Recognition of Numeric Postal Codes from Multi-script Postal Address Blocks

Subhadip Basu[1,*], Nibaran Das[1], Ram Sarkar[1], Mahantapas Kundu[1], Mita Nasipuri[1], and Dipak Kumar Basu[1,2]

[1] Computer Science & Engineering Department, Jadavpur University, Kolkata-700032, India
[2] A.I.C.T.E. Emeritus Fellow
{subhadip,nibaran,ramsarkar,mkundu,mnasipuri, dkbasu}@cse.jdvu.ac.in

Abstract. The objective of the current work is to recognize postal codes written in *Roman*, *Devanagari*, *Bangla* and *Arabic* scripts. In the first stage 25 unique digit patterns are identified from the handwritten numeral patterns of the said four scripts. A script independent unified pattern classifier is then designed to classify any digit pattern of these scripts into one of the 25 classes. In the next stage a rule-based script inference engine infers about the script of the numeric string, that invokes one of the four script specific classifiers. The average script-inference accuracy over a six digit numeric string is observed as 95.1% and the best recognition rates for the four script specific digit classifiers are obtained as 96.10%, 94.40%, 96.45 % and 95.60% respectively.

Keywords: OCR, script-identification, classification, postal automation.

1 Introduction

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, the postal code is often written in different regional scripts along with the *Roman* script. In the present work, we have attempted to address the problem related to the interpretation of handwritten pin codes of aforementioned four scripts, viz., *Roman*, *Devanagari*, *Bangla* and *Arabic (RDBA)*. We first identify the specific script in which the numeric postal code is written and then focus on recognition of that postal code.

Among the related works in this domain, Sinha *et al.* [1] and Roy *et al.* [2] developed similar techniques for word-wise identification of *Roman, Devanagari* and *Bangla* scripts in handwritten textual postal addresses using topological and structural features. However, they have not shown any result on identification of the scripts for the numeric postal codes. In another work, Zhou *et al.* [3] developed a connected

* Corresponding author.

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 381–386, 2009.

component profile analysis technique for separation of *Roman* and *Bangla* script based postal documents. Other works, reported in the literature, related to postal automations [4, 5] do 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 [2-5], 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.

Research contributions on recognition of handwritten numerals [4-10] mostly focus on feature based recognition of isolated handwritten digit samples of a given script using standard classifiers. In one of our earlier works [6], a two-pass feature based approach was designed for recognition of handwritten numerals of Bangla script. In another work [7], 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.* [8], used contour based directional features to recognize handwritten numerals of six popular Indian scripts, viz., *Devanagari, Bangla, Telegu, Oriya, Kannada* and *Tamil*. They used six different quadratic classifiers, each for the six different scripts, and obtained good recognition accuracy. In another recent work Wen *et al.* [9], developed a handwritten *Bangla* numeral recognition system for automatic sorting of mails for the Bangladesh Post. Using the principles of *Principal Component Analysis* and *Support Vector Machine*, they achieved high reliability in recognition of handwritten numerals of *Bangla* script, but remained silent over the script identification technique for the said pattern classes.

However, in any postal document the script of the numeric postal codes may vary from the script of the textual address part. More specifically, people often write postal address in two scripts, *i.e.*, the textual parts in regional scripts like *Devanagari*, *Bangla* or *Arabic* and the numeric part including the postal code in the *Roman* script. Therefore, inferring the script of the postal code on the address block may often mislead the script recognition process. This has been one of our key motivations behind the current work, discussed in this paper.

2 The Present Work

It is evident from the earlier discussions that limited research contributions have been reported so far on interpretation of multi-script postal documents. To address these issues we have developed a multi-stage approach for recognition of multi-script postal codes written in *Roman*, *Devanagari*, *Bangla* and *Arabic* scripts (as shown in Fig. 1). In the first stage, 25 unique digit patterns, as shown in Fig. 2, are identified from the handwritten numeral patterns of the said four scripts. A script independent unified pattern classifier is then designed to classify any digit pattern of the *RDBA* scripts into one of the 25 classes. In the next stage, a rule-based script inference engine is designed to infer about the script of the numeric string, based on the recognition decisions obtained in the first stage. This decision on the script of the postal code invokes one of the four script specific classifiers from the multi-script numeral recognition engine. Finally, the chosen pattern classifier is used to recognize normalized, binary digit patterns of the corresponding script.

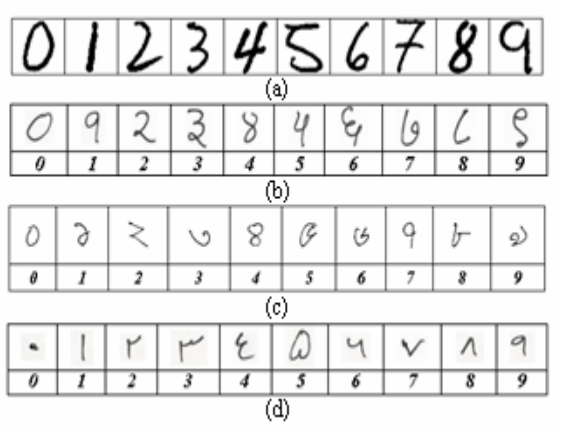

Fig. 1. (a). The decimal digit sets of *Roman* script. ***(b-d),***The respective digit sets of *Devanagri, Bangla and Arabic* scripts with corresponding labels in Roman script.

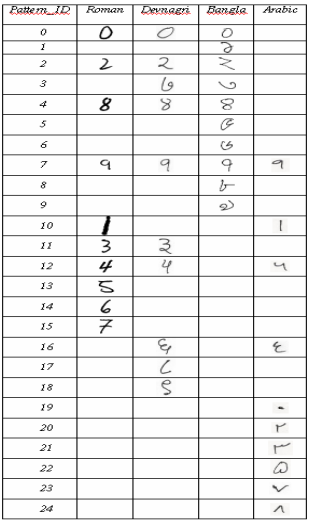

Fig. 2. 25 unique digit patterns are identified from the four RDBA scripts

2.1 Design of the Feature Descriptor

For extraction of the features for both the unified pattern classifier and the multi-script numeral recognition engine, quad-tree based longest-run features are used in the current work. 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 [11] 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.

In the current work, we have used a novel modified version of quad tree structure to 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 *Centre of Gravity* (*CG*) of black pixels in that region. In the current work, we have considered the depth of the quad-tree structure as 2. This generates 4^2, i.e., 16 sub-images at the leaf node positions, thereby resulting in 64 (16 x 4) longest-run features for any digit pattern.

2.2 Design of a Multi-script Pattern Classification Framework

As already mentioned, handwritten digit patterns of the *RDBA* scripts (as shown in Fig. 1) often bear significant similarities in shapes among themselves and we can identify 25 unique pattern shapes, as shown in Fig. 2, from the 40 digit patterns of four different scripts. A multi-layer perceptron based classifier is designed, with the aforementioned features, as a unified pattern classifier for recognizing these 25 different pattern classes.

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, {*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. 3.

Group_ID	The set of script(s) the pattern(s) represents						
0	(B)						
1	(R)						
2	(D)						
3	(A)						
4	(R, A)						
5	(D, B)						
6	(R, D)						
7	(D, A)						
8	(R, D, B)						
9	(R, D, A)						
10	(R, D, B, A)						

Fig. 3. Compositions of the pattern groups designed for the rule-based inference engine

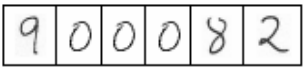

Fig. 4. An ambiguous string of numeric postal code is shown. Is it 900082 in *Roman* or 100042 in *Devanagari* or 700042 in *Bangla* ?

To make a final inference on the script of a numeric postal code, the developed rule based inference engine works in two phases. Firstly, inference about the script is made for each numeral, and secondly, 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.

In case the script of the digit pattern cannot be determined directly, multiple digit patterns are required to infer on 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 *Roman/Arabic* scripts and for the Group_ID#8, the ambiguity may be among the *Roman/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 *Roman*. The justification behind this may also be observed from the set intersection of the aforesaid groups, i.e. (R, A) $\cap$ (R, D, B) $\Rightarrow$ (R).

It may however be noted that due to inherent ambiguities of handwritten numerals the script inference engine may lead to indecision on the script of the numerals in postal codes (illustrated in Fig. 4). In all other cases the script inference engine identifies the true script for any numeral string. This in turn invokes the numeral recognition engine for the corresponding script. Details of the training and test datasets, prepared for each such classifier, are discussed in the following section. Similar to unified pattern classifiers, 64 longest-run features are extracted from each of the pattern classes of any given script using a quad-tree structure. A multi-layer perceptron with back-propagation learning algorithm is again used for each script.

3 Experimental Results

To evaluate the performance of the present technique, isolated handwritten numeral datasets of RDBA scripts are prepared at the CMATER laboratory of Jadavpur University, Kolkata, India. One of these datasets is formed for the unified pattern classifier consisting of 25 unique shaped numerals of the *RDBA* scripts and one each for the numeral recognition engines of the *Roman, Devanagri, Bangla* and *Arabic* scripts. Details of the script specific digit datasets are given in www.cmaterju.org.

The dataset for these 25 unique shaped numerals is formed from 6000 randomly selected handwritten samples of *RDBA* scripts, with 1500 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 in the overall dataset. 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.

For the present work an MLP [12] with one hidden layer is chosen. *Back Propagation* (BP) learning algorithm with learning rate (η) = 0.8 and momentum term (α) = 0.7 is used here for training of the MLP based classifier for different numbers of neurons in its hidden layer. As observed from Table 1, the best recognition rate achieved for the Unified Pattern Classifier with different numbers of neurons in the hidden layer is 88.8%. The decision on the label of the unique digit pattern, as obtained from the unified pattern classifier, is fed to the script inference engine, which subsequently re-groups the patterns into 11 categories.

To evaluate the performance of the script recognition engine on a string of numerals of any of the *RDBA* scripts, random strings of variable lengths are populated from the aforementioned numeral datasets. The average script-inference accuracy over a six digit numeric string is observed as 95.1%. Similar to the unified pattern classifier, for classification of the 10 digit patterns for each of the *RDBA* scripts, the 64 element feature set is again used. As observed from these experiments, the best recognition rates of the four classifiers for *Roman, Devanagari, Bangla* and *Arabic* scripts are obtained as 96.10%, 96.40%, 96.45 % and 95.60% respectively.

Table 1. Recognition performances of different MLP classifiers on the respective test samples with different numbers of neurons in the hidden layer of each

No of Hidden neurons	Unified pattern classifier	*Roman* digit classifier	*Devanagari* digit classifier	*Bangla* digit classifier	*Arabic* digit classifier
40	86.4	95.7	95.8	95.85	95.20
45	87.35	95.85	95.4	96.4	94.90
50	87.45	95.7	96.1	96.35	95.40
55	88.05	95.85	96.1	**96.45**	95.20
60	87.6	95.8	95.5	96.4	95.40
65	87.65	95.6	96	96.35	95.40
70	87.65	95.95	95.9	96.15	**95.60**
75	87.45	**96.1**	**96.4**	96.3	95.30
80	88.15	95.8	95.6	96.25	95.20
85	**88.8**	95.8	95.3	96.45	95.30
90	87.45	95.85	96.2	96.45	95.20

4 Conclusion

A novel multi-stage framework has been introduced here for automatic sorting of multi-script postal documents. 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. The technique is also having potential applications in numeral based script identification schemes from multi-script document images. The designed framework may be extended to incorporate rest of the regional Indian scripts for potential applications in the Nation-wide postal automation system. One of the limitations 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. In such cases, scripts of the numeric string may be inferred from the script of the textual address parts, as far as practicable.

Acknowledgement

Authors are thankful to the CMATER, SRUVM, CSE Department, Jadavpur University, for providing infrastructural facilities during progress of the work. 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. Sinha, S., Pal, U., Chaudhuri, B.B.: Word–Wise Script Identification from Indian Documents. In: Marinai, S., Dengel, A.R. (eds.) DAS 2004. LNCS, vol. 3163, pp. 310–321. Springer, Heidelberg (2004)
2. Roy, K., et al.: A System for Wordwise Handwritten Script Identification for Indian Postal Automation. In: IEEE INDICON 2004, pp. 266–271 (2004)
3. Zhou, L., Lu, Y., Tan, C.-L.: Bangla/English Script Identification Based on Analysis of Connected Component Profiles. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 243–254. Springer, Heidelberg (2006)
4. Roy, K., et al.: A System towards Indian Postal Automation. In: Proc. of the 9th IWFHR, pp. 361–367 (2004)
5. Roy, K., et al.: A System for Indian Postal Automation. In: Proc. of the 8th ICDAR (2005)
6. Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M., Basu, D.K.: A Two-Pass Approach to Pattern Classification. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 781–786. Springer, Heidelberg (2004)
7. Basu, S., Sarkar, R., Das, N., Kundu, M., Nasipuri, M., Basu, D.K.: Handwritten Bangla digit recognition using classifier combination through DS technique. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 236–241. Springer, Heidelberg (2005)
8. Pal, U., et al.: Handwritten Numeral Recognition of Six Popular Indian Scripts. In: ICDAR 2007, pp. 749–753 (2007)
9. Wen, Y., et al.: Handwritten Bangla numeral recognition system and its application to postal automation. Pattern Recognition 40(1), 99–107 (2007)
10. Basu, S., et al.: Recognition of Pincodes from Indian Postal Documents. Soft Computing, 239–245
11. Basu, S., et al.: A Hierarchical Approach to Recognition of Handwritten Bangla Characters. Pattern Recognition 42(7), 1467–1484 (2009)
12. Nilson, N.J.: Principles of Artifcial Intelligence, pp. 21–22. Springer, Heidelberg