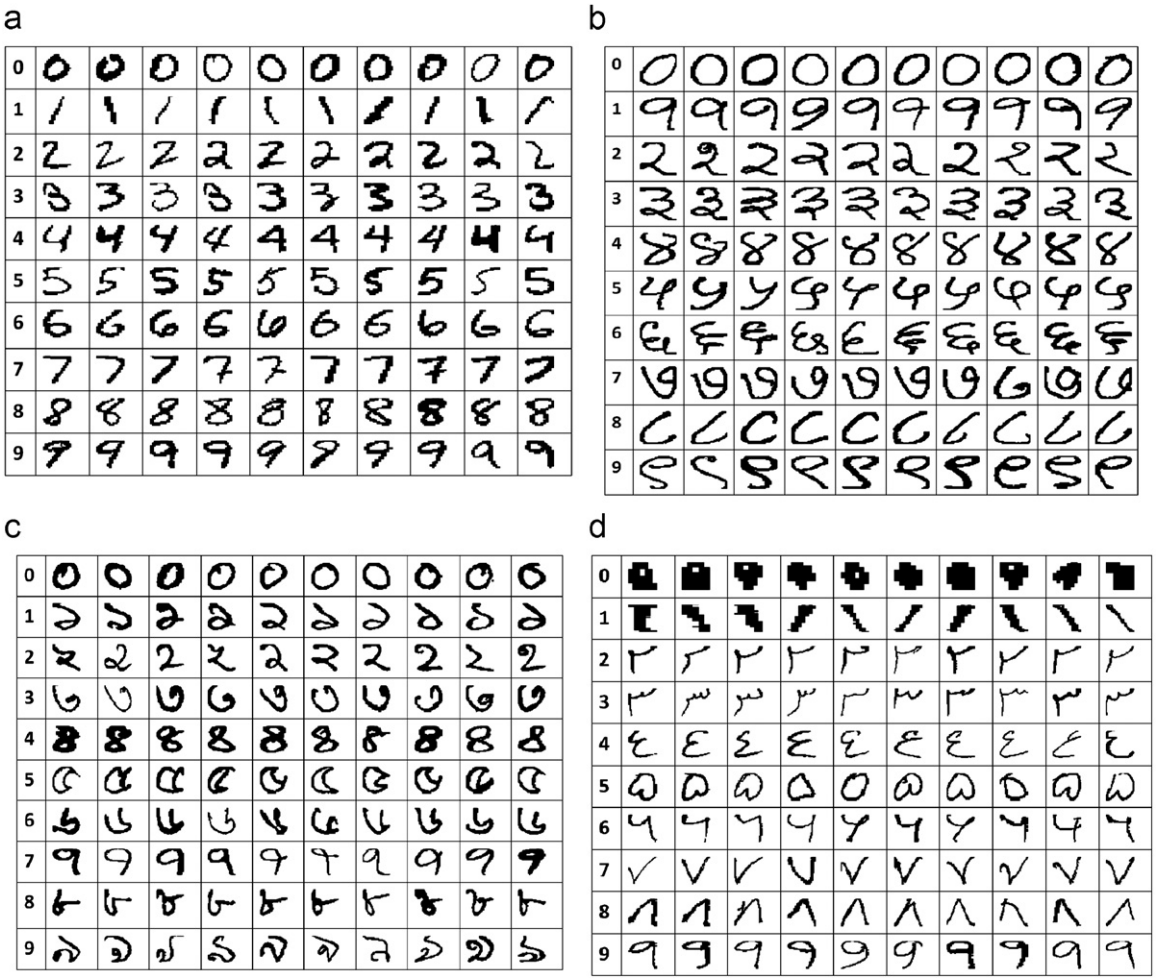

Fig. 3. Variations in handwriting samples of decimal digit sets of four different scripts are shown. (a) *Latin*, (b) *Devanagari*, (c) *Bangla* and (d) *Urdu* scripts with corresponding labels in *Latin* script.

algorithm [27]. It may be worth mentioning in this context that, numeric postal codes are often written in a connected fashion, making the extraction of individual digits difficult. In the current approach, we however consider postal codes written within the specified postal-code boxes. The developed smearing algorithm separates two successive boxes along the vertical lines identified by the Hough transformation technique. Therefore any connected numeral would automatically be truncated along the vertical boundary separating two neighbouring postal-code boxes. Two morphological operations [27], viz., erosion and dilation, are then applied on each digit image to eliminate possible noise pixels in it. The binarized digit images are then enclosed in a rectangular bounding region and finally normalized to a dimension of 32×32 pixels. We then extract quad-tree based longest run features from each such binarized digit image using a novel technique described below.

2.2. Description of the quad-tree based longest-run features

For extraction of the features for both the unified pattern classifier and the multi-script numeral recognition engine, a novel quad-tree based longest-run (QTLR) feature set is designed for the current work. Detailed descriptions of this feature set is given in the following sub-sections.

2.2.1. Longest run features

Within a rectangular image region, longest run features are computed in *four directions*, viz, row wise, column wise and along the directions of two major diagonals. The row wise longest run feature is computed by considering the *sum* of the lengths of the longest bars that fit consecutive black pixels along each of all the rows of the region, as illustrated in Fig. 7(a–b). A 6×6 pixel size sub image is considered here for the sake of simplicity.

In fitting a bar with a number of consecutive black pixels within a rectangular region, the bar may extend beyond the boundary of the region if the chain of black pixels is continued there. The three other longest-run features within the rectangle are computed in the same way. Each of the longest run feature values is to be normalized by dividing it with the product of the height (h) and the width (w) of the entire image. The product, $h \times w$, represents the sum of the lengths of the bars that fit consecutive black pixels individually in each of the four directions within the region completely filled with black pixels.

2.2.2. Quad-tree structure

A quad-tree is a tree data structure in which each internal node has up to four children. Quad-trees are most often used for representation of a two dimensional space by recursively subdividing it into four equal quadrants or regions. In the current work, we have used a modified version of quad tree structure to

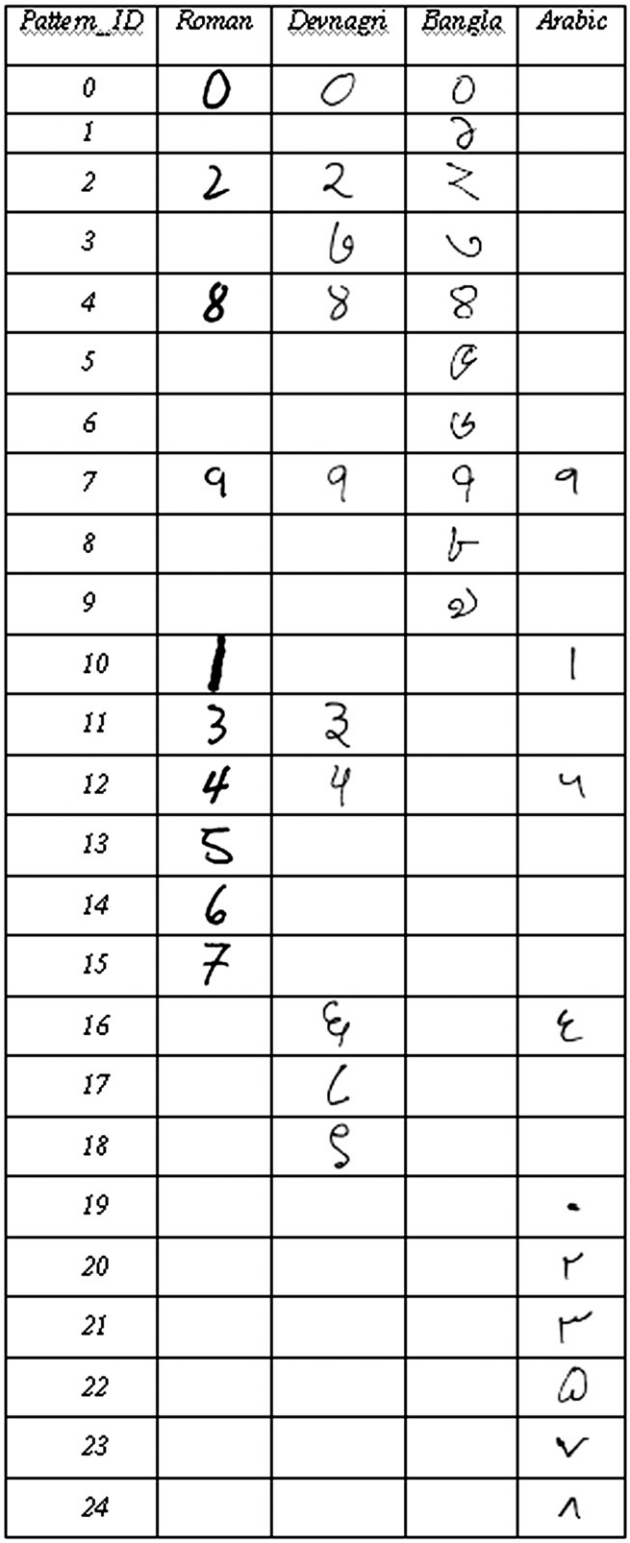

Fig. 4. 25 unique digit patterns are identified from the four *LDBU* scripts.

partition any digit pattern into multiple sub-images. Here, partitioning a digit pattern (or a subpart of it) into 4 regions is done by drawing a horizontal and a vertical line through the center of gravity (CG) of black pixels in that region. If the depth of the quad-tree structure is d, then total number of sub images for each digit pattern at leaf nodes would be 4^d. The coordinates of the CG of any image frame, (C_x, C_y), is calculated as follows:

$$C_x = \frac{1}{mn}\sum_{mn} x.f(x,y); \quad C_y = \frac{1}{mn}\sum_{mn} y.f(x,y)$$

$$f(x,y) = \begin{cases} 1; & \text{for all black pixels} \\ 0; & \text{otherwise} \end{cases}$$

where x and y are the coordinates of each pixel in the image of size $m \times n$ pixels. Fig. 8(a) shows a sample image and Fig. 8(b) shows the CG based partitioning for generating the quad-tree structure of depth 2. For each sub image at the leaf-node of the quad-tree structure, 4 longest-run features are computed. In the current work, we have considered up to the quad-tree structure depth (d) as 2. This generates 4^0, i.e., 1 sub-image at the root node, 4^1, i.e., 4 sub-images at the intermediate node and 4^2, i.e., 16 sub-images at the leaf node positions, thereby generating 21 sub-images at different levels of the hierarchy. This is illustrated with a sample digit image in Fig. 9. We then compute 4 longest-run features for each such sub-image, resulting in $(21 \times 4 = 84)$ QTLR features for any digit pattern.

Partitioning any digit pattern using CG based quad tree structure is a novelty of the current work. Equal partitioning, as usually done in many approaches, often generates less informative sub-images in comparison to the CG based partitioning. A sample digit image with equal partitioning structure is shown in Fig. 8(c). Comparing Figs. 8(b) and (c), it may be observed that the equal partitioning structure generates many sub-images with no information, which is avoided in the current CG based quad-tree structure. It may be worth mentioning in this context that the computation of CG of any pattern is suffered by an obvious *round-off* error. This is because of the approximation of real numbers representing the mathematical CG coordinates to the nearest integer coordinates. The performance of the system however remains unaffected by this apparent compromise.

2.3. Design of a unified pattern classifier

As already mentioned, handwritten digit patterns of the *LDBU* scripts often bear significant similarities in shapes among themselves. In the current work we have identified 25 such unique pattern clusters, as shown in Fig. 4, from the 40 digit patterns of four different scripts under consideration. Although some of the patterns in any given cluster are not exactly similar in shape, but due to their overall structural similarities they are put in the same group. A SVM based classifier is designed as a unified pattern classifier for recognizing these 25 different pattern classes. For this, 84 QTLR features are extracted from each of the pattern classes using a two-level quad-tree structure. The SVM is trained with sample patterns belonging to the 25 classes using the radial-basis-function kernel.

2.4. Design of a rule-based script inference engine

As discussed earlier, 25 unique shapes are identified from the numeric patterns of the *LDBU* scripts. These unique shapes may represent either a single numeral of any given script, or different numerals of different scripts. Considering these possibilities, 25 unique pattern classes are further classified into 11 groups. Each such group may be viewed as a triplet, represented as follows:

{*Group_ID*, (*Set of unique pattern IDs constituting the group*), (*Set of identity of scripts the unique patterns represent*)}.

Descriptions of the observed 11 groups of patterns are given below which are also illustrated in Fig. 10.

{0, (1, 5, 6, 8, 9), (B)}

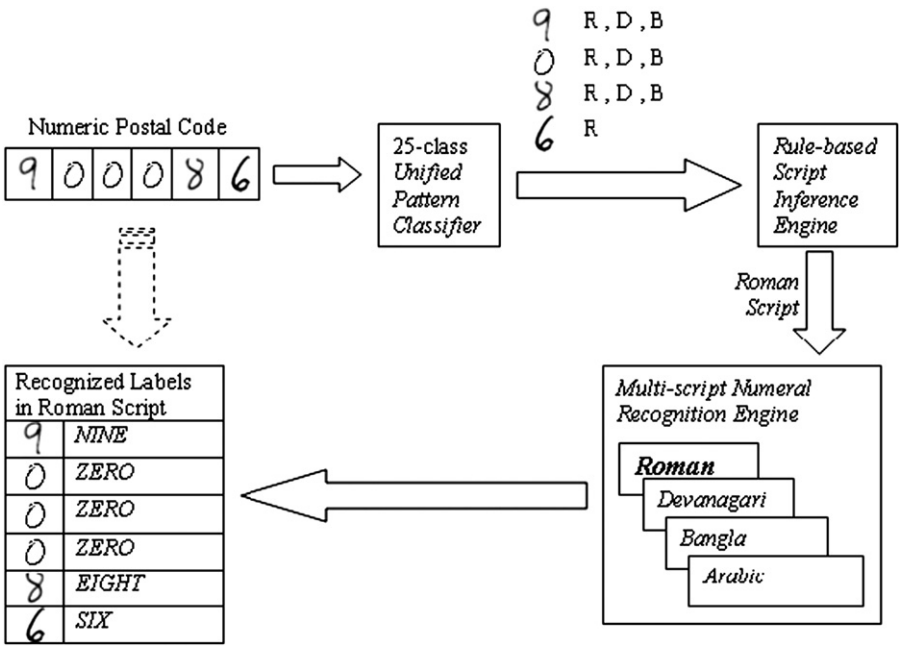

Fig. 5. A schematic block diagram of the overall system is shown.

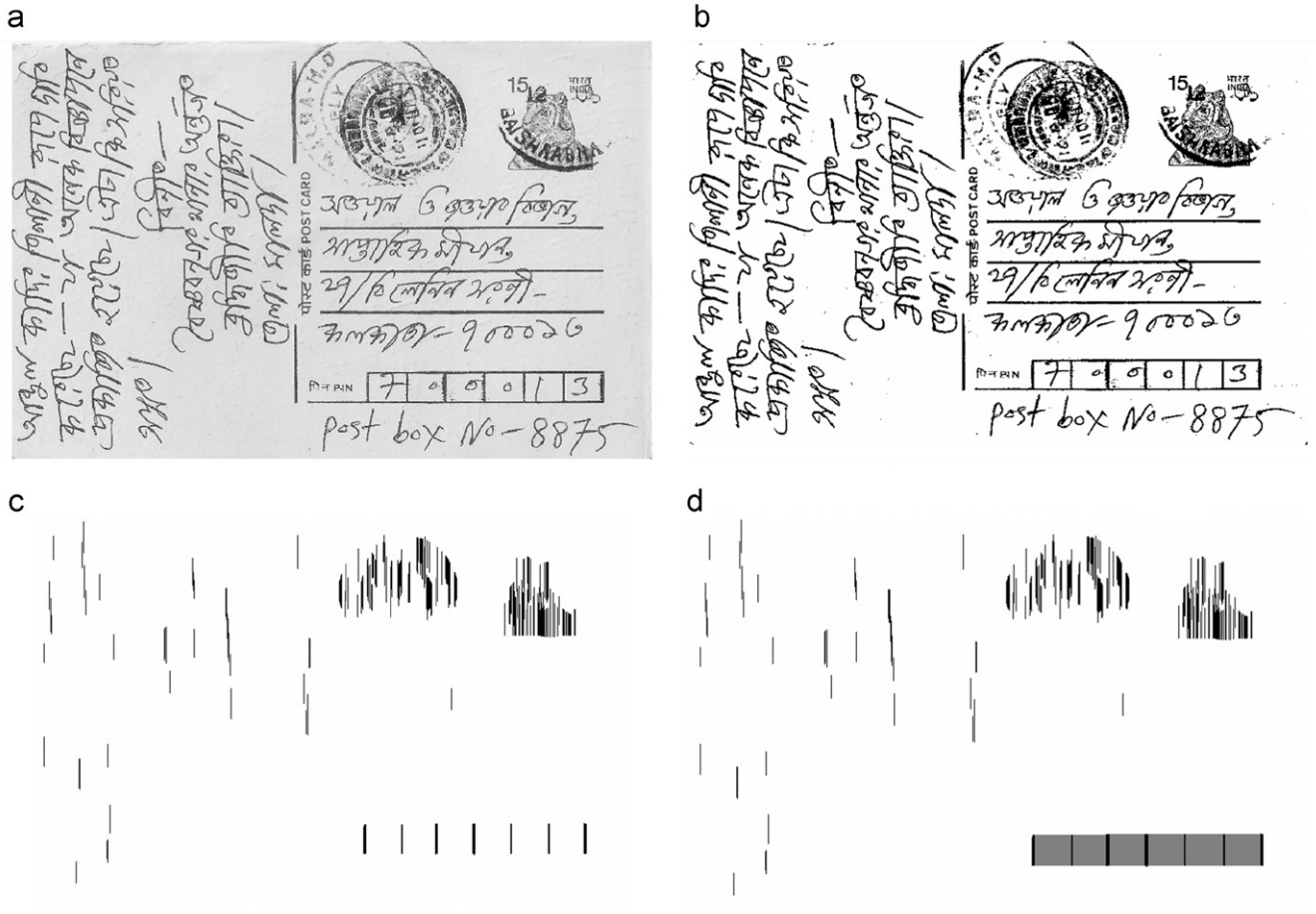

Fig. 6. Pre-processing steps for a postal document image. (a) A grey scale Indian postal document image, (b) the binarization output of the input image, (c) result after Hough transformation and (d) the localized postal-code region is marked with grey shades.

{1, (13, 14, 15), (L)}
{2, (17, 18), (D)}
{3, (19, 20, 21, 22, 23, 24), (U)}
{4, (10), (L, U)}
{5, (3), (D, B)}
{6, (11), (L, D)}
{7, (16), (D, U)}
{8, (0, 2, 4), (L, D, B)}

a

1	0	0	0	1	1
1	0	0	0	1	1
1	0	0	0	0	1
1	0	0	0	0	1
0	1	0	0	0	1
0	0	1	1	1	0

b

						Length of the Longest Bar	
1	0	0	0	2	2	*2*	
1	0	0	0	2	2	*2*	
1	0	0	0	0	1	*1*	
1	0	0	0	0	1	*1*	
0	1	0	0	0	1	*1*	
0	0	3	3	3	0	*3*	
			Sum =			*10*	

Fig. 7. An illustration for computation of the row wise longest–run feature. (a) The portion of a binary image enclosed within a rectangular region. (b) Every pixel position in each row of the image is marked with the length of the longest bar that fits consecutive black pixels along the same row.

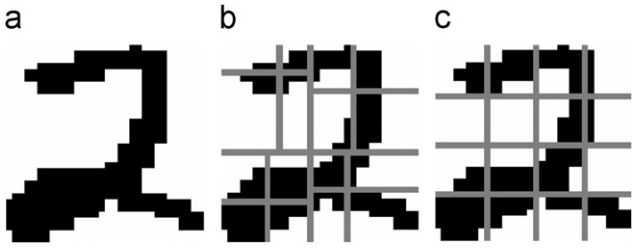

Fig. 8. Different image partitioning schemes for a sample digit image. (a) A sample digit image. (b) CG based partitioning of depth 2. (c) Equal partitioning.

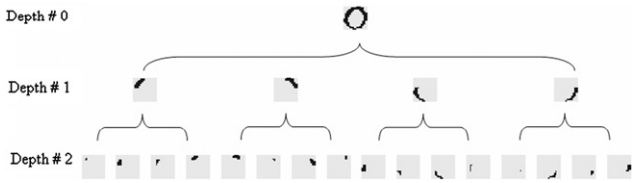

Fig. 9. Hierarchical partitioning of a sample sub-image using quad-tree of depth 2 is shown.

{9, (12), (L, D, U)}
{10, (7), (L, D, B, U)}

To make a final inference on the script of a numeric postal code, the developed rule based inference engine works in two phases. First, inference about the script is made for each numeral, and second, cumulative inference about the script is done on a string of numerals (as required in a numeric postal code). It is apparent from the above discussion that it is impossible to predict the script of a single numeral, unless the pattern belongs to any of the aforementioned four groups i.e., 0, 1, 2 and 3. The rule-set for the same is given below:

For any unknown digit pattern,
if Group_ID=0 then

Group_ID	The set of script(s) the pattern(s) represents						
0	(B)	৯	৪	৬	৮	৯	
1	(R)	5	6	7			
2	(D)	८	९				
3	(A)	٠	٢	٣	۵	۷	۸
4	(R, A)	۱					
5	(D, B)	৩					
6	(R, D)	३					
7	(D, A)	६					
8	(R, D, B)	০	২	৪			
9	(R, D, A)	۴					
10	(R, D, B, A)	۹					

Fig. 10. Compositions of the pattern groups designed for the rule-based inference engine.

The digit pattern belongs to Bangla script
else if Group_ID=1 then
The digit pattern belongs to Latin script
else if Group_ID=2 then
The digit pattern belongs to Devanagari script
else if Group_ID=3 then
The digit pattern belongs to Urdu script
else
The script of the numeral cannot be inferred from a single pattern

In case the script of the digit pattern cannot be determined directly, multiple digit patterns are required to infer on the identity of the script of the string of numerals. For example, it may be observed from above that either Group_ID#4 or 8 alone is incapable to decide on the script of a single pattern. In the case of numeric pattern of Group_ID#4, there may be ambiguity among the *Latin/Urdu* scripts and for the Group_ID#8, the ambiguity may be among the *Latin/Devanagari/Bangla* scripts. But if two numeric patterns belonging to these two groups appear simultaneously in a numeric string, the script inference engine decides that the script of the numeric string is *Latin*. The justification behind this may also be observed from the set intersection of the aforesaid groups, i.e. (L, U) ∩ (L, D, B) ⇒ (L).

With the above idea, the final rule set for the script inference engine is designed as follows:

1. *Initialize four LDBU script counters to zero.* i.e., *Lcnt=Dcnt=Bcnt=Ucnt=0.*
2. *for each digit pattern in the numeral string:*
   a. *identify its Group_ID and the set of scripts it represent. (e.g., Group_ID#8 with script set (L,D,B))*
   b. *for each of the members of the script set increment the corresponding script counters by one unit. (e.g., for Group_ID#8 increment Lcnt, Dcnt and Bcnt)*
3. *Identify the counter(s) with the maximum value.* i.e., *max { Lcnt, Dcnt, Bcnt, Ucnt }*
4. *if the number of script counter(s) with the maximum value is equal to 1*
   script of the numeral string is uniquely identified by the label of the counter. (e.g., 'Latin' for Lcnt)
   else
   The script of the numeral string is ambiguous

It may however be noted that due to inherent ambiguities of handwritten numerals the script inference engine may lead to

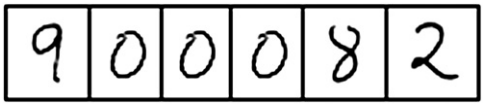

Fig. 11. An ambiguous string of numeric postal code is shown. Is it 900082 in *Latin* or 100042 in *Devanagari* or 700042 in *Bangla* ?

indecision on the script of the numerals in postal codes. This has been explained with examples in Fig. 11, where the inferences about the script for each individual digit pattern in a sequence are unable to converge on a specific script, even manually. In all other cases the script inference engine identifies a true script for any numeral string. This in turn invokes the numeral recognition engine for the corresponding script.

2.5. Design of the script-specific numeral recognizers

In our current work, four script specific digit classifiers are designed for each of the *LDBU* scripts. Each of such classifiers is trained with handwritten digit samples of corresponding script. Separate training and test datasets, prepared for this purpose, are discussed in the following section. Similar to unified pattern classifiers, 84 QTLR features are extracted from each of the pattern classes of any given script using a quad-tree structure. A SVM classifier with radial basis function (RBF) kernel is used for each script.

3. Experimental results

To evaluate the performance of the present technique, isolated handwritten numeral datasets of *LDBU* scripts are prepared at the CMATER laboratory of Jadavpur University, Kolkata, India. Five different datasets in all are prepared for this experiment. One of these datasets is formed for the unified pattern classifier consisting of 25 unique shaped numerals of the *LDBU* scripts and one each for the numeral recognition engines of the *Latin, Devanagari, Bangla* and *Urdu* scripts. Details of all these datasets are described below.

Devanagari and *Urdu* data sets were collected as handwritten numeral samples on a pre-defined data sheet at the CMATER from people of different age, sex, education groups. Regular black-ink gel pens with 0.5 mm tip were used for writing on those data sheets, which were optically scanned with a resolution of 300 dpi using a HP F380 flatbed scanner. These images were first pre-processed using two basic operations of skew correction [27] and morphological filtering [27] and then binarized using an adaptive global threshold of (min-Intensity+max-Intensity)/2; finally, the bounding rectangular box of each image was separately normalized to 32×32 pixels. A dataset of 3000 samples were collected under *Devanagari* script among which 2000 samples were used for training purpose and rest of the samples were used during test phase. For *Urdu* script, the size of training and test datasets were 2000 and 1000 samples, respectively. These datasets may be obtained for research purposes from the CMATER website (www.cmaterju.org/datasets.php) by requesting through an online data request form.

For handwritten *Bangla* digits, a dataset of 6000 samples was constructed by randomly selecting 600 samples for each of 10 digits from a larger database of 10,000 samples previously collected by both CVPR unit, Indian Statistical Institute, Kolkata and CMATER, Jadavpur University. A training set of 4000 samples and a test set of 2000 samples were then formed by considering equal number of samples for each digit. For *Latin* numerals a dataset of 4000 training samples was formed by randomly selecting from the training set of standard handwritten MNIST dataset [28] of size 60,000; similarly, a test dataset of 2000 samples were selected from the MNSIT test data set of size 10,000.

The dataset for these 25 unique shaped numerals is formed from 12,000 randomly selected handwritten samples of *LDBU* scripts, with 3000 samples taken from the dataset of each of the four scripts. If any unique pattern appears in multiple scripts, the same pattern is considered multiple times from multiple scripts with the same label, i.e., pattern ID, in the overall dataset. For example, a pattern similar to the shape '8' appears in the three scripts, viz., *Latin, Devanagari* and *Bangla*, with the labels 'Eight', 'Four', and 'Four', respectively. In the dataset under consideration, the said pattern is therefore considered thrice, i.e. once from each of the datasets for the respective scripts, but with same label (i.e., Pattern_ID#4 of Fig. 4). This is so because there may be minor variations of any unique shape across different scripts. Therefore, this dataset contains an unbalanced proportion of samples for each pattern.

3.1. Comparative analysis of different feature descriptors and classifiers

Detailed experimentations are conducted to evaluate the performances of different classical feature descriptors on the four digit datasets under consideration, to establish the efficiency of the novel QTLR feature set developed for this work. Different standard feature descriptors, popularly used by the researchers for classification of handwritten digits, are used for this comparison. More specifically, overlapping longest-run (OLR) features [29], shadow features [29], combinations of shadow-longest run (SLR) and shadow-longest run-octant centroid (SLOC) features [29], Gabor filter based features [30,31] and directional chain-code histogram (CCH) features [32,33] are compared with the QTLR features on the basis of digit recognition accuracies on the four datasets, used in our experiments.

Longest-run features, as described in Section 2.2.1, are computed over 9 overlapping sub-images and the overall image, to generate 40 OLR features. Lengths of projections of character images on three sides of each octant region are estimated to generate 24 shadow features. Coordinates of centroids of black pixels in all the 8 octants of a digit image are considered to generate 16 octant-centroid features. For computation of the Gabor features, the input 32×32 images are scaled down to 8×8 resolution and 4 directional Gabor coefficients are extracted for each of the 64 pixels of the down scaled image, resulting in 256 features. For computation of the CCH features, 4 directional chain-code frequencies (histograms) are considered as features in 16 (4×4) equally partitioned sub-images of any digit pattern, generating 64 features in all.

All the aforementioned feature descriptors are evaluated on four digit datasets using both multi layer perceptron (MLP) and SVM based pattern classifiers. To prepare the training and test datasets for these experiments, the respective datasets are randomly divided in the ratio of 2:1. Only one fold of experiment is conducted to estimate the relative strengths and weaknesses of the feature sets and the two classifiers under consideration. The MLP classifier, used in the work has only one hidden layer. Single hidden layer MLPs are chosen mainly to keep the computational requirement of the experiment low without affecting their function approximation capabilities [34]. Back propagation (BP) learning algorithm with learning rate $(\eta)=0.8$ and momentum term $(\alpha)=0.7$ is used here for training of the MLP based classifier for different numbers of neurons in its hidden layer.

Experiments are repeated for evaluating the recognition accuracy of the designed system with the SVM based pattern classifier. SVM is another popularly used supervised learning methodology used for classification tasks. A support vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional feature space, which can be used for a given classification problem. In the current work we have used RBF kernel (custom built from the open source LIBSVM project [35]) for designing the SVM classifiers. With SVM classifier, experiments are repeated for different values of kernel tuning parameters *alpha* (α) and *nu* (ν) for each feature descriptors. The network with best recognition accuracy is finally chosen for classification.

As observed from Tables 1 and 2 the QTLR feature descriptor clearly outperforms the other classical features with both MLP and SVM based classification schemes. This is mainly because of the prudent choice of quad-tree based image partitioning structure, resulting in significantly informative sub-images at the leaf-nodes in any given depth of the tree hierarchy. It is also observed from this experiment that the performances of SVM classifiers are superior in comparison to the MLP based classifiers for most the experiments under consideration. Therefore, we have chosen only SVM classifier for detailed performance evaluation at different stages of our experimental setup.

3.2. Performance of the pre-processing technique

In the current work we have employed a simple yet effective technique for localization of the postal-code region in digitized Indian postal document images. The technique is evaluated on 100 such images of post-cards, inland-letters and envelops. A sample subset of 50 such images is available at the CMATER database repository (http://code.google.com/p/cmaterdb/downloads/list) as the CMATER database series 5, version 1 (CMATERdb 5.1). Excepting two highly degraded images, the current technique localizes the postal-code regions successfully in all cases. Fig. 12(a–f) shows some grey scale postal document images with the corresponding Hough transformation output in black colour and the localized postal code regions in the grey shaded smeared regions. The isolated numerals are extracted from each such smeared regions using connected component labeling algorithm [27], enclosed in a rectangular bounding region and finally normalized to a dimension of 32×32 pixels for subsequent processing. The grey scale, binarized and Hough-transformed images of a postal document are shown in Fig. 13(a–c) where the current Hough transformation based localization technique fails. The reason behind the failure is mainly due to the binarization error, occurred due to the high degree of degradation in such document images.

3.3. Performance of the unified pattern classifier

For developing the training and test sets for the SVM based unified pattern classifier, employed for this work, the relevant dataset is divided in a ratio of 9:1. As discussed earlier, both the training and the test datasets contain unbalanced proportion of samples of the 25 pattern classes.

For classification of the unique digit patterns of the *LDBU* scripts into 25 pattern classes, the 84 element feature set, as discussed earlier, is used. For cross validation of results, ten different folds of test sets are formed by dividing the original dataset of 12,000 samples into ten equal mutually disjoint parts. For each fold of the test set, the corresponding training set is formed with the rest of the database. Thus ten pairs of the test and the training sets are formed for tenfold cross validation of results. In each of these pairs, the training and the test sets are of sizes 10,800 samples and 1200 samples, respectively.

Experiments are conducted for evaluating the recognition accuracy of the designed system with the SVM based pattern classifier with the RBF kernel function. With SVM classifier, experiments are again repeated with different values of kernel tuning parameters α and ν and the network with best recognition accuracy is finally chosen for classification. In ten folds of the current experiment, maximum, minimum and average recognition accuracies of 93.5%, 90.08% and 92.03% are achieved. The standard deviation of the recognition accuracies over the ten folds of cross validation is observed as 1.13%.

3.4. Performance of the rule-based script inference engine

The decision on the label of the unique digit pattern, as obtained from the SVM based unified pattern classifier, is fed to the script inference engine, which subsequently re-groups the patterns into 11 categories, as shown in Fig. 10.

Table 1
The success rate of different script using MLP classifier for different feature set.

Feature descriptor	Feature dimension	*Latin* digit dataset	*Devanagari* digit dataset	*Bangla* digit dataset	*Urdu* digit dataset
OLR	40	91.50	92.71	93.85	94.10
Shadow	24	89.05	90.75	92.30	90.60
SLR	64	94.70	94.30	96.00	94.20
SLOC	80	95.05	94.95	95.95	94.80
Gabor	256	94.55	88.50	93.30	94.90
CCH	64	94.00	91.30	92.85	90.60
QTLR	84	93.70	96.45	96.70	95.60

Table 2
The success rate of different script using SVM classifier for different feature set.

Feature descriptor	Feature dimension	*Latin* digit dataset	*Devanagari* digit dataset	*Bangla* digit dataset	*Urdu* digit dataset
OLR	40	92.00	95.04	93.20	93.80
Shadow	24	91.50	94.30	94.15	91.09
SLR	64	95.40	94.60	95.25	93.70
SLOC	80	95.60	95.56	96.00	94.80
Gabor	256	96.10	92.81	95.05	95.10
CCH	64	94.25	93.60	93.75	91.30
QTLR	84	95.10	97.85	96.10	94.60

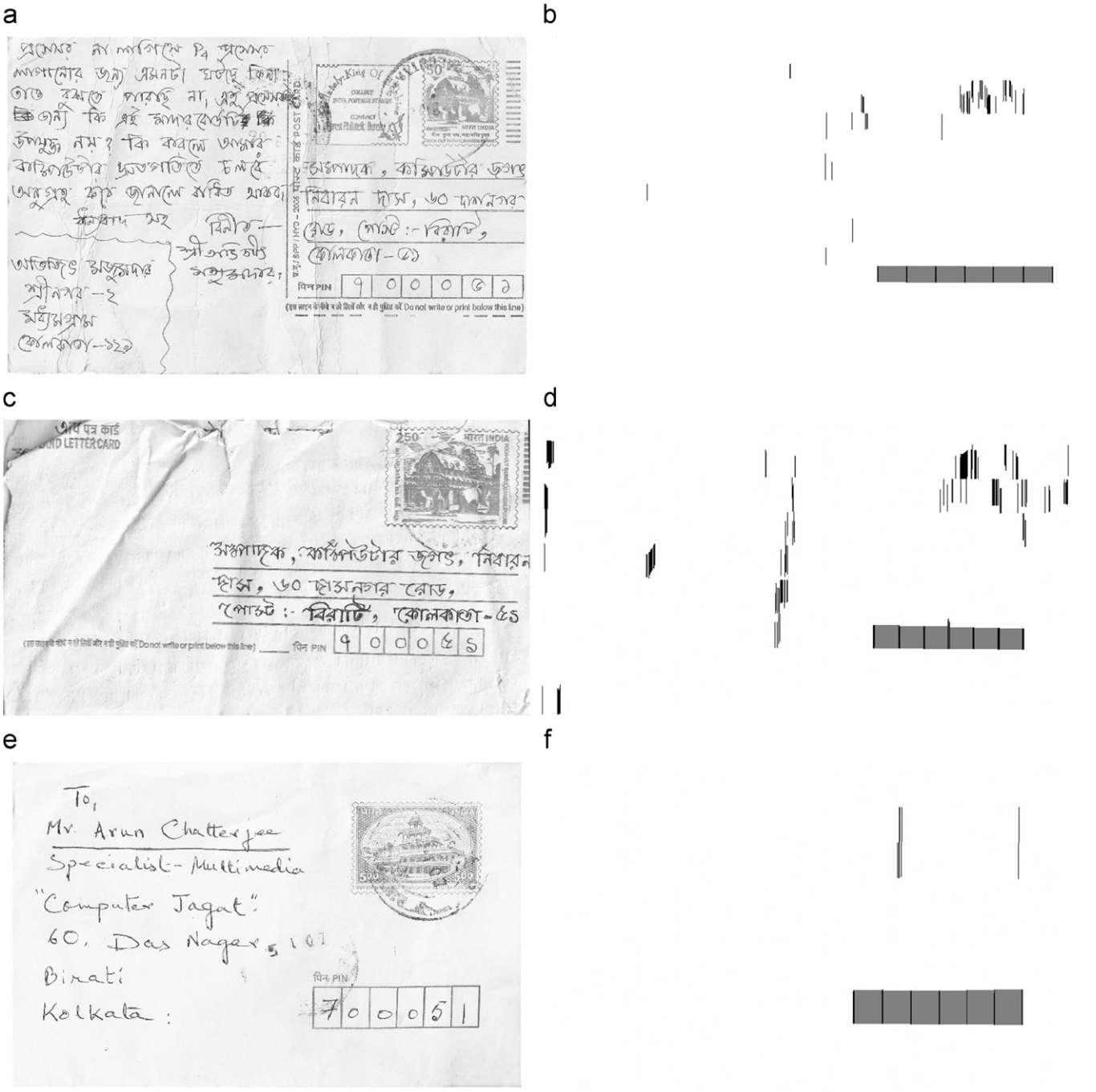

Fig. 12. (a–f) Sample grey scale images of Indian postal documents are shown in (a), (c), (e) and the corresponding postal-code localization results in (b), (d) and (f).

To evaluate the performance of the script recognition engine on a string of numerals of any of the *LDBU* scripts, random digit images of any given script are populated randomly from the database of 12,000 images generated for the 25-class unified pattern classifier. Six similar script digit patterns are considered together to simulate the extracted numerals, that might appear in any Indian postal code. All such six digit strings ideally belong to any of the four *LDBU* scripts. We then employ the rule-based script inference algorithm, as described in sub-Section 2.4, to infer the script of the complete postal-code string. We populate 10,000 such random postal-code strings for in one experimental fold, for any given script. We generate 10 such folds of experiments for each of the four LDBU scripts and estimate the maximum, minimum, average and standard deviation of respective script detection accuracies. The average script-inference accuracy over a six digit numeric string (standard length of postal codes in India) is observed as 96.72%. The detailed analysis of average, maximum, minimum accuracies over ten random runs of the experiment is shown in Table 3 and graphically illustrated in Fig. 14.

3.5. Performance of the script-specific numeral recognizers

For developing a training set and a test sets for the SVM based numeral recognizers for each of the *LDBU* scripts, the relevant

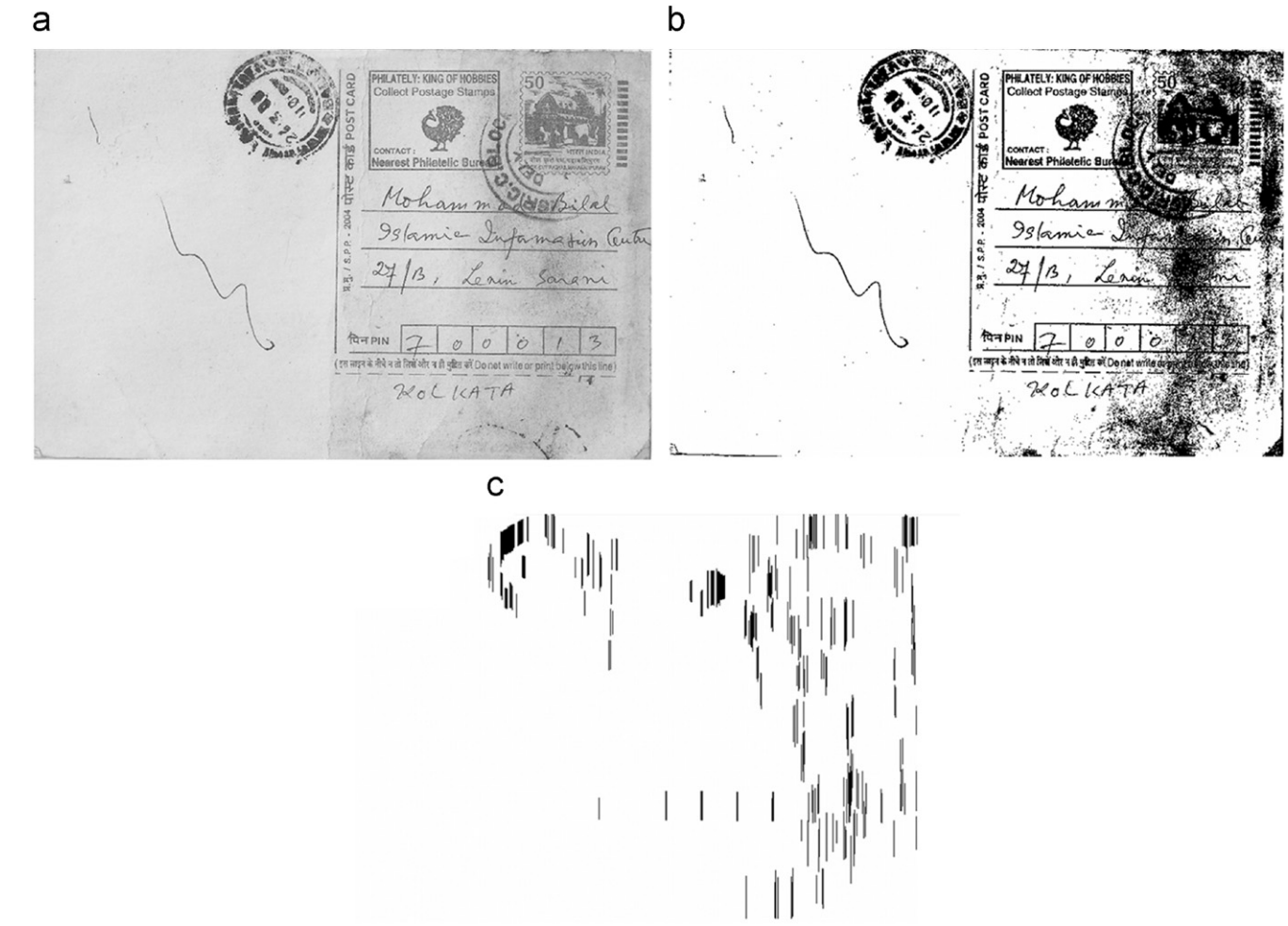

Fig. 13. (a–c) A sample Indian postal document image where the postal-code localization technique fails; (a) a sample grey scale image, (b) the corresponding binarized image with many noise pixels near the address block and (c) the Hough transformation image with many vertical lines near the postal-code region.

Table 3
Variations in the script inference accuracies for the *LDBU* scripts and their averages over different numeric string lengths.

Experimental folds	*Latin*	*Devanagari*	*Bangla*	*Urdu*
Fold-1	95.48	95.73	96.70	98.45
Fold-2	95.65	96.13	96.64	98.59
Fold-3	95.42	96.02	96.98	98.55
Fold-4	95.69	95.50	96.58	98.90
Fold-5	95.60	96.22	97.03	98.61
Fold-6	95.66	96.17	96.81	98.48
Fold-7	95.36	95.88	96.77	98.56
Fold-8	95..50	95.79	97.00	98.56
Fold-9	95.60	96.25	96.79	98.46
Fold-10	95.68	95.51	96.78	98.54
Maximum over 10 folds	95.69	96.25	97.03	98.90
Minimum over 10 folds	95.36	95.50	96.58	98.45
Average (mean)	95.56	95.92	96.81	98.57
Standard deviation	0.1165	0.2818	0.1528	0.1278

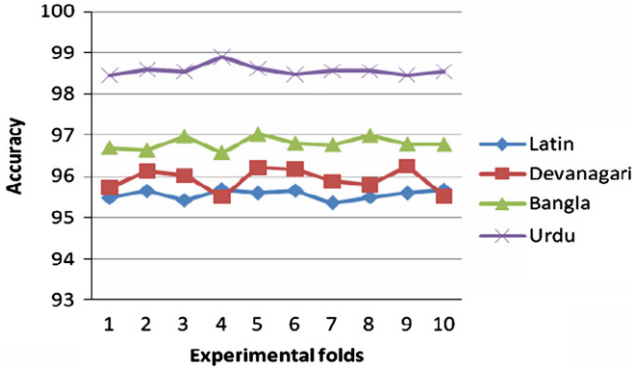

Fig. 14. Variations in the script inference accuracies and their average for the *LDBU* scripts over different numeric string lengths are shown.

datasets are divided in a ratio of 9:1. For classification of the 10 digit patterns for each of the *LDBU* scripts, the 84 element feature set, as discussed earlier, is again used.

Exhaustive variations of recognition performances of the SVM classifiers for the four different scripts are recorded on tenfolds of training and test data with exhaustive variations in SVM kernel parameters. As observed from these experiments, the average recognition rates over the ten folds of the classifiers for *Latin, Devanagari, Bangla* and *Urdu* scripts are 95.55%, 95.63%, 97.15% and 96.20%, respectively. The maximum, minimum accuracies in any fold of experiment, along with the standard deviation with respect to mean, are given in Table 4 for each of the four scripts under consideration. Fig. 15 shows the variations in recognition accuracies of the SVM classifier over tenfolds of cross validation of results for the four digit datasets and the 25-class unified pattern classifier. Fig. 16(a–d) shows sample digit images of the four

Table 4
Analysis of classification accuracy (in percentage) of the SVM based classifiers in tenfold test datasets of *LDBU scripts* is shown.

CV10 accuracy (in percentage)	*Latin* digit dataset	*Devanagari* digit dataset	*Bangla* digit dataset	*Urdu* digit dataset
Fold-1	95.33	97.00	97.00	97.33
Fold-2	94.50	96.33	97.17	96.67
Fold-3	95.83	93.00	96.67	94.33
Fold-4	96.67	89.67	97.17	98.33
Fold-5	95.83	92.00	97.33	96.67
Fold-6	96.50	96.33	97.83	97.67
Fold-7	96.33	96.67	96.83	93.67
Fold-8	94.83	98.33	97.17	93.67
Fold-9	94.67	98.33	97.67	95.67
Fold-10	95.00	98.67	96.67	98.00
Maximum over 10 folds	96.67	98.67	97.83	98.33
Minimum over 10 folds	94.5	89.67	96.67	93.67
Average (mean)	95.55	95.63333	97.15	96.2
Standard deviation	0.7937	3.0447	0.3885	1.7722

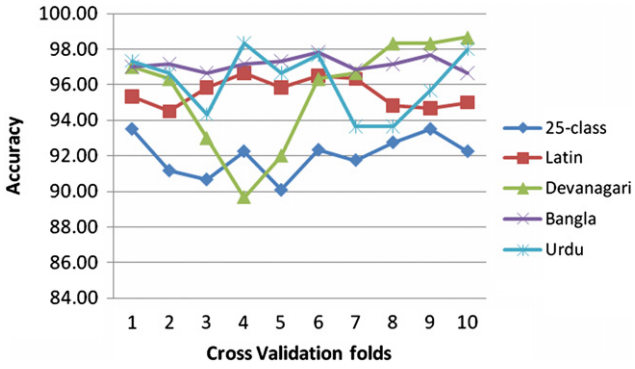

Fig. 15. Curves show variations in recognition performances of the SVM classifiers over ten folds cross validation of results for the 25-class unified pattern classifier and the individual classifiers for the *Latin*, *Devenagari*, *Bangla* and *Urdu* scripts.

scripts where the respective recognizers fail to classify the patterns in the true class.

4. Conclusion

A novel multi-stage framework is introduced here for automatic sorting of multi-script postal documents. The Indian multilingual scenario is particularly highlighted throughout the experiment. Unlike existing works on script identification from handwritten/printed document pages or text lines/words, the present work emphasizes on extraction of script information from the numeric patterns of different scripts as might be required in many applications. Related research contributions on postal automation schemes, reported so far, assume similar script information for the textual address part and the numeric postal codes of the address block. This contradicts the real-life scenario, as illustrated in Figs. 1 and 2. Since the postal sorting primarily depends on the postal codes alone, a complete framework was required to interpret multi-script postal addresses from the numeric postal codes only.

Under the current work, we have developed an effective Hough transformation based technique for localization of postal-code regions form potentially cluttered and noisy background. On a limited set of postal documents, we achieved around 98% postal-code localization accuracy. A subset of this dataset is made available at the public domain for researchers interested in this field of study. We however look forward to build a larger repository of such postal documents, as an extension of this work in future. The four scripts that are used in this experiment are not

a

Latin Script	*Digit image*			
	True label	2	7	9
	Misclassified label	8	1	4

b

Devanagari Script	*Digit image*			
	True label	6	4	8
	Misclassified label	5	2	7

c

Bangla Script	*Digit image*			
	True label	7	9	1
	Misclassified label	9	4	6

d

Urdu Script	*Digit image*			
	True label	9	7	7
	Misclassified label	8	6	4

Fig. 16. (a–d) Some of the misclassified digit images of *LDBU* scripts with the respective true and misclassified labels.

only popular in India, but also have a wide usage in its neighbouring countries like Bangladesh, Pakistan and Nepal. The developed multi-lingual framework is therefore relevant to most parts of the Indian sub-continent. We however excluded some more popular Indian scripts like *Oriya*, *Telugu*, *Kannada*, *Tamil* and *Malayalam* due to non-availability of postal documents and handwritten training samples. In our future research we however propose to incorporate rest of the Indian scripts to develop an improved postal automation software. In the current work we have compared the performances of different feature sets with both MLP and SVM classifiers. We finally chose the QTLR feature with SVM classifier for giving better recognition accuracy on test datasets under consideration. The design of the quad-tree based longest-run features is another novelty of the current work, reported in this paper. A detailed experimentation, as discussed in Section 3.1, validates the choice and superiority of this new QTLR feature set.

However, one major limitation of our technique is the *commitment* to a *specific* feature descriptor and a *specific* classifier.

In our future research, we propose to design a feature combination scheme and a classifier ensemble for improvement of classification decisions. Another limitation of the designed system is its bottleneck in resolving inherent ambiguities in script identification in a string of handwritten numerals. In a random numeric string, if the rule-based inference engine fails to converge on a specific script, the numeric string remains ambiguous even through manual intervention, as illustrated in Fig. 9. In such cases, scripts of the numeric string may be inferred from the script of the textual address parts, as far as practicable.

The designed framework is novel in the sense that it addresses the need of a practical mail sorting system in a multi-script environment based on the analysis of numeric postal codes alone. Apart from postal automation this framework is relevant in reading amounts from bank-cheques, interpretation of hand-filled digitized form documents, etc. in any multi-lingual environment. In view of above discussion, it may finally be concluded that the current framework opens up a new direction of OCR research for dealing with the complex problems related to automatic sorting of multi-script postal documents and similar applications.

Acknowledgements

Authors are thankful to the "Center for Microprocessor Application for Training Education and Research", "Project on Storage Retrieval and Understanding of Video for Multimedia" both of Computer Science and Engineering Department, Jadavpur University, for providing infrastructural facilities during progress of the work. Authors are also thankful to the CVPR Unit, ISI Kolkata, for providing the necessary dataset of handwritten *Bangla* script. One of the authors, Prof. Dipak Kumar Basu is thankful to the A.I.C.T.E. (New Delhi, India) for awarding him an Emeritus Fellowship (F. No: 1-51/RID/EF(13)/2007-08).

References

[1] ⟨http://www.rajbhasha.nic.in/dolacteng.htm⟩.
[2] ⟨http://en.wikipedia.org/wiki/Languages_with_official_status_in_India⟩.
[3] ⟨http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers⟩.
[4] S. Chaudhury R. Sheth, Trainable script identification strategies for Indian languages, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition, 1999, pp. 657–660.
[5] V. Singhal, N. Navin, D. Ghosh, Script-based classification of hand-written text document in a multilingual environment, Research Issues in Data Engineering (2003) 47–54.
[6] Gopal Datt Joshi, Saurabh Garg, Jayanthi Sivaswamy, Script identification from Indian documents, DAS 2006, LNCS, vol. 3872, 2006, pp. 255–267.
[7] Gopal Datt Joshi, Saurabh Garg, Jayanthi Sivaswamy, A generalised framework for script identification, IJDAR Vol. 10 (2007) 55–68.
[8] U. Pal, B.B. Chaudhuri, Automatic separation of different script lines from Indian multi-script documents, in: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, 1998, pp. 141–146.
[9] U. Pal, B.B. Chaudhuri, Script line separation from Indian multi-script documents, in: Proceedings of Fifth International Conference on Document Analysis and Recognition, 1999, pp. 406–409.
[10] U. Pal, B.B. Chaudhuri, Automatic identification of English, Chinese, Arabic, Devnagari and Bangla script line, in: Proceedings of the International Conference on Document Analysis and Recognition, 2001, pp. 0790–0794.
[11] U. Pal, B.B. Chaudhuri, Identification of different script lines from multi-script documents, Image and Vision computing 20 (13-14) (2002) 945–954.
[12] U. Pal, S. Sinha, B.B. Chaudhuri, Multi-script line identification from Indian documents, in: Proceedings of the Seventh International Conference on Document Analysis and Recognition, vol. 2, 2003, pp. 880–884.
[13] Sinha, Suranjit,Umapada Pal, B.B. Chaudhuri, Word–wise script identification from Indian documents, DAS 2004, LNCS, vol. 3163, 2004, pp. 310–321.
[14] K. Roy, A. Banerjee, U. Pal, A system for wordwise handwritten script identification for Indian postal automation, in: Proceedings of the IEEE INDICON-04, 2004, pp. 266–271.
[15] Lijun, ZhouYue Lu Chew Lim Tan, Bangla/English script identification based on analysis of connected component profiles, DAS 2006, LNCS, vol.3872, 2006, pp. 243–254.
[16] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, A system towards Indian postal automation, in: Proceedings of the Ninth IWFHR, 2004, pp. 361–367.
[17] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, A. Belaid, A system for Indian postal automation, in: Proceedings of the Eighth ICDAR, 2005.
[18] R. Plamondon, S.N. Srihari, On-line and off-line handwritten recognition: a comprehensive survey, IEEE Transactions on PAMI 22 (2000) 62–84.
[19] Y. Wen, Y. Lu, P. Shi, Handwritten Bangla numeral recognition system and its application to postal automation, Pattern Recognition 40 (2007) 99–107.
[20] U. Bhattacharya et al., Neural combination of ANN and HMM for handwritten Devnagari numeral recognition, in: Proceedings of the 10th IWFHR, 2006, pp. 613–618.
[21] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, D.K. Basu, A two pass approach to pattern classification, in: N.R. Pal, et al. (Eds.), Lecture Notes in Computer Science, vol. 3316, ICONIP, Kolkata, November 2004, pp. 781–786.
[22] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, D.K. Basu, Handwritten Bangla digit recognition using classifier combination through DS technique, in: S.K. Pal et al. (Eds.), Lecture Notes in Computer Science, vol. 3776, PReMI, ISI, Kolkata, December 2005, pp. 236–241.
[23] Umapada Pal, N. Sharma, Tetsushi Wakabayashi, Fumitaka Kimura, Handwritten numeral recognition of six popular Indian scripts, in: ICDAR, 2007, pp. 749–753.
[24] Sabri Mahmoud, Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models, Signal Processing 88 (no. 4) (April, 2008) 844–857.
[25] S. Basu, S.S. Seth, P. Sarkar, B. Das, S. Dey, S. Ghosh, Recognition of Pincodes from Indian Postal Documents, Soft Computing, Allied Publishers, 817764632-X, 9788177646320, pp. 239–245.
[26] Ying Wen, Yue Lu, Pengfei Shi, Handwritten Bangla numeral recognition system and its application to postal automation Pattern Recognition 40 (Issue 1) (January 2007) 99–107.
[27] R.C. Gonzalez, R.E. Woods, in: Digital Image Processing, first ed., Prentice-Hall, India, 1992.
[28] http://yann.lecun.com/exdb/mnist/.
[29] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, A hierarchical approach to recognition of handwritten *Bangla* characters, Pattern Recognition vol. 42 (no. 7) (2009) 1467–1484.
[30] X. Wang, X. Ding, C. Liu, Gabor filters-based feature extraction for character recognition, Pattern Recognition 38 (Issue 3) (March 2005) 369–379.
[31] C.L. Liu, M. Koga, H. Fujisawa, Gabor feature extraction for character recognition: comparison with gradient feature, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR) 2005, pp. 121–125.
[32] U. Bhattacharya, S.K. Parui, M. Sridhar, F. Kimura, Two-stage recognition of handwritten *Bangla* alphanumeric characters using neural classifiers, in: Proceedings of the Second Indian International Conference on Artificial Intelligence (IICAI), 2005, pp. 1357–1376.
[33] U. Bhattacharya, M. Sridhar, S.K. Parui, On recognition of handwritten *Bangla* characters, in: Proceedings of the ICVGIP-06, Lecture Notes in Computer Science, vol. 4338, 2006, pp. 817–828.
[34] N.J. Nilson, Principles of Artificial Intelligence, Springer-Verleg, pp. 21–22.
[35] ⟨http://www.csie.ntu.edu.tw/∼cjlin/libsvm/⟩.

Subhadip Basu received his B.E. degree in Computer Science and Engineering from Kuvempu University, Karnataka, India, in 1999. He received his Ph.D. (Eng.) degree thereafter from Jadavpur University (J.U.) in 2006. He joined J.U. as a senior lecturer in 2006. He is the recipient of BOYSCAST fellowship from the Department of Science and Technology, Government of India, EMMA fellowship from European Union and the HIVIP fellowship from Hitachi Limited, Japan. He is a member of IEEE, U.S.A., and IUPRAI, India. His areas of current research interest are OCR of handwritten text, gesture recognition, real-time image processing.

Nibaran Das received his B.Tech degree in Computer Science and Technology from Kalyani Govt. Engineering College under Kalyani University, in 2003. He received his M.C.S.E degree from Jadavpur University, in 2005. He joined J.U. as a lecturer in 2006. His areas of current research interest are OCR of handwritten text, Bengali fonts, biometrics and image processing. He is a member of IEEE, U.S.A. He has been an editor of Bengali monthly magazine "Computer Jagat" since 2005.

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 degree from Jadavpur University, in 2005. He joined J.U. as a lecturer in 2008. His areas of current research interest are document image processing, line extraction and segmentation of handwritten text images.

Mahantapas Kundu received his B.E.E, M.E.Tel.E and Ph.D. (Eng.) degrees from Jadavpur University, in 1983, 1985 and 1995, respectively. Prof. Kundu has been a faculty member of J.U since 1988. His areas of current research interest include pattern recognition, image processing, multimedia database, and artificial intelligence.

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 J.U since 1987. Her current research interest includes image processing, pattern recognition, and multimedia systems. She is a senior member of the IEEE, U.S.A., Fellow of I.E (India) and W.B.A.S.T, Kolkata, India.

Dipak Kumar Basu received his B.E.Tel.E., M.E.Tel., and Ph.D. (Engg.) degrees from Jadavpur University, in 1964, 1966 and 1969, respectively. Prof. Basu has been a faculty member of J.U from 1968 to January 2008. He is presently an A.I.C.T.E. Emiretus Fellow at the CSE Department of J.U. His current fields of research interest include pattern recognition, image processing, and multimedia systems. He is a senior member of the IEEE, U.S.A., Fellow of I.E. (India) and W.B.A.S.T., Kolkata, India and a former Fellow, Alexander von Humboldt Foundation, Germany.