A Blind Indic Script Recognizer for Multi-script Documents

Peeta Basa Pati & A G Ramakrishnan
Medical Intelligence and Language Enginering Laboratory
Department of Electrical Engineering
Indian Institute of Science, Bangalore, INDIA – 560012.

Abstract

We report a hierarchical blind script identifier for 11 different Indian scripts. An initial grouping of the 11 scripts is accomplished at the first level of this hierarchy. At the subsequent level, we recognize the script in each group. The various nodes of this tree use different feature-classifier combinations. A database of 20,000 words of different font styles and sizes is collected and used for each script. Effectiveness of Gabor and Discrete Cosine Transform features has been independently evaluated using nearest neighbor, linear discriminant and support vector machine classifiers. The minimum and maximum accuracies obtained, using this hierarchical mechanism, are 92.2% and 97.6%, respectively.

Key Words: Gabor filter, DCT, script identification.

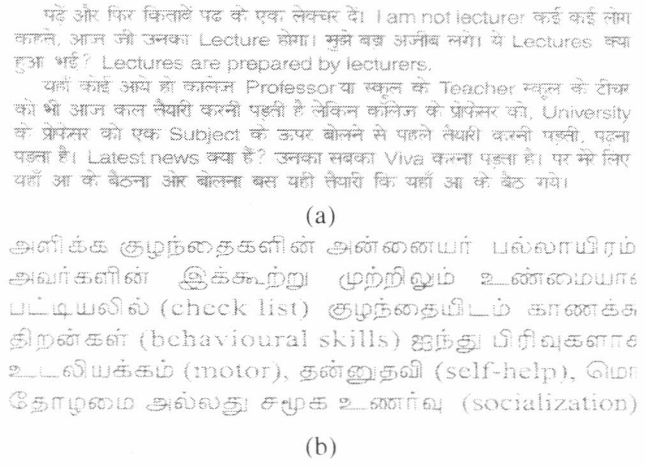

Figure 1. Sample bi-script documents showing English words interspersed in (a) Hindi & (b) Tamil documents.

1 Introduction

Demand for tools with capability to recognize, search and retrieve documents from multi-script and multi-lingual environments, has increased many folds in the recent years. Thus, recognition of the script and language play an important part for automated processing and utilization of documents. Plenty of research has been carried out for accomplishing this task of script recognition at a paragraph/block or line level. While the former assumes that a full document page is of the same script, the latter imagines documents to contain text from multiple scripts but changing at the level of the line. Though the latter is a realistic assumption in some cases, most of the practical situations have the script changing with words. Fig. 1 shows two bi-script documents where the presence of interspersed English words in Hindi and Tamil documents, respectively, is clearly seen.

Script identification acts as a preliminary level of filtering to reduce the search complexity. Moreover, for scripts such as Devanagari and Bengali, identification of script helps decide the further course of processing, such as, removal of the *shirorekha*[1] (the headline) from the word to separate the constituent symbols and recognize them. Thus, identification of the script is one of the necessary challenges before the designer of OCR systems, dealing with multi-script documents.

Most algorithms developed for identifying Indic scripts work on paragraph or line images [1]. However, in general, the scripts in Indic multi-lingual documents change with words. In this work, efficacy of various combinations of two different features and three different classifiers is evaluated for the script recognition task at a word level. Different script recognition scenarios (bi-script, tri-script & eleven-script), are explored. The observations of these experiments are utilized to design a hierarchical & blind script recognition system, where the script of the words may blindly change from one to the other of the 11 scripts.

Eleven different Indic scripts, namely, Bengali, Roman,

[1] All the characters in a word are joined together by a headline in these scripts to form a word. Refer figure 1 for examples.

Gujarati, Devanagari, Kannada, Malayalam, Odiya, Gurumukhi, Tamil, Telugu and Urdu, are considered for their recognition in this work. A set of 20000 words are collected from each of the 11 scripts and binarized before use [1]. The words are cleared of any blank rows or columns from the beginning and the end of the words (*i.e.*, the word is made to touch the boundary of each word matrix) without any normalization to the size. A random selection of 7000 of these words, in each script, is used to make the training set while the rest 13000 words are used for testing.

2 System Description

Two approaches can be pursued for script identification in a multi-script scenario. One of them extensively studies the similarities and differences in the structures between the co-occurring scripts [2, 3, 4], while the second method treats each script as a different texture [5, 6, 7]. The textural method is more robust because it deals with the script regardless of the size or style of the font. We, therefore, employ a multi-channel filter bank, using Gabor functions. A Gabor filter is defined by its three independent parameters: (i) radial frequency, (ii) angle of orientation, and (iii) sine/cosine function. We denote such a filter by ${}_cH^u_\theta$ where the subscript 'c' to the left of H represents a cosine filter (this becomes 's' for a sine filter); 'u' & 'θ' denote the radial frequency and angle, respectively. The image (W) is filtered by spatial convolution with the filter (H).

$$\hat{W} = W * {}_cH^u_\theta \tag{1}$$

$${}_cf^u_\theta = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C} \hat{W}^2(i,j)}{\sum_{i=1}^{R}\sum_{j=1}^{C} W^2(i,j)} \tag{2}$$

where $\hat{W}$ (in Eqn. 1) is the output image. Energy of the filtered image, f, is evaluated by adding the squares of pixel values in the output image. This energy is normalized by the input energy of the word image to make a feature (see Eqn. 2). Thus, each feature is associated with a specific combination of the three different filter parameters. We chose to use a radial frequency bandwidth of one octave and an angular bandwidth of 30° for this experiment. So, three different radial frequencies (0.125, 0.25 & 0.5) and six angles of orientation (0, 30, 60, 90, 120 & 150 degrees) are used. The three radial frequencies with six θ's give a combination of eighteen odd and eighteen even filters. A given word image is filtered with these thirty-six filters to generate a thirty-six dimensional feature vector, F, as mentioned in equation 3. These feature vectors are used for the identification of the script of the word. Figure 2(a) shows a block diagram of the Gabor feature extraction process.

$$F = \left[{}_cf^{u_1}_{\theta_1}, \cdots, {}_cf^{u_3}_{\theta_6}, {}_sf^{u_1}_{\theta_1}, \cdots, {}_sf^{u_3}_{\theta_6}\right] \tag{3}$$

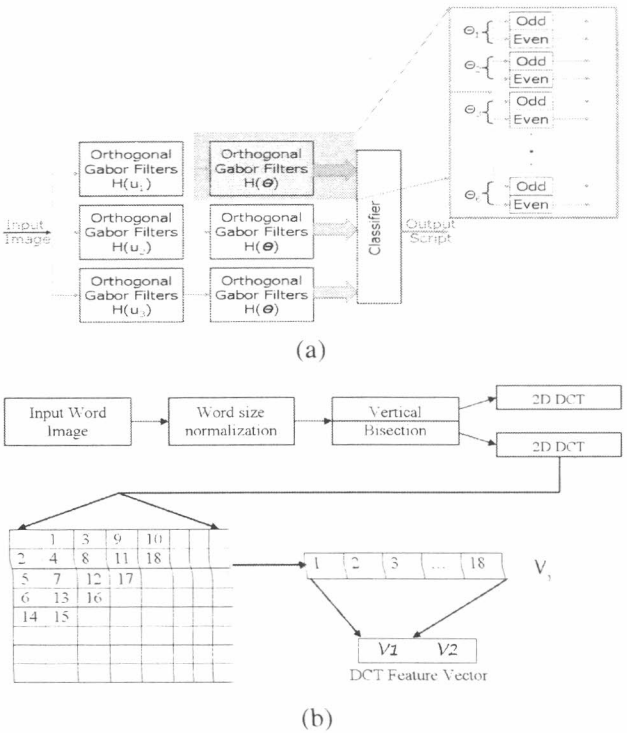

Figure 2. Block diagrams of (a) Gabor and (b) DCT feature extractors.

Discrete Cosine Transform (DCT) concentrates the information content in a relatively few coefficients. For natural signals and images, the data compaction of DCT is close to that of the optimal KL transform (KLT). The energy compactness property of DCT justifies its use for script identification. Figure 2(b) diagrammatically presents the extraction of the DCT feature vector. Initially, the input word image is normalized to a standard size. It is then vertically divided into two equal blocks and 2-D DCT is performed on each of the block, independently. As shown in Fig. 2(b), eighteen low frequency coefficients are chosen in a zig-zag fashion from this DCT matrix of each half of the word image. The vectors are appended to form a 36-dimensional feature vector, which is used for classification. We have taken 36 coefficients for a fair comparison with the Gabor filter based system.

We have used three different classifiers to decide about the script of the test words: (i) the nearest neighbor classifier (NNC), (ii) linear discriminant classifier (LDC), and (iii) the support vector machines (SVM's). In NNC, Euclidean distance of the test pattern is evaluated in the feature space, with each of the training patterns. The class value of the nearest neighbor is assigned to the test pattern. A linear discriminant function partitions the feature space using a hyper-plane where the two sides of this plane represent

Table 1. The mean and standard deviation (STD) of the Bi-Script recognition accuracies with various feature classifier combinations. The statistics of the accuracies are presented in % and involve all the 55 different bi-script scenarios.

	Gabor			DCT		
	NNC	LDC	SVM	NNC	LDC	SVM
Mean	98.4	98.3	98.4	98.3	88.5	97.8
STD	2.0	1.5	1.7	1.5	4.4	1.4

the two classes. The class value of the test pattern is decided based on which side of the plane it lies. A multi-class scenario could be handled as a number of bi-class scenarios. Using the Support Vector Machine [8], optimal hyperplanes are generated that decide the separation between individual classes of patterns. The creation of a unique model to represent a class, derived by training the model with prototypes of each class, aids in maximization of the correct classification rate. We have used the **SVMTorch – II** toolbox [9].

3 Experiments and Results

3.1 Bi-script Recognition

The bi-script documents, such as books, newspapers and magazines have an Indic script as the major script with interspersed English words. Besides, in the border areas of the states, people know more than one language and the documents reflect that. Thus, a good recognition in bi-script scenario is very useful. With eleven scripts, we have 55 different bi-class problems. The mean and the variance of such accuracies with various feature classifier combinations are presented in the table 1.

Close inspection of Table 1 reveals that though both the features have fared well with NNC, the Gabor (mean (μ) = 98.4, standard deviation (σ) = 2.0) has performed slightly better than the DCT features ($\mu = 98.3$, $\sigma = 1.5$). The same trend has become much more dominant with linear discriminant classifier where the gap between Gabor ($\mu = 98.3$, $\sigma = 1.5$) and DCT ($\mu = 88.5$, $\sigma = 4.4$) features is widened. With SVM as the classifier, the Gabor ($\mu = 98.4$, $\sigma = 1.7$) again leads the performance (for DCT: $\mu = 97.8$, $\sigma = 1.4$). Thus, it may be established that for bi-script recognition, Gabor may be preferred over DCT, though for some specific scripts, DCT has outperformed the Gabor. Moreover, SVM and NNC have performed comparably and with consistency. Combinations of LDC, for both features, yield the least.

Table 2. The mean and standard deviation (STD) of the Tri-Script recognition accuracies with various feature classifier combinations. The statistics of the accuracies are presented in %. Each triplet involves a combination of Devanagari, Roman and one of the local scripts.

	Gabor			DCT		
	NNC	LDC	SVM	NNC	LDC	SVM
Mean	98.2	96.9	98.2	97.9	83.7	98.1
STD	1.9	1.8	2.1	1.2	2.9	1.0

The most frequent scenario that warrants script recognition involves the official documents by State Governments and the text books in state languages. These documents involve the script of the official language of the state and English. The recognition rates we have achieved for this important application exceed 99% for all the biscript combinations, using the NN classifier. The lowest performance is for Malayalam, which is 99.1%, whereas the best performance is for Kannada, which is 99.9%. From this table, it may be noted that Gabor features score a clear margin over DCT with all three classifiers.

3.2 Tri-script Recognition

A number of official documents in India contain three languages, namely, the state's official language, Hindi and English. We refer to such a combination as the triplet of the state. In the second series of experiments, we discriminate the scripts in all such triplets. Here, Devanagari & Roman are common to all the triplets while the script of the state varies. The results of such experiments are presented in Table 2. In this table, the Mean and STD rows report the average and standard deviation, respectively, of the recognition rates for all the nine triplets.

Table 2 shows that the maximum average accuracy obtained is 98.2%. This result occurs for Gabor features with NN and SVM classifiers. This result is a clear improvement of 1% over the results reported in [3, 4] for Telugu and Kannada triplets involving a smaller dataset. However, the Gabor-NNC combination is more consistent and results in a lower standard deviation ($\sigma_{NNC} = 1.9$ while $\sigma_{SVM} = 2.1$). So, the combination of the Gabor feature with the NNC has a slight edge. A prior knowledge of the triplet, however, may decide in favor of a different feature-classifier combination for a triplet-specific optimal result.

Table 3. The statistics of the 11-script recognition accuracies. These statistics are generated from the accuracy of self-classification for each class.

	Gabor			DCT		
	NNC	LDC	SVM	NNC	LDC	SVM
Mean(μ)	89.4	83.1	94.8	91.1	49.8	93.8
STD(σ)	4.4	8.8	1.9	4.1	18.3	3.9

Table 4. Statistics of Grouping accuracy of the eleven scripts with the various features & classifiers.

	Gabor			DCT		
	NNC	LDC	SVM	NNC	LDC	SVM
Mean(μ)	96.2	90.2	97.7	95.1	62.9	96.9
STD(σ)	0.1	4.1	0.3	3.0	18.4	1.3

3.3 Eleven-script Recognition

We observe that our feature – classifier combinations are delivering us very good recognition accuracy thus far. So, we tried to identify the scripts in a scenario, involving all the 11 Indic scripts. Here, every test sample is compared with the reference samples from all the classes. We compare the efficacies of the Gabor and the DCT features, with the three classifiers. The average recognition accuracy is evaluated by considering the correct recognition accuracy of each of the presented 11 scripts. The average and standard deviation of these accuracies for all combinations are presented in Table 3. This table shows that the Gabor-SVM combination has an edge, with an average accuracy of 94.8% followed by the DCT-SVM accuracy of 93.8%. Similarly, the Gabor feature leads with a performance of 83.1% using the LDC over the DCT (49.8%). On the contrary, it is the DCT which better combines with the NNC to generate 91.1% while the Gabor with the same classifier results in a slightly lower performance of 89.4%.

Using Gabor features with the NNC, the worst performance of 83.8% is for Telugu, and the best performance is for English, namely, 97.8%. Similar performances are repeated with the SVM as the classifier. Thus, in a multi-script scenario, it makes better sense to separate the English words first, and the remaining scripts in a hierarchical manner, subsequently.

Table 5. Intra-group accuracy statistics for classification of the groups G1, G2, G3 and G4 scripts, using combinations of various features & classifiers. In this table the average and standard deviation of the accuracies are presented with μ and σ, respectively.

		Gabor			DCT		
		NNC	LDC	SVM	NNC	LDC	SVM
G1	μ	90.9	90.2	95.7	92.4	70.3	94.5
	σ	4.3	7.5	1.7	1.9	8.0	1.1
G2	μ	91.6	90.4	96.3	94.0	68.3	94.6
	σ	2.3	2.3	0.5	2.5	4.2	5.0
G3	μ	89.2	92.9	93.6	93.4	74.8	95.7
	σ	4.3	5.3	0.6	2.4	2.4	0.8
G4	μ	99.8	99.5	99.7	98.9	91.4	99.6
	σ	0.0	0.4	0.3	0.1	7.1	0.5

3.4 A Blind Script Recognizer

A close observation of the confusion matrices for the 11-script scenario, reveals that some scripts tend to group among themselves. A neighborhood analysis of the scripts shows that such scripts share a large number of neighbors. These scripts should be considered as a group at an initial level and then discriminated among themselves. So, a multi-level identification strategy is proposed. It is also possible to employ different feature-classifier combinations at different nodes of such a hierarchical classification system. Then, the best-minimal subset of features can be used at each node for maximizing the efficiency of this system.

Four groups are formed from the eleven scripts based on their visual similarity and class inter-mixing in the feature spaces. The four groups are: (i) G1 – Devanagari, Bengali and Punjabi, (ii) G2 – Gujarati, Malayalam, Odiya & Tamil, (iii) G3 – Kannada and Telugu, and (iv) G4 – Roman and Urdu. A test pattern is initially assigned to one of these groups and gets a script identity, subsequently. Table 4 tabulates the efficacy of such a grouping strategy, with various feature-classifier combinations. Here, it is observed that the SVM has marginally out-performed the NNC and the accuracies with LDC are the lowest for both Gabor and DCT features. The Gabor–SVM combination yields the maximum average grouping accuracy.

On similar lines, the different intra-group classification accuracies for all the four groups with various feature-classifier combinations are presented in Table 5. The notations μ and σ are used to represent the mean and the standard deviation, respectively, of the accuracies in any category. It is observed that the SVM classifier has, for both Gabor and DCT features, produced comparable performance

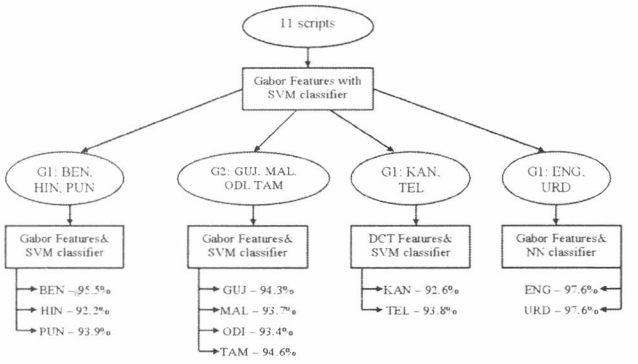

Figure 3. A hierarchical blind script classification system. The numbers shown with the script names are the final recognition accuracies in percentages.

with NNC for all the groups. However, the best performances are group specific. Gabor–SVM combination leads in G1 and G2 while Gabor in combination with NNC has the best results for G4. DCT–SVM is the winner for G3. This performance of DCT over the Gabor features is expected, as DCT has better accuracies for the bi-script scenario involving members of this group. With these results, a 2-layered hierarchical blind script identifier is designed. In this tree, the initial grouping is accomplished with Gabor-NNC combination while the second level has four nodes, with each node using the best feature-classifier combination for that group. This tree structured blind recognizer, with the selected feature-classifier combination and associated accuracies at various levels, is shown in figure 3. With such a structure, we achieve a maximum accuracy of 97.6% for both of Roman & Urdu scripts while a minimum of 92.2% is obtained for Devanagari script.

4 Discussion & Conclusion

To our knowledge, this is the most comprehensive work involving recognition of most of the Indic scripts. Further, we have evaluated the performance on a very large data set, and with the realistic assumption that the script changes at the level of the word. This system is seen to work with different kinds of variations, such as font sizes and styles, for any script. This, also, is observed to efficiently perform at bi-script, tri-script and eleven-script scenarios. Thus, we are confident that this methodology can be extended to recognize combinations of other scripts not included in the above experiments, with the only possible fine tuning of the Gabor filter parameters.

A comparison between the Gabor and DCT features, shows Gabor to be leading in many of the bi-script and tri-script cases. Design of a blind script identifier using a tree structure has yielded good results. We feel that employing feature reduction techniques together with fine tuning of the SVM kernel parameters is likely to further enhance the accuracy at each nodal point, finally contributing to the overall increase in performance.

References

[1] P. B. Pati, *Document Analysis for Complex layout & Content and Multi-lingual Documents*. PhD thesis, Indian Institute of Science, Bangalore, 2007.

[2] A. L. Spitz, "Determination of Script and Language Content of Document Images," *IEEE Trans. PAMI*, vol. 19, no. 3, pp. 235–245, 1997.

[3] A. R. Chaudhuri, A. K. Mandal, and B. B. Chaudhuri, "Page layout analyser for multilingual Indian documents," *Proc. Lang. Engg. Conf.*, pp. 24–32, 2002.

[4] M. C. Padma and P. Nagabhushana, "Identification and separation of text words of Kannada, Hindi and English languages through discriminating features," *Proc. National Conf. on Doc. Anal. & Rec.*, pp. 252–260, 2003.

[5] D. Dhanya, A. G. Ramakrishnan, and P. B. Pati, "Script identification in printed bilingual documents," *Sadhana*, vol. 27, no. 1, pp. 73–82, 2002.

[6] P. B. Pati and A. G. Ramakrishnan, "HVS inspired system for script identification in indian multi-script documents," 7^{th} *IAPR Workshop on Doc. Anal. Sys. 2006, LNCS-3872*, (New Zealand), February 2006.

[7] S. Jaeger, H. Ma, and D. Doermann, "Identifying script on word-level with informational confidence," *icdar*, pp. 416–420, 2005.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining & Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.

[9] "SVMTorch – II," IDIAP Research Institute, Martigny, Switzerland, www.idiap.ch/learning/SVMTorch.html, last visited: Oct, 2006.