Word Level Multi-script Identification

Peeta Basa Pati[1] and A G Ramakrishnan

MILE Laboratory, Department of Electrical Engineering,
Indian Institute of Science, Bangalore, INDIA.

Abstract

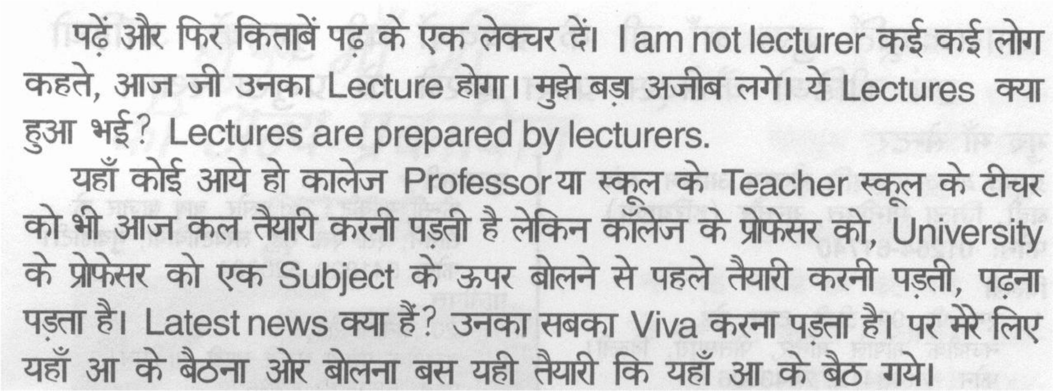
We report an algorithm to identify the script of each word in a document image. We start with a bi-script scenario which is later extended to tri-script and then to eleven-script scenarios. A database of 20 000 words of different font styles and sizes has been collected and used for each script. Effectiveness of Gabor and DCT features have been independently evaluated using nearest neighbor, linear discriminant and SVM classifiers. The combination of Gabor features with nearest neighbor or SVM classifier shows promising results; *i.e.*, over 98% for bi-script and tri-script cases and above 89% for the eleven-script scenario.

Key words: Gabor filter, DCT, script identification.

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 & 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 has the script changing with words. In figures 1 (a) & (b), we show two text

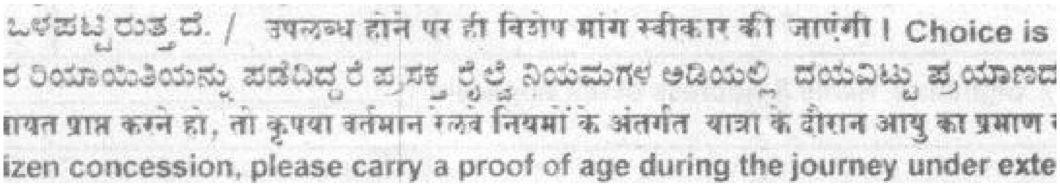
[1] Corresponding Author:
Department of Electrical Engineering, Indian Institute of Science,
Bangalore, INDIA - 560012, Fax: +918023600444.
e-mail: pati@ee.iisc.ernet.in



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।

(a)



ಒಳಪಟ್ಟಿರುತ್ತದೆ. / उपलब्ध होने पर ही विशेष मांग स्वीकार की जाएंगी। Choice is
ರ ರಿಯಾಯಿತಿಯನ್ನು ಪಡೆದಿದ್ದರೆ ಪ್ರಸಕ್ತ ರೈಲ್ವೆ ನಿಯಮಗಳ ಅಡಿಯಲ್ಲಿ ದಯವಿಟ್ಟು ಪ್ರಯಾಣದ
ायत प्राप्त करते हो, तो कृपया वर्तमान रेलवे नियमों के अंतर्गत यात्रा के दौरान आयु का प्रमाण
izen concession, please carry a proof of age during the journey under exte

(b)

Fig. 1. Sample documents to demonstrate the variation of script at the word level. (a) a bi-script document showing interspersed Hindi and English words. (b) a tri-lingual railway reservation form with words from Kannada, Devanagari and Roman scripts; the first line contains words from all the three scripts.

images where the script changes at the word level. Fig. 1(a) shows a bi-script document where the presence of interspersed English words in a document of Devanagari script is clearly seen. Similarly, Fig. 1(b) shows the variation of script at both line and word level. It is important to mention that, many researchers assume that multi-script documents generally contain text from two scripts. With the figure 1(b), we emphasize the presence of three scripts in a document, which is a common occurrence in India.

Most of the OCR systems are designed using statistical pattern recognition techniques. It is generally observed that these systems generate good output for specific kinds of documents and when the number of classes is reasonable. Including all the various symbols used for writing in the world, together in one reference set as different classes will be prohibitively high. Most of the Indian scripts have 13 vowels and about 35 consonants. Unlike Roman script, in Indian scripts, a consonant combines with another consonant or a vowel to generate a completely different symbol. This is demonstrated in figure 2 where two different symbols combine to generate a completely new symbol. Figure 2(a) presents the combination of two consonants in Odiya script while Fig. 2(b) presents a sample case in Devanagari. Thus the CV and CCV combinations, which appear frequently, generate a huge set of graphemes.

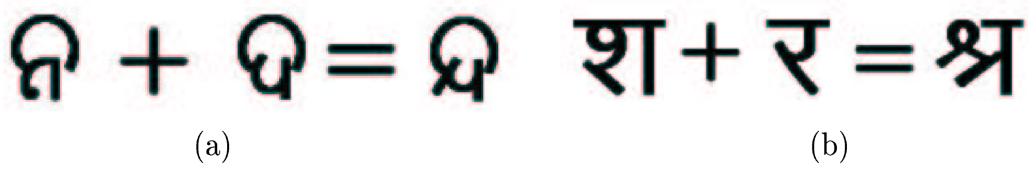


Fig. 2. Example of to consonants combining to form a completely new symbol for (a) Odiya script, and (b) Devanagari (Hindi) script.

In Telugu script alone, Rajasekaran and Deekshatulu [1,2] have identified some 2000 symbols which are used regularly. Kannada, with a very similar script & rules, has comparable number of symbols. Devanagari and similar scripts have close to 6000 such combinations each and other scripts have around 300 symbols. Thus, script identification can act as the preliminary level of filter and reduce the complexity of the search for classifying a test pattern. Moreover, for scripts such as Devanagari and Bangla, its identification decides the further course of processing. This includes removal of the *shirorekha*[2], the headline, from the word to separate the different symbols forming the word so that each of them can be individually recognized. Thus, identification of the script is one of the necessary challenges for the designer of OCR systems when dealing with multi-script documents.

2 Literature Review

Quite a few results have been reported in the literature, identifying the scripts at the level of paragraphs or lines. In sub-section 2.1 below, we review this literature. However, very few research works deal with script identification at the word level, which we review in subsection 2.2.

2.1 Script variation with Paragraphs/Blocks & Lines

Spitz [3] uses the spatial relationship between the structural features of characters for distinguishing Han from the Latin script. 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 number of ON-pixels per unit area is employed to distinguish one from the other. Hochberg *et al.* [4] use cluster based templates for script identification. They consider thirteen different scripts including Devanagari. They cluster the textual symbols (connected components) and create a representative symbol or a template for

[2] Both Devanagari and Bangla scripts have a horizontal line at the top, known as ***shirorekha*** or headline, which connects the characters in a word. Refer images in Fig. 1 for examples.

each cluster. Identification is through comparison of textual symbols of the test documents with those of the templates. However, the requirement of the extraction of connected components makes this feature a local one. Wood *et al.* [5] suggest a method based on Hough transform, morphological filtering, and analysis of projection profile. Their work involves the global characteristics of the text.

Chaudhuri *et al.* [6] 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**. Devanagari is differentiated from Bangla by identifying the principal strokes [6]. In [7], Pal & Chaudhuri have extended the above work to the identification of the script from a given triplet consisting of Devanagari, Roman and a state language, where each line of text contains only a single script. 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.

Tan [8] 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 Russian scripts, with multiple font sizes and styles (font invariance within the block), are identified. Chaudhuri and Seth [9] 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.

On similar lines, Chan & Sivaswamy [10] and Joshi *et al.* [11] have used Gabor features for classification of most of the Indian script documents. They assume a block of text to consist of characters from the same script and employ a multi-channel log-Gabor filter bank to discriminate between the various scripts. Manthalkar and Biswas [12] have used rotation-invariant texture features, using Gabor filterbank, to differentiate between text blocks for various Indian scripts. Ablavsky and Stevens [13] make use of geometric properties of the text structures at connected component level to classify the script of lines of text. Gllavata and Freisleben [14] make use of a set of textural and structural features such as wavelet coefficients and horizontal projection profile, to separate the Roman script from Ideographic scripts.

Recognition of the script, using statistical features, at word the level has been reported by Dhanya *et al.* [15]. Here the authors differentiated Tamil from Roman script. This work has been extended by Pati *et al.* [16] and [17], for identification of Odiya and Hindi scripts, besides Tamil, against Roman. They use Gabor filters with linear discriminant classifier. Besides, redesign of the Gabor functions has enhanced the system efficiency. Ma & Doermann [18] use Gabor filterbank to separate Roman script from Arabic, Chinese, Devanagari or Korean. Jaeger *et al.* [19] have combined the informational confidence values of various classifiers to improve upon the above system. Both of these works deal only with bi-script documents.

Pal and Chaudhuri have performed script recognition at the word level [6,20] for two sets of triplets: Roman, Devanagari and Bengali/Telugu. Later, using teh same structural features, they identified 12 different Indian scripts. On similar lines, Padma & Nagabhushan [21] have discriminated Roman, Devanagari and Kannada scripts.

3 System Description

In the present work, we explore the effectiveness of our approach [17] in recognizing word-level change of script up to eleven scripts. We studied the structural properties of the scripts before designing an identifier for these scripts. Since, the scripts of Assamese and Bengali are nearly the same, we consider them as one script. An observation of these eleven Indian scripts reveals the following properties:

- Bengali, Devanagari and Punjabi scripts have a *shirorekha* joining the individual symbols forming any word.
- These three scripts have a lot of vertical strokes. Besides, the structural limbs of Devanagari are more circular than those of Bengali and Punjabi.
- There are many limbs along 60^0 and 120^0 directions in Bengali scripts.
- Punjabi script has a lot of half-length vertical strokes.
- Kannada and Telugu scripts are very similar. There is a tick feature associated with Telugu script, which does not appear in Kannada.
- Tamil and Malayalam are somewhat similar to each other. Malayalam appears like Tamil script with corners smoothed.
- Visually, Gujurati script looks like Devanagari but has more loops. In addition to high number of loops, Odiya script also has a lot of vertical strokes.
- Urdu looks quite unlike any other Indian script. It has a lot of connectivities and curvatures, aligned either along horizontal direction or at about 75^0

angle.

With these observations, we decided to employ textural features that are both frequency and direction sensitive. Gabor functions generate such filters as well as meet the equality criterion of the space-bandwidth limitation [22,23]. This, in our opinion, would be best able to discriminate between the scripts. In addition, we also used DCT features for comparison.

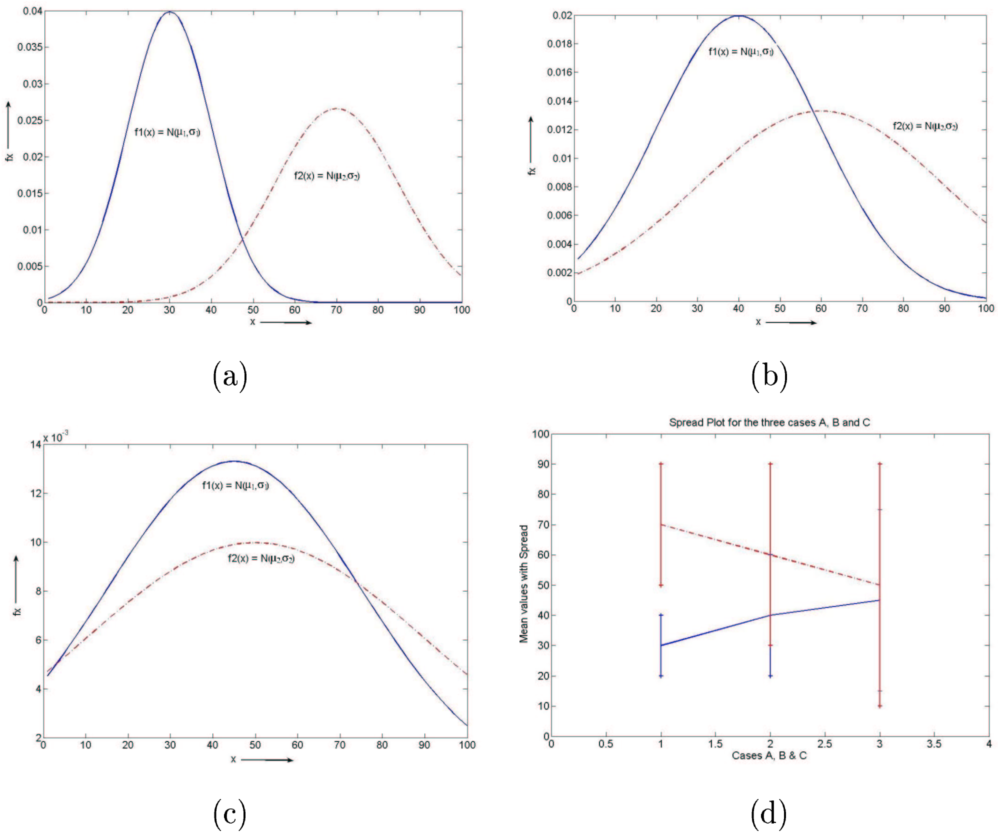
Two approaches can be persued for script identification in a multi-script scenario. One of them extensively studies the similarities and differences in the structures between the co-occurring scripts, while the second method deals with each script as a different texture. In our view, the latter method is more robust as it deals with the script regardless of the size or style of the font. This claim of ours is supported by our earlier employment of a bank of Gabor filters for successful page layout analysis [24,25] and script identification [15–17].

Thus, we employ a multi-channel filtering approach, using Gabor functions. We have used a radial frequency bandwidth of 1 octave. This is because, the optical channel of the HVS is studied to have a bandwidth of 1 octave. It is also observed that the coding of natural images is best attained by this bandwidth [26]. An angular bandwidth of 30° is also chosen for this experiment. This comes from the study of the properties of the scripts under consideration. We considered 5 different radial frequencies (0.0625, 0.125, 0.25, 0.5 and 1). We studied the efficacy of these frequencies for their separability with all bi-class problems involving Roman script and one of the Indian scripts. This was accomplished by observing: (i) the spread and (ii) the divergence values of the features.

Each feature from any class is assumed to follow a normal distribution pattern. Thus, we evaluate the mean and the standard deviation of each feature from each of the classes. When we plot the Gaussian curves for the features of the two classes, it generates a figure similar to figures 3(a), (b) or (c). If the means are close and the standard deviations large, the two classes are said to have a large overlap. With the mean values far apart and the standard deviations small, the feature is said to be discriminable. The spread plot for the features, with the distributions shown in figures 3(a), (b) and (c) is shown in Fig. 3(d). We generate this plot for all the features and observe which of them better discriminate than the others.

Divergence for a feature [27], in a bi-class scenario, is defined as:

$$d = \frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} - 2\right) + \frac{1}{2}(\mu_1 - \mu_2)^2\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right) \tag{1}$$



(a)

(b)

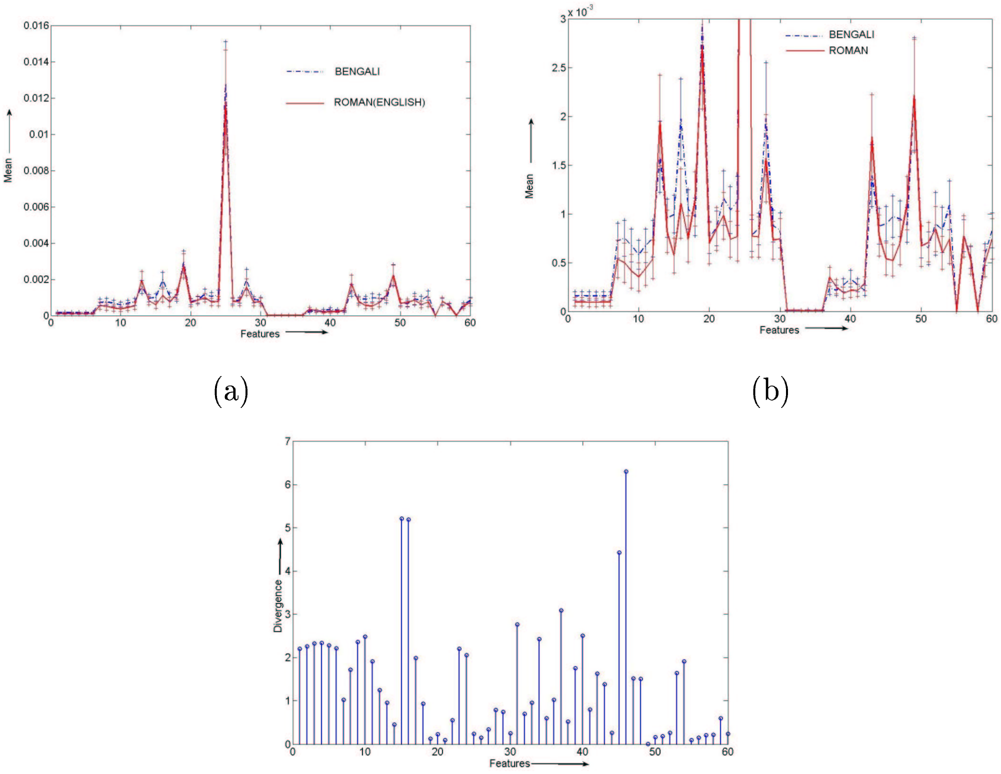
(c)

(d)

Fig. 3. Demonstration of the discriminability of features and their spread plot. (a) a discriminable feature. (b) a partially discriminable feature. (c) a non-discriminable feature. (d) the spread plots of the features shown in (a), (b) and (c).

where μ_1 and σ_1 are the mean and the standard deviation of the feature for class 1, assuming a normal distribution. $N(\mu_2, \sigma_2)$ represents the distribution of the same feature for class 2. Bigger the divergence value, the classes are better discriminated by the feature. We evaluate the divergence of each of the features extracted and decide on the nature of separability of the classes with respect to the features using the divergence plot.

Figure 4(a) shows a spread plot for the Gabor features generated by the filter bank, as mentioned above. Here, we use 5 radial frequencies (1, 0.5, 0.25, 0.125 and 0.0625) and six angles (0, 30, 60, 90, 120 and 150 degrees). Therefore, there are 30 cosine and 30 sine filters giving rise to a feature vector of dimension 60. The first 6 features correspond to the coefficients for cosine filters with radial frequency, $u = 0.0625$, and increasing angles in the order mentioned above. Thus, the feature number 1 comes from the filter with radial frequency 0.0625 and angle 0 degrees while the 10th feature corresponds to the filter with $u = 0.125$ and 90^0 angle. The 31^{st} to 60^{th} features are derived in the same order of filter parameters, but with sine filters. A magnified version of the spread plot in 4(a) is provided in 4(b) for better observability. The divergence plot for the same case is presented in 4(c). It is seen from figure 4(c) that the features



(a)

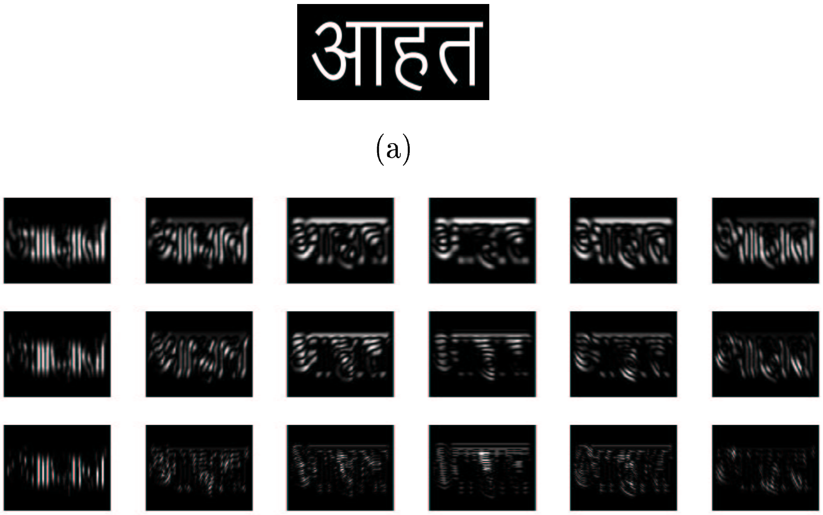
(b)

(c)

Fig. 4. Example case of Bengali and English bi-class discriminability analysis. (a) Spread plot using the 60-dimensional feature vectors from Bengali and English scripts, (b) a magnified version of the spread plot in (a), and (c) the divergence plot for the same case.

numbered 15, 16, 45 and 46 have considerably larger values than others which means these features are better discriminable. On a closer look at the spread plot shown in 4(b), it is these features which are clearly discriminable. Thus, it may be assumed that these features, would generate a good accuracy, for this bi-script cases of Bengali and English.

We observed the spread plot and divergence of all the bi-class cases involving separation of Roman script from each of other 10 scripts. Based on our observation, we decided to use three different radial frequencies (0.125, 0.25 & 0.5) and all the six angles of orientation. The spatial spread of each filter along the $x-$ and $y-$coordinates are determined by the standard deviations of the Gaussian's, 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. Figure 5 presents the output images for a sample Hindi word when filtered by the 18 co-sine filters. Each of these filtered images is sum-squared to evaluate the energy content of these images. This energy is normalized by the total energy of the input word image. Thus, the feature is



(a)

(b)

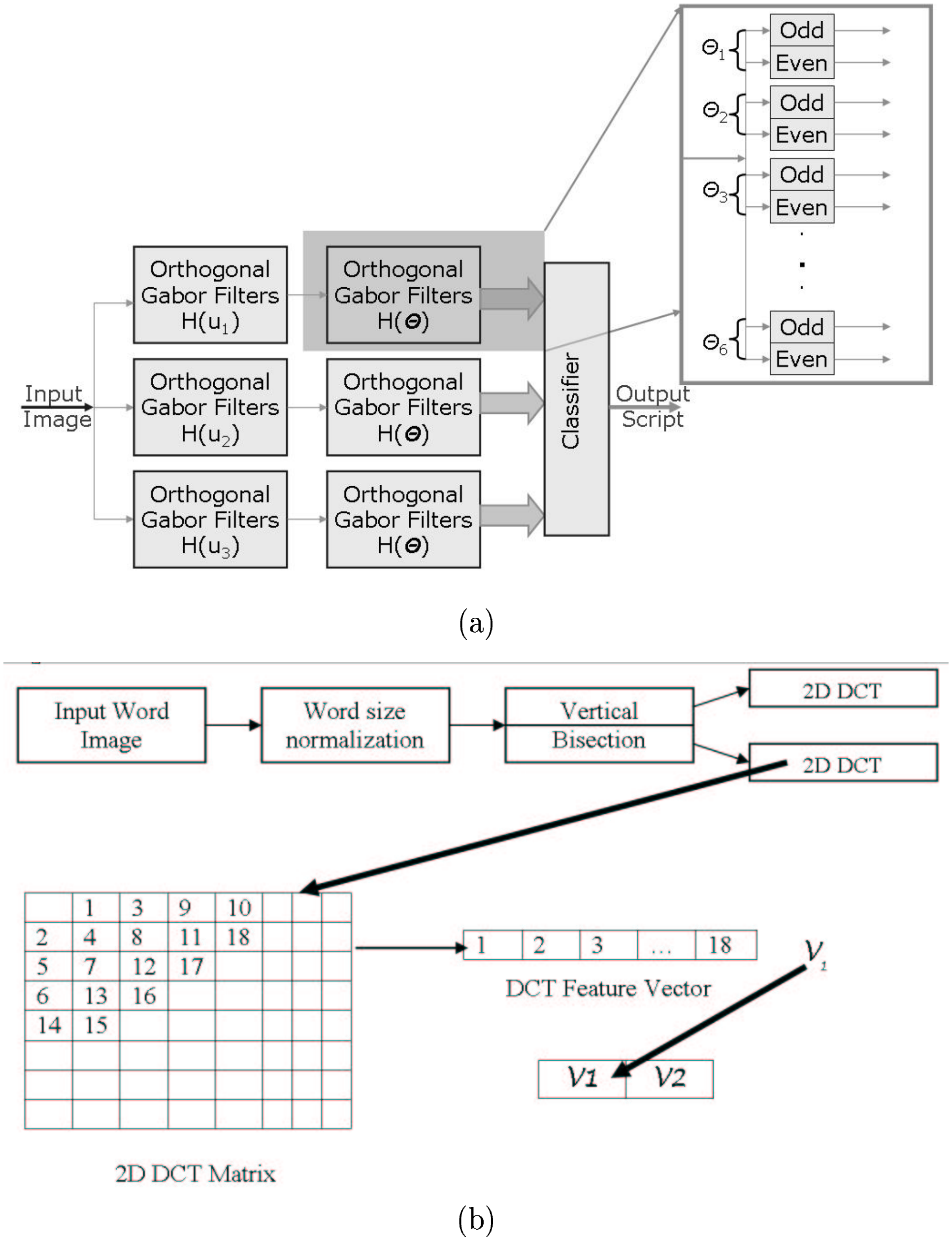
Fig. 5. (a) A sample Devanagari word, and (b) the 18 co-sine Gabor filtered output images. Here, each row corresponds to a radial frequency, in increasing order, and each column corresponds to an angle, in increasing order.

a ratio between the filtered image energy and the input image energy. Such a normalization makes the features independent of the size of the input image. We have a 36 dimensional feature vector, derived from 36 filters. Figure 6(a) shows a block diagram of the script identification system using the Gabor features.

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. But unlike KLT, DCT is not dependent on the data. Its transform matrix exhibits symmetries which can be exploited to obtain efficient hardware and software implementations [28]. Most image and video coders employ DCT. It has also been employed for other applications such as pitch modification for speech synthesis [29]. Pati [30] has used DCT coefficients for machine recognition of printed Odiya characters. Salazar and Trans [31] have reported better quality image resizing in the DCT domain. DCT has also been used for motion estimation [32], image compression with morphological descriptor [33] and object boundary detection [34].

For an image $I(i, j)$, the DCT coefficient matrix $B(k, l)$ is given by:

$$B(k,l) = \sum_{i=0}^{R-1} \sum_{j=0}^{C-1} I(i,j) \cos\left(\frac{\pi k(2i+1)}{2\,R}\right) \cos\left(\frac{\pi l(2j+1)}{2\,C}\right) \tag{2}$$



	1	3	9	10			
2	4	8	11	18			
5	7	12	17				
6	13	16					
14	15						

1	2	3	…	18

Fig. 6. (a) Block diagrams. (a) Gabor and (b) DCT feature extractors.

where R and C are the number of rows and columns of the image matrix; k and l are the frequency indices along the i and j directions, respectively.

The energy compactness property of DCT justifies its use for script identification. Figure 6(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 performed on each of the block, independently. As shown in Fig. 6(b), 18 low frequency coefficients are chosen in a zig-zag fashion from the 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). Nearest neighbor has been a standard and time tested classifier. This classifier has proven to deliver good output, when we have class representative training sets. Here, 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. The two sides of this plane represent 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. Amongst the discriminant approaches for classification, the most recent is the Support Vector Machine [35], where the optimal hyper-plane decides 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.

4 Data Description

Document images are scanned using: (i) Umax Astra 5400, and (ii) HP Scanjet 2200c scanners at 300 dpi resolution and stored in 8-bit gray format. The images are scanned from magazines, newspapers, books and laser printed documents. Variations in printing style and sizes are ensured. Eleven different scripts are considered for this database, namely, Bengali (Bangla), Roman (English), Devanagari (Hindi/Marathi), Gujurati, Kannada, Malayalam, Odiya, Gurumukhi (Punjabi), Tamil, Telugu and Urdu. About 100 pages are scanned from each of these scripts and are segmented into words by an automatic process [30].

Each segmented word is visually inspected to make sure that at least a single base character is present. The blank rows and columns at the beginning and end of each word are removed to make sure that the word touches the boundaries of the matrix representing it. Each word matrix is available to us in binary form, where ON-pixels form the structural limbs of a character/pattern in the word. The sizes of the words are not normalized, *i.e.*, the words are retained in their original sizes as they appeared in the document image. Words of various lengths and ON-pixel densities are retained. Figure 7 presents the distribution of the aspect ratios and the ON-pixel densities for the word image dataset. It may be noted from Fig. 7(a) that the words have a wide range of aspect ratios in all the scripts. Similarly, Fig. 7(b) shows that the ON-pixel densities of words across scripts also have a wide variability.

Once a large number of such words are collected, 2-D DCT is performed on the size normalized version of these words and feature vectors are formed. The Euclidean distance is evaluated between all the combinations of two words from the same script. If the distance between any two words becomes zero, only one of the two is retained in the database. This ensures that no word is repeated in the feature space. At the end, 20,000 words are randomly selected from this large collection to form our word data set. 7000 of these are selected randomly from each script, to form the training set, while the rest 13,000 words form the test set. This set is available for public use from the world wide web network [36].

5 Results and Discussion

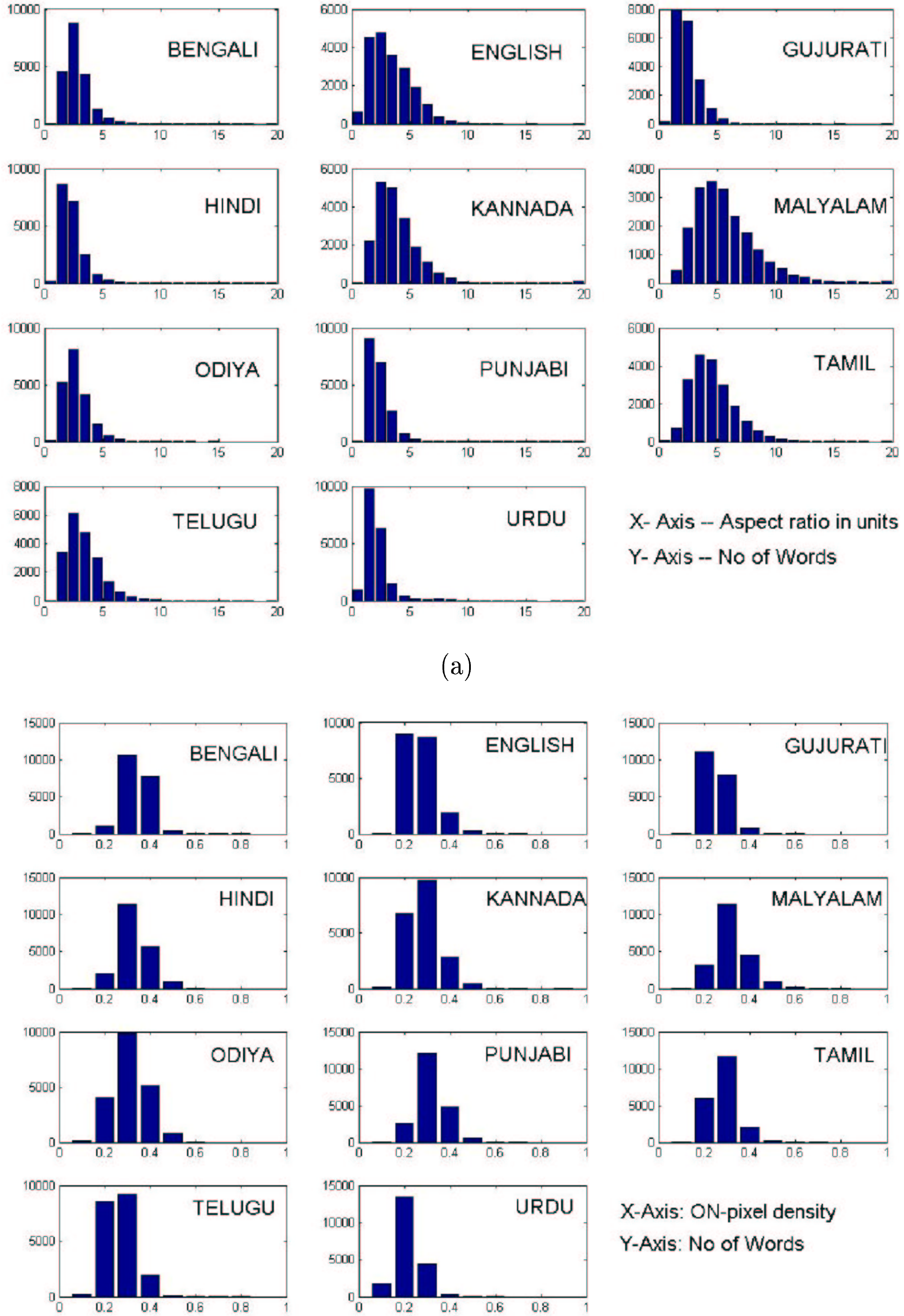
Most of the multi-lingual documents in India are bi-script in nature. So, all bi-script cases are handled first. Based on the encouraging results that we got in these experiments, we decided to extend the experiments to tri-script case as well. Since most of the official documents of national importance have three scripts, such an experiment is justified. Finally, we also explore the possibility of recognizing the script of a word without any prior information. This is a blind script recognizer, where the training set contains samples from all the classes. Thus, it is a 11-script scenario. In the sections below, we present each of these cases, separately.

In the following sections, the codes BE, EN, GU, HI, KA, MA, OD, PU, TA, TE and UR denote the scripts representing Bengali, English, Gujurati, Hindi, Kannada, Malayalam, Odiya, Punjabi, Tamil, Telugu and Urdu, respectively.

5.1 Bi-script Recognition

The bi-script documents, such as books, newspapers and magazines, have a local 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. We report the accuracies of the bi-class script identification in Tables 1, 2 and 3. The performance is presented in %age, which gives the average recognition accuracy for both the involved scripts.

Table 1 presents the results for all bi-script combinations with nearest neighbor



(a)

(b)

Fig. 7. Characteristics of the word database. (a) The distribution of the aspect ratio. (b) ON-pixel density.

Table 1
Bi-Script recognition accuracies with NNC. The lower triangle part of the matrix gives the results with Gabor features & the upper triangle, the DCT. The results presented are average bi-script accuracies in %. The script names are presented in their abbreviated forms as mentioned in section 5.

	BE	EN	GU	HI	KA	MA	OD	PU	TA	TE	UR
BE	–	**99.6**	99.1	95.1	99.2	98.9	99.0	96.9	98.9	99.0	99.5
EN	99.8	–	99.3	**99.6**	98.2	97.9	99.1	99.4	97.7	97.5	99.4
GU	99.3	99.7	–	99.4	99.4	98.9	98.4	99.3	98.3	98.4	98.2
HI	95.2	99.6	99.2	–	99.3	99.0	99.2	94.0	99.1	99.4	**99.6**
KA	99.3	**99.9**	99.3	99.5	–	96.8	98.9	99.3	98.9	*93.4*	99.3
MA	98.7	99.1	99.1	98.7	98.5	–	97.7	98.8	93.8	95.9	99.1
OD	98.5	99.2	96.1	99.1	97.6	93.4	–	98.3	97.8	98.3	99.0
PU	98.2	99.2	99.3	91.5	99.5	96.6	98.6	–	98.9	99.2	**99.6**
TA	98.5	99.4	99.5	99.0	99.5	96.1	97.4	97.9	–	97.6	98.3
TE	99.0	99.8	99.3	99.6	*89.2*	98.9	97.8	99.6	99.5	–	97.5
UR	99.6	99.7	99.2	99.6	98.1	99.2	98.8	99.6	99.2	98.7	–

classifier. In this table, the upper triangular part of the matrix presents the results with the DCT based feature vector while the lower triangular part shows the Gabor feature results. Similarly, Tables 2 and 3 present the results with linear discriminant classifier and support vector machines, respectively.

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. In fact, in Table 2, 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 (see Table 3), 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. Similarly, a comparison of Tables 1, 2 and 3, shows that SVM and NNC have performed comparably and with consistency. Combinations of LDC, for both features, yield the least.

All the classifiers have performed very well with Gabor feature vectors, though in most cases the NNC performs marginally better. The highest accuracy obtained is 99.9% for the bi-class problem of English vs. Kannada, while the lowest is 89.2% for the Kannada – Telugu combination, both with NNC. The

Table 2
Bi-Script recognition accuracies with linear discriminant classifier. The lower triangular part of the matrix shows the results with Gabor features while upper triangle, those using DCT. The results presented are average bi-script accuracies in %. The script are denoted by their abbreviated codes (see section 5).

	BE	EN	GU	HI	KA	MA	OD	PU	TA	TE	UR
BE	–	**96.6**	92.3	80.3	91.9	88.4	92.1	87.6	94.0	89.0	95.1
EN	99.4	–	88.8	95.3	90.3	87.7	89.4	91.3	88.3	87.0	91.4
GU	99.3	98.5	–	91.9	90.7	85.7	88.6	90.3	85.1	86.9	87.5
HI	94.6	98.7	99.3	–	91.2	88.6	90.1	78.9	92.4	90.1	93.9
KA	98.6	99.0	98.6	98.6	–	84.8	90.0	90.5	91.9	*74.8*	91.5
MA	98.0	98.3	98.4	98.3	98.8	–	80.7	88.0	83.4	76.9	88.6
OD	98.8	98.8	93.5	99.2	98.0	97.4	–	87.1	88.4	85.5	91.9
PU	99.0	98.8	99.3	93.2	99.6	97.7	99.0	–	89.4	88.7	92.4
TA	98.7	98.0	98.7	98.2	99.1	96.0	97.7	97.8	–	86.4	83.9
TE	99.2	99.5	98.8	99.2	*92.9*	99.3	98.4	**99.7**	99.6	–	85.9
UR	99.3	99.4	98.2	99.4	97.7	99.3	98.6	**99.7**	99.3	99.0	–

high misclassification between Kannada and Telugu scripts, is due to their structural similarity.

The most frequent scenario that warrants script recognition involves the official documents by State Governments and the text books in state languages. These documents contain the script of the official language of the state and English. Table 4 shows the results of these bi-script scenarios for various feature-classifier combinations. Here, GT is Gabor transform and, hence, the code GT-NNC stands for the combination of Gabor features with the NNC. It may be observed from this table that the Gabor-NNC combination is leading with more than 99% for all such bi-script cases. The lowest performance is for Malayalam, which is 99.1%, whereas the best performance is for Kannada, which is 99.9%.

5.2 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 of three scripts as the triplet of the state. Figure 1(b) presents a train reservation form containing Kannada, Devanagari and Roman scripts. In the second series

Table 3
Bi-Script recognition accuracies with support vector machines. The values in the lower triangle part of the matrix are the results using Gabor features; in upper triangle due to DCT. The results presented are average bi-script accuracies in %. The script codes presented denote the scripts mentioned in section 5.

	BE	EN	GU	HI	KA	MA	OD	PU	TA	TE	UR
BE	–	**99.5**	98.1	95.6	99.1	98.7	98.1	97.2	98.0	98.8	**99.5**
EN	**99.8**	–	98.9	98.8	97.5	97.1	99.2	96.4	97.2	98.3	99.0
GU	99.1	98.7	–	99.2	98.7	98.8	97.9	98.5	98.4	98.6	96.1
HI	95.7	**99.8**	98.9	–	98.6	97.8	99.2	95.2	97.6	98.6	97.2
KA	99.0	99.7	99.0	99.5	–	96.8	97.9	98.2	99.2	94.1	97.8
MA	98.7	99.1	98.2	98.9	99.2	–	96.2	98.2	*93.9*	94.6	98.0
OD	**99.8**	99.3	98.5	99.7	98.2	97.8	–	97.4	98.3	96.9	98.5
PU	98.5	99.2	98.7	94.2	99.5	97.4	98.2	–	99.1	98.6	99.0
TA	98.9	95.7	98.8	99.2	99.6	96.9	98.6	98.7	–	97.1	96.4
TE	99.2	**99.8**	97.9	99.5	*91.0*	97.4	97.2	99.1	98.6	–	97.0
UR	99.7	**99.8**	99.3	**99.8**	98.7	99.6	99.6	93.7	96.2	98.5	–

Table 4
The recognition accuracies of the various feature-classifier combinations for the bi-script cases involving English words with one of the ten Indian scripts. Codes denoting the Indian scripts are described in Sec. 5. The last two columns present the mean (μ) and standard deviation (σ) for the bi-script results shown in the same row.

	BE	GU	HI	KA	MA	OD	PU	TA	TE	UR	μ	σ
GT-NNC	99.8	99.7	99.6	99.9	99.1	99.2	99.2	99.4	99.8	99.7	99.6	0.3
GT-LDC	99.4	98.5	98.7	99.0	98.3	98.8	98.8	98.0	99.5	99.4	98.8	0.5
GT-SVM	99.8	98.7	99.8	99.7	99.1	99.3	99.2	95.7	99.8	99.8	99.1	1.3
DCT-NNC	99.6	99.3	99.6	98.2	97.9	99.1	99.4	97.7	97.5	99.4	98.8	0.8
DCT-LDC	96.6	88.8	95.3	90.3	87.7	89.4	91.3	88.3	87.0	91.4	90.6	3.2
DCT-SVM	99.5	98.9	98.8	97.5	97.1	99.2	96.4	97.2	98.3	99.0	98.2	1.1

of experiments, we have discriminated amongst such triplets. Here, Devanagari & Roman scripts are common to all the triplets, while the other Indian script varies. The results of such experiments are presented in Table 5. In this table, the μ & σ columns report the average & standard deviation of the recognition rates for all the nine triplets presented in the same row.

Table 5
Tri-Script recognition accuracies (common scripts in all experiments are Devanagari and Roman).

		BE	GU	KA	MA	OD	PU	TA	TE	UR	μ	σ
GABOR	NNC	96.5	99.0	99.4	98.4	98.8	93.8	98.8	99.4	99.4	98.2	1.9
	LDC	94.5	97.5	97.9	96.9	98.0	93.4	97.3	98.3	98.3	96.9	1.8
	SVM	97.0	99.2	99.6	98.6	99.0	92.9	98.8	99.2	99.7	98.2	2.1
DCT	NNC	96.5	99.1	98.3	97.9	98.7	95.6	97.8	97.9	99.2	97.9	1.2
	LDC	82.2	86.1	85.5	82.4	83.6	77.6	85.5	83.0	87.2	83.7	2.9
	SVM	97.1	99.5	98.7	97.6	99.0	96.3	98.0	98.2	99.1	98.1	1.0

Table 5 shows that the maximum average accuracy obtained is 98.2%. This result occurs for Gabor features with NN and SVM classifiers. However, the Gabor-NNC combination is more consistent and results in lower standard deviation ($\sigma_{NNC} = 1.9$ while $\sigma_{SVM} = 2.1$). However, given a prior knowledge of the triplet, one may consider using a different feature-classifier combination to produce a triplet-specific optimal result.

We have compared our results with the earlier reported results. Pal & Chaudhuri [20] have reported an accuracy of 97.2% for recognition of Devanagari words from the triplet of Telugu. Padma & Nagabhushan [21] have reported 97% for the same from the triplet involving Kannada script. We achieve an average of 99.6% for the recognition involving these triplets, with a larger test set.

5.3 Multi-script Recognition

Since our identification scheme depends on statistical rather than structural features, we have an advantage that we could consider any number of scripts together and identify them. On the observation that our feature – classifier combinations are delivering us very good recognition accuracy, we tried to identify the scripts in a multi-script scenario, involving all the eleven earlier mentioned Indian scripts. Here, every test sample is compared with the reference samples from all the classes. The result of correct classification of each of the eleven scripts for various feature-classifier combinations is reported in Table 6. In this table, the last two rows show the average classification accuracy and its standard deviation, respectively, for the corresponding feature-classifier combination. The results reported in this table clearly show that SVM is leading with both the features. On the feature front, the Gabor is marginally leading. Thus, it may be inferred that for a blind multi-script classification

Table 6
The recognition accuracies of the various feature-classifier combinations for the eleven-script scenario. The correct classification accuracy for each of the eleven Indian scripts is reported. The last two rows present the mean (μ) and standard deviation (σ) for the results in the same column.

	Gabor			DCT		
	NNC	LDC	SVM	NNC	LDC	SVM
BE	92.1	89.4	96.2	92.8	51.3	92.6
EN	97.8	95.2	98.2	97.7	68.9	96.6
GU	90.2	85.7	95.5	94.7	62.2	95.4
HI	86.6	66.0	93.3	90.6	49.8	94.3
KA	88.7	71.5	93.3	89.6	70.6	90.3
MA	88.2	82.4	93.6	90.0	23.5	84.4
OD	84.5	77.6	94.0	94.2	32.6	94.3
PU	86.2	84.3	93.8	89.2	56.2	92.1
TA	91.5	85.9	95.2	87.7	56.7	93.3
TE	83.8	82.8	92.3	82.2	15.4	91.0
UR	94.5	93.6	97.9	93.0	61.0	98.4
μ	89.4	83.1	94.8	91.1	49.8	93.8
σ	4.4	8.8	1.9	4.1	18.3	3.9

scenario, Gabor features when combined with SVM as the classifier, result in the optimal output.

Using Gabor feature with NNC, the worst performance of 83.8% is for Telugu, and the best performance of 97.8% is for English. A similar trend is observed for SVM, where the minimum of 92.3% and the maximum of 98.2% are obtained for the Telugu and Roman scripts, respectively. Thus, in a multi-script scenario, with little a priori information, it makes sense to separate the English words first, and then deal with the remaining scripts in a hierarchical manner. The least performance for Telugu script is to be expected, since the script is very similar to Kannada. When the confusion matrix of this experiment is analyzed, it is observed that the maximum misclassification of Telugu words has been to Kannada, which is 12.7%. Similarly, 7.1% of the Kannada words are misclassified as Telugu. The next such pair is Hindi – Punjabi, where 7.3% of words from Hindi go to Punjabi and 7.7%, the reverse way. Notwithstanding all these observations, all the scripts are fairly well spread out in the Gabor feature space, since with 11 different scripts, we are still able to achieve an average script recognition accuracy of 94.8%.

Use of DCT features with NNC, improves the recognition of Hindi, Odiya and Punjabi scripts by 4.0%, 9.7% and 3.0%, respectively. Similarly, when DCT is used in combination with SVM, there is improvement for Hindi, Odiya and Urdu scripts. Thus, DCT feature is able to evaluate Hindi and Odiya better than its Gabor counterpart.

6 Conclusion

The combination of Gabor filter bank with either SVM or NN classifier handles the important issue of script recognition at the word level quite well. For most cases, NNC performs at par with SVM and they both outperform LDC. However, the actual performance is script dependent. For example, using the Gabor-NNC combination, the overall classification performance for the tri-script combination involving Kannada, Devanagari and English is 99.6%, whereas the average correct recognition is only 89.2% for the bi-script combination of Kannada with Telugu, and it is only 91.5% when Punjabi is recognized against Hindi. English script has a recognition accuracy of 97.8%, using NNC, against all the eleven scripts which is the best in the eleven-class scenario.

When Gabor and DCT features are compared, Gabor seems to be leading in most of the bi-script and tri-script cases. However, in the 11-script scenario, using DCT improves the recognition accuracies of Hindi, Odiya and Punjabi scripts. In bi-script experiments, the average recognition accuracy of Kannada and Telugu duo improves to 93.4% using DCT features from a value of 89.2% using Gabor. Interestingly, with Gabor features, when LDC is used for the above case, the recognition accuracy increases to 92.9% from 89.2% with NNC. However, NNC performs better (93.4%) with DCT, than LDC which has a recognition accuracy of 74.8%. Thus, it is immediately not very clear whether it is the feature or the classifier or the specific combination that is responsible for the peculiar behaviour observed in this case. We are also planning to investigate if the length of a word has any role to play in the recognition accuracy achieved.

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.

The size of 7000 training patterns per script is fairly large. This consumes a lot of time while NNC is at work. Moreover, LDC and SVM require a selected few boundary patterns per class. Thus it is worth trying some of the prototype selection techniques to reduce the training set.

An evaluation of the divergence of each of the features, shows that some features have better discriminating property than others. Thus a selected set of features might help in reducing the computation, while enhancing the efficacy of the system. Such a set could be uniquely selected for each of the cases. Different sets could exist for different bi-class or multi-class cases.

References

[1] S. N. S. Rajasekaran and B. L. Deekshatulu, "Recogntion of printed Telugu characters," *Computer Graphics and Image Processing*, vol. 6, pp. 335–360, 1977.

[2] P. B. Pati and A. G. Ramakrishnan, "OCR in Indian Scripts: A Survey," *IETE Technical Review*, vol. 22, no. 3, pp. 217–227, 2000.

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

[4] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic script identification from document images using cluster based templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 176–181, 1997.

[5] S. L. Wood, X. Yao, K. Krishnamurthi, and L. Dang, "Language identification for printed text independent of segmentation," in *Proc. Intl. Conf. on Image Processing*, pp. 428–431, 1995.

[6] A. R. Chaudhuri, A. K. Mandal, and B. B. Chaudhuri, "Page layout analyser for multilingual Indian documents," in *Proc. the Language Engineering Conference*, pp. 24–32, 2002.

[7] U. Pal and B. B. Chaudhuri, "Script line separation from Indian muliscript document," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 406–409, 1999.

[8] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 751–756, 1998.

[9] S. Chaudhuri and R. Seth, "Trainable Script Identification Strategies for Indian languages," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 657–660, 1999.

[10] W. Chan and J. Sivaswamy, "Local energy analysis for text script classification," in *Proc. of Image and Vision Computing New Zealand (1999)*, (New Zealand), 1999.

[11] G. D. Joshi, S. Garg, and J. Sivaswamy, "Script identification from indian documents," in *Seventh IAPR Workshop on Document Analysis Systems 2006, LNCS-3872*, (New Zealand), pp. 255–267, February 2000.

[12] R. Manthalkar and P. K. Biswas, "An automatic script identification scheme for Indian Languages," in *Proc. Eighth Nat. Conf. on Comm., 2002*, (Mumbai, INDIA), pp. 31–34, 2002.

[13] V. Ablavsky and M. R. Stevens, "Automatic feature selection with applications to script identification of degraded documents," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 750–754, 2003.

[14] J. Gllavata and B. Freisleben, "Script recognition in images with complex backgrounds," in *Proc. of the Fifth IEEE Intl. Sym. on Sig. Proc. & Info. Tech., 2005*, pp. 589–594, 2005.

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

[16] P. B. Pati, S. S. Raju, N. K. Pati, and A. G. Ramakrishnan, "Gabor filters for document analysis in Indian bilingual documents," in *Proc. Int. Conf. on Intelligent Sensing and Information Processing*, pp. 123–126, 2004.

[17] P. B. Pati and A. G. Ramakrishnan, "HVS inspired system for script identification in indian multi-script documents," in *Seventh IAPR Workshop on Document Analysis Systems 2006, LNCS-3872*, (New Zealand), pp. 380–389, February 2006.

[18] H. Ma and D. Doermann, "Gabor filter based multi-class classifier for scanned document images," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 968–972, 2003.

[19] S. Jaeger, H. Ma, and D. Doermann, "Identifying Script onword-Level with Informational Confidence," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 416–420, 2005.

[20] U. Pal, S. Sinha, and B. B. Chaudhury, "Word-wise script identification from a document containing English, Devanagari and Telugu text," in *Proc. National Conf. on Document Analysis and Recognition*, pp. 213–220, 2003.

[21] M. C. Padma and P. Nagabhushana, "Identification and separation of text words of Kannada, Hindi and English languages through discriminating features," in *Proc. National Conf. on Document Analysis and Recognition*, pp. 252–260, 2003.

[22] D. Gabor, "Theory of communication," *J. IEE (London)*, vol. 93, pp. 429–457, 1946.

[23] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[24] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan, "Gabor filter based block energy analysis for text extraction from digital document images," in *Proc. First Int. Workshop on Document Image Analysis for Libraries (DIAL'04)*, pp. 233–243, 2004.

[25] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan, "Text localization and extraction from complex color images," in *Proc. Int. Sym. on Visual Computing, LNCS - 3804*, pp. 486–493, 2005.

[26] D. J. Field, "Relation between the statistics of natural images and the response properties of cortical cells," *Journal of Opt. Soc. of Am. A*, vol. 4, no. 12, pp. 2379–2394, 1987.

[27] S. Theodoridis and K. Koutroumbas, *Pattern Recognition.* London: Academic Press, 1999.

[28] K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithms, Advantages, Applications.* New York: Academic Press, 1990.

[29] R. Muralishankar, A. G. Ramakrishnan, and P. Prathibha, "Modification of pitch using DCT in the source domain," *Speech Communication*, vol. 42, pp. 143–154, 2004.

[30] P. B. Pati, "Machine recognition of printed Odiya text documents," Master's thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, 2001.

[31] C. L. Salazar and T. D. Tran, "On resizing images in the DCT domain," in *Proc. Int. Conf. on Image Processing*, pp. 2797–2800, 2004.

[32] U.-V. Koc and K. J. R. Liu, "DCT based motion estimation," *IEEE Tr. on Image Processing*, vol. 7, no. 7, pp. 948–965, 1998.

[33] D. Zhao, W. Gao, and Y. K. Chan, "Morphological representation of DCT coefficients for image compression," *IEEE Tr. CSVT*, vol. 12, no. 9, pp. 819–823, 2002.

[34] J. Tang and S. T. Acton, "A DCT based gradient vector flow snake for object boundary detection," in *Southwest04*, pp. 157–161, 2004.

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

[36] P. B. Pati and A. G. Ramakrishnan, "Indian Script Word Image Dataset," www.ee.iisc.ernet.in/new/people/students/phd/pati/, 2005.