HVS inspired System for Script Identification in Indian Multi-script Documents

Peeta Basa Pati and A G Ramakrishnan*
Department of Electrical Engineering
Indian Institute of Science, Bangalore, INDIA – 560 012.

Abstract. Identification of the script of the text, present in multi-script documents, is one of the important first steps in the design of an OCR system. Much work has been reported relating to Roman, Arabic, Chinese, Korean and Japanese scripts. Though some work has already been reported in Indian scenerio, the work is still in its nascent stage. In this work we report a script identification algorithm at the word level. Initially, we deal with the identification of the script of words, using Gabor filters, in a bi-script scenerio. Later, we extend this to tri script and then, five-script scenerios. The combination of Gabor features with nearest neighbor classifier shows promising results. Words of different font styles and sizes are used.

Key Words: Gabor filter, script identification, prototype selection.

1 Introduction

India has 26 official languages represented with 13 unique scripts. Many official documents are multi-script in nature. OCR has the potential to integrate people of different languages for socio-economic purposes. Besides, there are other Asian countries, where multi-script documents exist. Thus identification of the script is one of the necessary challenges for the designer of OCR systems, dealing with such multi-script documents. Quite a few results have been reported in the literature, identifying the scripts in multi-script documents. However, very few of these works deal with script identification at the word level.

Spitz *et al.* [1] use the spatial relationship between the structural features of characters for distinguishing Han from the Latin script. Asian scripts (Japanese, Korean and Chinese) are differentiated from Roman by an uniform vertical distribution of upward concavities. In the case of the above Asian scripts, the measure of optical density (*i.e.* the number of ON-pixels per unit area) is employed to distinguish one from the other. Hochberg *et al.* [2] use cluster based templates for script identification. They consider thirteen different scripts including Devanagari, an Indian script used to write Hindi, Sanskrit, Marathi and Nepali languages. They cluster the textual symbols (connected components) and create a representative symbol or a template for each cluster. Identification is through comparison of textual symbols of the test documents with those of the templates.

* Corresponding Author: ramkiag@ee.iisc.ernet.in

However, the requirement of the extraction of connected components makes this feature a local one. Wood *et al.* [3] suggest a method based on Hough transform, morphological filtering, and analysis of projection profile. Their work involves the global characteristics of the text.

Tan [4] has suggested a method for identifying six different scripts using a texture based approach. Textual blocks of 128×128 are taken and filtered with angular spacings of 11.25^o. This method requires image blocks containing text of same script. Roman, Persian, Chinese, Malayalam, Greek and Russsian scripts, with multiple font sizes and styles (font invariance within the block), are identified.

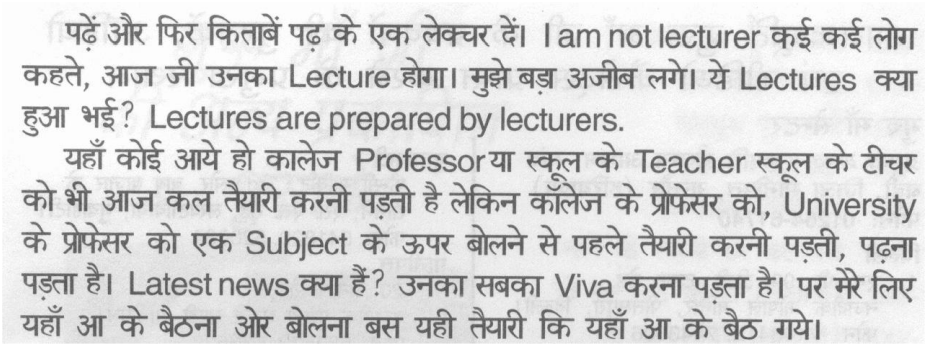
Pal and Chaudhuri [5] have proposed a method, based on a decision tree, for recognizing the script of a line of text. They consider Roman, Bengali and Devanagari scripts. They have used the projection profile, besides statistical, topological and stroke-based features. At the initial level, the Roman script is isolated from the other two, by examining the presence of the **headline**[1] (shirorekha). Devanagari is differentiated from Bangla by identifying the principal strokes [5]. In [6], they have extended their work to the identification of the script from a given triplet. Here, they have dealt with almost all the Indian scripts. Besides the headline, they have used some script-dependent structural properties, such as the distribution of ascenders and descenders, the position of the vertical line in a text block, and the number of horizontal runs.

Chaudhuri and Seth [7] have proposed a technique using features such as the horizontal projection profile, Gabor transform and aspect ratio of connected components. They have handled Roman, Hindi, Telugu and Malayalam scripts.

For most of the above reported works, the textual block taken for the script recognition task is either a mono-script line or a paragraph. However, in the Indian context, in the official documents, technical reports, magazines and application forms, the script could, in principle, vary at the word level. Figure 1 demonstrates this with an example bilingual document containing Devanagari and Roman scripts.

Recognition of the script, using statistical features, at word level has been reported by Dhanya *et al.* [8]. Here the authors tried to differentiate the Tamil and Roman scripts using various feature-classifier combinations. This work has been extended by Pati *et al.* [9]. Here the authors have identified Odiya and Hindi scripts, besides Tamil, against the Roman script. They use a bank of Gabor filters for feature extraction with linear discriminant classifier for decision making. Besides, redesign of the Gabor functions has allowed to enhance the system efficiency. Pal and Chaudhury [5] have tried to discriminate Roman, Devanagari and Bangla scripts at the word level using a set of structural features and a tree based classifier. They have extended their approach to identify scripts in another triplet, Roman, Devanagari and Telugu scripts [10]. Padma and Nagabhushan [11] have discriminated Roman, Devanagari and Kannada script on similar lines.

[1] Both Devanagari and Bangla scripts have a horizontal line at the top, known as ***shirorekha***, which connects the characters in a word


पढ़ें और फिर किताबें पढ़ के एक लेक्चर दें। I am not lecturer कई कई लोग कहते, आज जी उनका Lecture होगा। मुझे बड़ा अजीब लगे। ये Lectures क्या हुआ भई? Lectures are prepared by lecturers.

यहाँ कोई आये हो कालेज Professor या स्कूल के Teacher स्कूल के टीचर को भी आज कल तैयारी करनी पड़ती है लेकिन कॉलेज के प्रोफेसर को, University के प्रोफेसर को एक Subject के ऊपर बोलने से पहले तैयारी करनी पड़ती, पढ़ना पड़ता है। Latest news क्या हैं? उनका सबका Viva करना पड़ता है। पर मेरे लिए यहाँ आ के बैठना ओर बोलना बस यही तैयारी कि यहाँ आ के बैठ गये।

Fig. 1. A sample biscript document showing interspersed Hindi and English words.

Though some works have been reported for identifying the script of the Indian documents, the work is still in its infancy. In the present work, we explore the effectiveness of our approach [9] in recognizing up to three or five scripts.

2 Data Description

Our database consists of about 1000 scanned documents, containing combinations of different scripts. These documents are scanned from various sources such as (i) laser printed pages, (ii) printed books, (iii) magazines, (iv) newspapers and (v) official documents. We have collected about 4500 words each from Tamil, Kannada, Odiya, Devanagari and Roman scripts [12]. 3000 of these are selected randomly for each script, to form the training set, while the rest 1500 words form the test set. Most of these words have been segmented and collected from actual bilingual documents, with the script varying at the word level. We have assumed that a word contains at least two characters. These words contain a lot of variability in terms of font size and style, as well as the age and nature of the document from which they are collected. We also ensure that there are no blank rows or columns present in the word image.

3 System Description

HVS is best modelled by Gabor functions [2] [15], because the mammalian visual system involves a set of parallel filter channels which are quasi-independent [16] as well as direction dependent [17], at the cortical level. This supports the theory of multi-channel filtering of signals in systems that model biological vision [18]. This scheme has been successful in texture segmentation [19] and researchers have also used it for text page segmentation [20, 21].

[2] Gabor [13] introduced these functions, a Gaussian modulated by a complex sinusoid, in 1946 for one-D case which was later extended to 2D case by Daugman [14].

There are two approaches taken for script identification in a multi-script scenerio. One of them extensively studies the similarities and differences in the structures between the co-occuring scripts while the second method deals with each script as a different texture. In our view, the latter method is more robust as it deals with the script regardless of the size or style of the font. Thus, we are employing a multi-channel filtering approach, employing Gabor functions, for script identification. This claim of ours is supported by our earlier employment of these banks for successful page layout analysis [21] and script identification [8, 9]. We have used the Gabor filter bank, earlier designed and implemented for script identification by us [9].

The Gabor filter bank, we use, has three different radial frequencies (0.125, 0.25 & 0.5) and six different angles of orientation (θ = 0, 30, 60, 90, 120 and 150 degrees). Thus, we have used a radial frequency bandwidth of 1 octave and an angular bandwidth of 30°. The spatial spread of each filter along the $x-$ and $y-$coordinates are determined by the standard deviations of the Gaussians, σ_x and σ_y, respectively. Both of them are functions of the radial frequency and angular bandwidth. The three radial frequencies with six θ's give a combination of 18 odd and 18 even filters. A given word image is filtered with these 36 filters. A 36 dimensional feature vector is formed with the total energy in each of the filtered images being a feature.

We use linear discriminant (LD) and nearest neighbor (NN) classifiers as the decision makers. Each feature vector acts as a point in a $d-$dimensional space, where d (= 36) is the dimensionality of the feature vector. The NN classifier finds the closest neighbor of the test pattern in this space. It then assigns the class value of this closest pattern to the test pattern. Thus, in a bi-class case, if the test pattaern has a closet neighbor from class ω_1, then the test pattern is assumed to be from class ω_1. In LDC, however, we try to find a hyper-plane which best discriminates the classes in this d-dimensional space. In a bi-class case, this hyper-plane divides the feature space into two parts such that the training patterns of the two different classes lie on the opposite sides of this plane. Thus, when a test pattern lies on one side of the hyper-plane, it gets the class value meant for that side of the hyper-plane. A multi-class scenerio could be considered to be consisting of a number of 2-class scenerios, for LDC evaluation. Thus, by combining the 36-dimensional Gabor features with the LDC or NN classifier, we try to identify the script of a word in bi-script, tri-script and five-script cases.

3000 training patterns is a fairly large set per script. Moreover, quite a lot of samples could be actually very closely placed in the feature space. For deciding the discriminating hyper-plane, only few representing samples which lie on the border of the class are sufficient. Besides, a large training set consumes a lot of computational time while the NN classifier is at work. Thus, it is worth looking at prototype selection mechanisms, where a reduced set called a prototype set takes the place of the training or reference set. This being a small set has the advantage of combined space and computational time efficiency.

Susheela Devi and M N Murthy [22] have proposed an efficient incremental prototype set building algorithm. This they have tested over a variety of benchmark data sets and reported that it performs consistently. The algorithm proposed by them has two stages: (i) set growing stage and (ii) set reduction stage. While the set growing stage increases the number of prototypes till all the training samples are recognized, the set reduction technique removes those prototypes which are responsible only for self-classification. We have taken the training sets of all the five scripts and employed a NN classifier for this purpose. We have used only the set growing technique and declare the process to end when it achieves 100% recognition accuracy of the training set against the prototype set, for all the five classes.

4 Results

To confirm that our selection of test and training sets are class representive in nature, we tested the accuracy of recognition of the test set against that of the training and vice versa. We use the NN classifier for this purpose. The test set recognition accuracy against the train set is 96.0% while that of the train set against the test set is 94.7%. This shows that the selection of the test and train sets are fairly independent and are class representative in nature.

The prototype selection mechanism was able to reduce the training set to a much smaller set. At the end of the process, we had 291, 290, 258, 570, 485 samples representing the English, Hindi, Kannada, Odiya and Tamil scripts, respectively. Thus, we had a total of only 1894 samples representing 15000 training samples of all the five classes, saving about 87% of memory and computation.

For the five scripts, we have considered for our script identification experiments, we have 10 different bi-class problems. We report the accuracies of the bi-class script indentification in Table 1. The performance is presented in %age and give the average recognition accuracies of both the scripts involved. We are presenting the accuracies of the LD and NN classifiers together. In the table, the characters E, D, K, O and T represent the Roman (English), Devanagari (Hindi), Kannada, Odiya and Tamil scripts, respectively. It can be observed that both the LD and NN classifiers have performed very well with the Gabor feature vectors, though in most of the cases the NN classifier performs marginally better. The highest attainable accuracy for the test set against the full training set is 99.7% for the bi-class problem of Hindi vs. Kannada with a NN classifier while the lowest attainable, with the same classifier, is 97.3% for Odiya and Tamil combination. It could be noted that, the accuracies of the test set against the prototype set is also quite high, though trailing behind the full training set. This could be owing to the following reasons. Since we have tried to attain 100% accuracy for the prototype selection process, there is every likelihood that to attain that accuracy, the spurious patterns have also come into the prototype set. Thus, eliminating such outlier patterns by the technique proposed by Susheela Devi *et. al.* could be of help. We could also try to saturate the process little earlier than attaining 100% accuracy and see how such a prototype set behaves.

The observation of good bi-class accuracies makes us infer that the five classes are well separated in the feature space.

Table 1. Recognition rate for pairs of scripts involving various Indian scripts. The figures indicate the average correct recognition of words in both scripts of the test set against the train set and the prototype set using NN or LD classifiers.

			TRAIN SET					PROTOTYPE SET				
			E	D	K	O	T	E	D	K	O	T
TRAIN SET	LDC	E	–	99.1	99.2	97.8	97.4	–	98.7	98.7	97.6	96.0
		D	99.1	–	99.5	99.1	98.9	98.7	–	99.0	99.1	98.4
		K	99.2	99.5	–	98.0	99.2	98.7	99.0	–	97.1	99.2
		O	97.8	99.1	98.0	–	98.2	97.6	99.1	97.1	–	97.8
		T	97.4	98.9	99.2	98.2	–	96.0	98.4	99.2	97.8	–
	NNC	E	–	99.4	99.6	98.5	98.8	–	98.9	99.5	97.4	97.6
		D	99.4	–	99.7	99.4	99.2	98.9	–	99.5	98.8	98.0
		K	99.6	99.7	–	98.2	99.4	99.5	99.5	–	96.2	99.0
		O	98.5	99.4	98.2	–	97.3	97.4	98.8	96.2	–	95.6
		T	98.8	99.2	99.4	97.3	–	97.6	98.0	99.0	95.6	–

Most official documents in India contain three scripts. They are English, Hindi and the local script of the place where the document is used. In the second series of experiments, we have attempted discrimination between three scripts, out of which two are Roman and Devanagari. Based on the experiments on pairs of scripts, we expected good separation between scripts in a triscript scenario using both of the LD and NN classifiers. The results of such experiments are presented in Tables 2 and 3 for LDC and NNC, respectively. The test sets have again been tested for their recognition accuracies against the train and the prototype sets.

While the recognition accuracies are generally good, it is observed that Tamil performs the worst when compared against the combination of Hindi and English scripts (see table 2). The discrimination between Tamil and combined Hindi and English is low at 96.4% for the full train set and at 94.4% for prototype set. Here, also, the recognition accuracies obtained with the prototype set closely follows the accuracies with the train set. When we compare the results presented in Table 3 (obtained from NNC) with those in Table 2 (obtained from LDC), NN classifier is observed to fare better than the LDC. This is in confirmation with the results for bi-class experiments. For the NN classifier, it is again Tamil script which is observed to be having the lowest average accuracy amongst the three local scripts, for both train and prototype sets.

We have compared our results with the earlier reported results of Pal & Chaudhuri [10] and Padma & Nagabhushan [11]. Pal & Chaudhuri have reported an accuracy of 97.2 % for recognition of Devanagari words while Padma & Nagabhushan have reported 97 % for the same scripts. We achieve a minimum of 97.6 % for the recognition of Devanagari words, against Kannada and

Table 2. Average recognition accuracies for script triplets comprising of Roman (E), Devanagari(D) and Indian local scripts(L). Here L/DE means the average recognition accuracy of the local script against the combined Hindi and English scripts. Values shown are percentages for Gabor feature with LDC as the decision maker.

	TRAIN			PROTOTYPE		
	K	O	T	K	O	T
L/DE	98.7	97.3	96.4	98.4	97.1	94.4
D/EL	98.7	99.0	98.6	97.6	98.7	98.3
E/DL	98.9	97.8	98.2	98.7	96.9	97.7

Table 3. Triscript recognition rate (in percentage) for the scripts of Kannada (K), Odiya (O) and Tamil (T) with English and Hindi. The Gabor features have been used with NN classifier, against both the training and prototype sets.

	TRAIN			PROTOTYPE		
	K	O	T	K	O	T
English	98.9	96.9	97.5	98.8	96.1	96.8
Hindi	98.7	98.9	98.3	98.1	97.4	97.5
Local	99.3	98.9	97.3	99.3	97.6	96.1
Average	99.0	98.2	97.7	98.7	97.0	96.8

English. This is the worst case we have when LDC is the decision maker and the test set is compared against the prototype set. The system performs better for combinations involving Kannada and Odiya as local scripts. The discrimination rate of Devanagari words against the other two is 98.7 and 98.3 %, for Odiya and Tamil as the local scripts, with prototype set, respectively.

Thus a system, using LDC, employed for the discrimination between scripts in a tri-script scenerio involving Kannada, Hindi and English scripts would involve the separation of English at the first level with an accuracy of 98.9% and the separation between Kannada and Hindi at the second level, with an accuracy of 99.5%. This makes the overall accuracy of the system to be 98.4% with the train set. A similar arrangement with the prototype set gives 97.7%. The average recognition accuracies for such a tri-script scenerio, employing NNC as the decision maker, is 99.0% and 98.7% against the train and prototype sets, respectively. Thus our system, at all its configurations outperforms the system reported by Padma & Nagabhushan which has a maximum achievable accuracy of 96.7% for the same tri-script scsnerio.

Since our identification scheme depends on statistical rather than structural features, we have an advantage of taking any number of scripts and identifying them. On the observation that our feature classifier combinations are delivering us with very good recognition accuracy, we tried to identify the scripts in a 5-script scenerio. In this case, any test sample is compared with the reference samples from all the classes by a NN classifier. The LDC classifier finds a discriminant function, which separates the script associated with the func-

tion, from rest four scripts. The test pattern is checked with all the discriminant functions for a score which measures its distance from the respective discriminant hyper-planes. The test pattern belongs to the script of the discriminant function yielding the maximum score. Table 4 presents the results of the script identification in this scenerio. Consistent with our earlier recorded results, the combination of the NN classifier with the train set yeilds the best average result. Interestingly, while Kannada fares low with LDC, it is the Tamil and Odiya scripts that fare low with NN classifier.

Table 4. Recognition rate (in percentage) for a penta-class case involving all the five scripts, using the 36-dimensional Gabor feature with LD and NN classifiers.

	TRAIN		PROTOTYPE	
	LDC	NN	LDC	NN
English	95.8	96.1	94.4	94.9
Hindi	97.1	97.5	96.9	95.9
Kannada	89.9	97.2	76.0	95.7
Odiya	93.5	94.5	96.1	91.1
Tamil	91.3	94.5	89.6	91.7
Average	93.5	96.0	90.6	93.9

5 Conclusion and Discussion

The combination of Gabor filter bank with either LD or NN classifier handles the issue of script recognition at the word level reasonably well. LDC closely follows NNC when a comparison is made between the classifiers for all cases. However, the actual performance is highly script dependent. For example, the overall classification performance, using the train set, for the tri-script combination involving Kannada, Devanagari and English is 98.4%, whereas the average correct recognition is only 97.4% for the bi-script combination of Tamil with English, and it is only 97.8% when Odiya is recognized against English. These are the results using the LDC. On similar lines, when we consider NNC, the average tri-script recognition accuracy for the triplet using Kannada as the local script is 99.0% while the bi-script cases of Odiya and English is 98.5% and that of the Odiya against Tamil is only 97.3%. Hindi script has a recognition accuracy of 97.5%, using NNC, against all the five scripts being present in the train set while the recognition accuracy of Roman script from the triplet having Odiya as the local scipt is 96.9%.

The prototype selection mechanism has been successfully able to reduce the train set by about 87%. But the cardinality of the sets for different scripts shows the sets are highly skewed. Kannada gets represented by less than half the number of prototypes needed for the Odiya script. Similarly, Tamil script needs a little less than double the number of prototypes present for Kannada

script. This gives us an impression that these two scripts, Odiya and Tamil, are relatively spread out in the feature space while the other three scripts form compact clusters. When we look at the accuracy results recorded in Tables 1, 2, 3 & 4, we can observe that it is these two scripts which have been consistently faring low in all cases. This shows that even the test patterns are so widely distributed as to get mis-classified with greater likelihood. Thus, we infer that these two scripts are well spread out in the feature space. Despite all this, the results substantiate our assumption that the HVS inspired system is well suited for script identification in multi-script documents. However, this needs to be tested with other Indian and non-Indian scripts. Further, it will be interesting to compare the results against other features.

References

1. Spitz, A.L.: Determination of Script and Language Content of Document Images. IEEE transaction on Pattern Analysis and Machine Intelligence **19** (1997) 235–245
2. Hochberg, J., Kelly, P., Thomas, T., Kerns, L.: Automatic script identification from document images using cluster based templates. IEEE transaction on Pattern Analysis and Machine Intelligence **19** (1997) 176–181
3. Wood, S.L., Yao, X., Krishnamurthi, K., Dang, L.: Language identification for printed text independent of segmentation. In: Proc. of Intl. Conf. on Image Processing. (1995) 428–431
4. Tan, T.N.: Rotation invariant texture features and their use in automatic script identification. IEEE transaction on Pattern Analysis and Machine Intelligence **20** (1998) 751–756
5. Chaudhuri, A.R., Mandal, A.K., Chaudhuri, B.B.: Page layout analyser for multilingual indian documents. In: Proceedings of the Language Engineering Conference. (2002) 24–32
6. Pal, U., Chaudhuri, B.B.: Script line separation from Indian muli-script document. In: Proceedings of the International Conference on Document Analysis and Recognition. (1999) 406–409
7. Chaudhuri, S., Seth, R.: Trainable Script Identification Strategies for Indian languages. In: Proceedings of the International Conference on Document Analysis and Recognition. (1999) 657–660
8. Dhanya, D., Ramakrishnan, A.G., Pati, P.B.: Script identification in printed bilingual docuements. Sadhana **27** (2002) 73–82
9. Pati, P.B., Raju, S.S., Pati, N.K., Ramakrishnan, A.G.: Gabor filters for document analysis in indian bilingual documents. In: Proc. of the Int. Conf. on Intelligent Sensing and Information Processing. (2004) 123–126
10. Pal, U., Sinha, S., Chaudhury, B.B.: Word-wise script identification from a document containing english, devanagari and telugu text. In: Proc. of National Conf. on Document Analysis and Recognition. (2003) 213–220
11. Padma, M.C., Nagabhushana, P.: Identification and separation of text words of kannada, hindi and english languages through discriminating features. In: Proc. of National Conf. on Document Analysis and Recognition. (2003) 252–260
12. Pati, P.B.: Indian Script Word Image Dataset, (www.ee.iisc.ernet.in/new/people/students/phd/pati/)
13. Gabor, D.: Theory of communication. J. IEE (London) **93** (1946) 429–457
14. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Am. A **2** (1985) 1160–1169

15. Marcelja, S.: Mathematical description of the response of simple cortical cells. J. Opt. Soc. Am. **70** (1980) 1297–1300
16. Campbell, F.W., Robson, J.G.: Application of Fourier analysis to the visibility of gratings. J. Physiol. **197** (1968) 551–566
17. Morrone, M.C., Burr, D.C.: Feature detection in human vision: a phase dependent energy model. Proc. Roy. Soc. Lon.(B) **235** (1988) 221–245
18. Porat, M., Zeevi, Y.Y.: The generalized gabor scheme of image representation in biological and machine vision. IEEE transaction on Pattern Analysis and Machine Intelligence **10** (1988) 452–467
19. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using Gabor filters. Pattern Recognition **24** (1991) 1167–1186
20. Chan, W., Coghill, G.: Text analysis using local energy. Pattern Recognition **34** (2001) 2523–2532
21. Raju, S.S., Pati, P.B., Ramakrishnan, A.G.: Gabor filter based block energy analysis for text extraction from digital document images. In: Proc. of the First Int. Workshop on Document Image Analysis for Libraries (DIAL'04). (2004)
22. Devi, V.S., Murthy, M.N.: An incremental prototype set building technique. Pattern Recognition **35** (2002) 505–513