Script Identification from Indian Documents

Gopal Datt Joshi, Saurabh Garg, and Jayanthi Sivaswamy

Centre for Visual Information Technology,
IIIT Hyderabad, India
gopal@research.iiit.ac.in, jsivaswamy@iiit.ac.in

Abstract. Automatic identification of a script in a given document image facilitates many important applications such as automatic archiving of multilingual documents, searching online archives of document images and for the selection of script specific OCR in a multilingual environment. In this paper, we present a scheme to identify different Indian scripts from a document image. This scheme employs hierarchical classification which uses features consistent with human perception. Such features are extracted from the responses of a multi-channel log-Gabor filter bank, designed at an optimal scale and multiple orientations. In the first stage, the classifier groups the scripts into five major classes using global features. At the next stage, a sub-classification is performed based on script-specific features. All features are extracted globally from a given text block which does not require any complex and reliable segmentation of the document image into lines and characters. Thus the proposed scheme is efficient and can be used for many practical applications which require processing large volumes of data. The scheme has been tested on 10 Indian scripts and found to be robust to skew generated in the process of scanning and relatively insensitive to change in font size. This proposed system achieves an overall classification accuracy of 97.11% on a large testing data set. These results serve to establish the utility of global approach to classification of scripts.

1 Introduction

The amount of multimedia data captured and stored is increasing rapidly with the advances in computer technology. Such data include multi-lingual documents. For example, museums store images of all old fragile documents having scientific or historical or artistic value and written in different scripts which are stored in typically large databases. Document analysis systems that help process these stored images is of interest for both efficient archival and to provide access to various researchers. Script identification is a key step that arises in document image analysis especially when the environment is multi-script and multi-lingual. An automatic script identification scheme is useful to (i) sort document images, (ii) help in selecting appropriate script-specific OCRs and (iii) search online archives of document image for those containing a particular script.

Existing script classification approaches can be classified into two broad categories, namely, local and global approaches. The local approaches analyse a list

H. Bunke and A.L. Spitz (Eds.): DAS 2006, LNCS 3872, pp. 255–267, 2006.

of connected components (like line, word and character) in the document images to identify the script (or class of script) in the document image. However, these components are available only after line, word and character (LWC) segmentation of the underlying document image. In contrast, global approaches employ analysis of regions comprising at least two lines and hence do not require fine segmentation. Consequently, the script classification task is simplified and performed faster with the global rather than the local approach. This is attractive feature for a fast script-based retrieval systems.

In the category of local approaches, Spitz [1] proposed a method for discriminating Han based (Asian) and Latin based (includes both European and non-European) scripts. This method uses the vertical distribution of upward concavity in the characters of both the scripts. Furthermore, method uses optical density distribution in character and characteristic word's shape for further discrimination among Han and Latin scripts, respectively. Hochberg [2] proposed a script classification scheme which exploits frequently occurring character shapes (textual symbols) in each script. All textual symbols are rescaled to a fixed size (30×30) following which representative templates for each script are created by clustering textual symbols from a training data. Textual symbols from a new document are compared to the representative templates of all available scripts to find the best matched script. In India, a multi-lingual multi-script country, languages have scripts of their own, though some scripts like Devanagari, Bengali may be shared by two or more languages. Some classification methods have been proposed for Indian language scripts as well[6, 5]. These use Gabor features extracted from connected components [6] or statistical and topological features [5]. In [6], a connected component is processed only if its height is greater than three-fourth or less than the one-fourth of the average height of characters in document image. The training data is formed by representing each connected component with a feature vector (12 Gabor feature values) and a script label. This scheme has been shown to classify 4 major Indian language scripts (Devanagari, Roman (English), Telugu and Malayalam). A tree based classification scheme for twelve Indian language scripts in [5] uses horizontal profiles, statistical, topological and stroke based features. These features are chosen at a non-terminal node to get optimum tree classifier. These features, however, are very fragile in presence of noise.

In all the above approaches, the success of classification is dependent on the accuracy of character segmentation or connected component analysis. The problem of character segmentation presents a paradox similar to that presented by OCR, namely that character segmentation is best performed when the script of the document is known [4]. Some scripts, such as Chinese, have characters laid out in a regular array which greatly helps in character segmentation. Arabic scripts in contrast, are more difficult to segment due to the overlapping and conjoining of cursive characters during the typesetting process. On the other hand, Indian languages have a mixture of attributes in their scripts which help to segment at word level easily but not at the character level. As a result, one segmentation method does not work well for all the scripts. Due to this limi-

tation, local approaches are slower, computationally expensive and have to be developed with attention to a specific class of scripts.

Global approaches, in contrast, are designed to identify the script by analysing blocks of text extracted from the document image. Wood [3] proposes methods using Hough transforms and analysis of density profile resulting from projection along text lines. However, it is not clear how the projection profile can be analysed automatically to determine the script. A texture based classification scheme in [4] uses the rotationally invariant features from Gabor filter responses. Since the texture images formed by different scripts patterns are found to be consistent, the text blocks are normalized to have equal height and width with uniform spaces between the lines and the words. Many steps are used to make the script texture consistent, as a result of which the scheme is computationally expensive. Chan et al [8, 7] take a biologically inspired approach to text script classification and derive a set of descriptors from oriented local energy and demonstrate their utility in script classification. Testing on a standard or large size data set however, has not been reported.

Global approaches have practical importance in script based retrieval systems because they are relatively fast and reduce the cost of document handling. However, the shortcomings of existing global methods are poorer classification accuracy compared to the local approaches and inability to handle large classes of scripts. We propose a Gabor energy based classification scheme which uses a global approach and demonstrate its ability to classify 10 Indian language scripts. In section (2), we describe the proposed scheme in detail. Results of the scheme tested over a large data set are presented in section (3).

2 The Proposed Scheme

The proposed scheme is inspired from the observation that humans are capable of distinguishing between unfamiliar scripts just based on a simple visual inspection. Examination of the type of processing carried out at the pre-attentive (image data driven processing) level of the human visual system reveals the presence of cells which extract oriented line features. These cells have been shown to be modelled by Gabor functions [10]. With this as a starting point, we consider script identification as a process of texture analysis and classification similar to [4]. However, our analysis treats the line textures as deterministic features.

In general, a texture is a complex visual pattern composed of sub patterns or Textons. The subpatterns give rise to the perceived lightness, uniformity, density, roughness, regularity, linearity, frequency, phase, directionality, coarseness, randomness, fineness, smoothness, granulation etc; as the texture as a whole. Although subpatterns can lack a good mathematical model, it is well established that a texture can be analysed completely if and only if its subpatterns are well defined. For example, consider the images in Fig.1(in first row) showing the cross section of a basket, a pile of seeds and a synthetic image formed by several 'T's. Each of these texture images has its own subpattern such as different size rectangles in the basket texture, different size

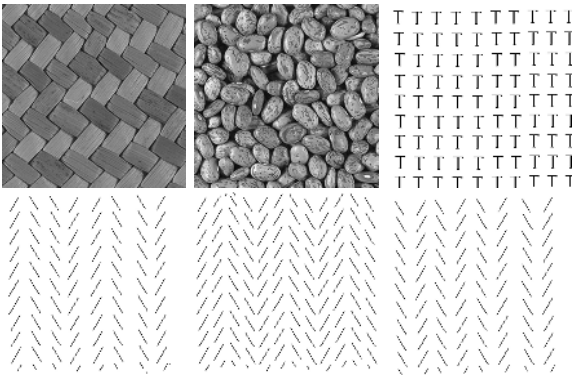

Fig. 1. Sample natural and synthetic textures

ovals in the seed texture and 'T' shape patterns in synthetic texture. In addition to the nature of the subpatterns, the manner in which they are organised can also affect the *look* of the textures. This is seen from the images in the second row of Fig.1. All the synthetic images in this row have the same subpattern. However, from a quick glance, it is seen that the one in the middle looks different from the other two, due to the compactness in the placing of subpatterns. But a more carefully look reveals that the first and third image are actually different. Despite the two images having the same compactness, their perceptions are different due to a rearrangement in the subpatterns. The perception after a quick glance is the result of a global(or coarse) analysis of the three images while the second perception after a more careful observation is a result of a local(or finer) analysis of the images. Script patterns can be considered to be textures formed by oriented linear subpatterns as the curved components are also decomposable into several oriented linear subpatterns. We argue that any script (not a language) can be characterised by the distribution of linear subpatterns across different orientations, the information about which can be obtained by a global analysis of the script image. For example, Chinese scripts are very compact and contain predominantly linear features. In contrast, many Indian scripts are composed of mostly curved features while Roman (English) scripts contain a good mixture of linear and curved features. These scripts can be easily classified using global analysis whereas, local analysis can be reserved for tasks such as distinguishing between (i) two different languages, such as English and French, written in one script or (ii) two similar scripts such as Urdu and Arabic. It is useful to study the extent of classification possible and the accuracy that is attainable using only global features. We use Indian language scripts as a test bed to perform this study.

2.1 Indian Language Scripts

India has 18 official languages which includes Assamese, Bangla, English, Gujarati, Hindi, Konkanai, Kannada, Kashmiri, Malayalam, Marathi, Nepali, Oriya,

Fig. 2. Indian language scripts

Punjabi, Rajasthani, Sanakrit, Tamil, Telugu and Urdu. All the Indian languages do not have the unique scripts. Some of them use the same script. For example, languages such as Hindi, Marathi, Rajasthani, Sanskrit and Nepali are written using the Devanagari script; Assamese and Bangla languages are written using the Bangla script; Urdu and Kashmiri are written using the same script and Telugu and Kannada use the same script. In all, ten different scripts are used to write these 18 languages. These scripts are named as Bangla, Devanagari, Roman(English), Gurumukhi, Gujarati, Malayalam, Oriya, Tamil, Kannada and Urdu. The image blocks of these images are shown in Fig. 2.

Some Indian scripts, like Devanagari, Bangla, Gurumukhi and Assamese have some common properties. Most of the characters have a horizontal lines at the upper part called *headline* and primarily the characters of words in these scripts are connected by a these headlines (shown in Fig. 4). Due to these properties they can be differentiable from the Roman (English), Telegu, Oriya, Urdu and other scripts. Furthermore, some characters have a part extended above the headline in these scripts. Presence of this portion is also useful for script classification. It would be advantageous to capture these distinguishing features using global level processing, if possible, as they can be used in script classification. We next propose a method to extract features through a global analysis of a given text document image and use them to classify the underlying script.

2.2 Preprocessing

Our scheme first segments the text area from the document image by removing the upper, lower, left and right blank regions. After this stage, we have an image which has textual and non-textual regions. This is then binarised after removing the graphics and pictures (at present the removal of non-textual information is performed manually, though page segmentation algorithms such as [12] could be readily been employed to perform this automatically). Text blocks of predefined size (100×200 pixels) are next extracted. It should be noted that the text block may contain lines with different font sizes and variable spaces between lines,

words and characters. Numerals may appear in the text. We do not perform any processing to homogenise these parameters. It is necessary only to ensure that at least 40% of the text block region contains text.

2.3 Feature Extraction

Log-Gabor Filtering. A traditional choice for texture analysis is to use Gabor filters at multiple scales. Based on our observations of the HVS performance, we have selected oriented local energy features for script classification, with the local energy defined to be the sum of squared responses of a pair of conjugate symmetric filters. One of the major advantage with these features is that it is not necessary to perform analysis at multiple scales which reduces the computational cost of the scheme. Hence, features can be obtained from the image by using a single, empirically determined optimal scale. The optimal scale is one in which filters respond maximally to the given input. This response can be further enhanced by increasing filter bandwidth at the same optimal scale. The maximum bandwidth obtainable from a Gabor filter is only about 1 octave which is a disadvantage as it limits the feature size that can be captured. A log-Gabor filter on the other hand, allows large bandwidths from 1 to 3 octaves which makes the features more effective, reliable and informative [15]. In our scheme, features are extracted using a log-Gabor filter bank designed at a single optimal scale but at different orientations.

Due to the singularity in the log-Gabor function at the origin, one cannot construct an analytic expression for the shape of log-Gabor function in the spatial domain. Hence, one has to design the filter in the frequency domain. On a linear scale, the transfer function of a log-Gabor filter is expressed as

$$\Phi_{(r_o,\theta_o)} = \exp\left\{-\frac{(\log(\frac{r}{r_o}))^2}{2(\log(\frac{\sigma_r}{r_o}))^2}\right\}\exp\left\{-\frac{(\theta-\theta_o)^2}{2\sigma_\theta^2}\right\} \tag{1}$$

where r_o is the central radial frequency, θ_o is the orientation of the filter, σ_θ and σ_r represent the angular and radial bandwidths, respectively.

The oriented local energy $E^{r_o}_{\theta_o}(x,y)$ at every point in the image defines an energy map. This is obtained as:

$$E^{r_o}_{\theta_o}(x,y) = \sqrt{(O^{r_o,even}_{\theta_o}(x,y))^2 + (O^{r_o,odd}_{\theta_o}(x,y))^2} \tag{2}$$

where $O^{r_o,even}_{\theta_o}(x,y)$, $O^{r_o,odd}_{\theta_o}(x,y)$ are the responses of the even and odd symmetric log-Gabor filters, respectively. The real-valued function given in (1) can be multiplied by the frequency representation of the image and, transform the result back to the spatial domain, the responses of the oriented energy filter pair are extracted as simply the real component for the even-symmetric filter and the imaginary component for the odd-symmetric filter. Let $Z_{(r_o,\theta_o)}$ be the transformed filtered output. The responses of even and odd symmetric log-gabor filters are expressed as:

$$O^{r_o,even}_{\theta_o} = Re(Z_{(r_o,\theta_o)}); \qquad O^{r_o,odd}_{\theta_o} = Im(Z_{(r_o,\theta_o)}) \tag{3}$$

The total energy over the entire image can be computed as as follows:

$$\widetilde{E}(\theta_o) = \sum_{x=1}^{m} \sum_{y=1}^{n} (E_{\theta_o}^{r_o}(x, y)) \tag{4}$$

where $m \times n$ pixels is the size of the text block. This is nothing but the histogram function for the energy map. This energy histogram is a global feature which expresses the oriented energy distribution in a given text block. We will use it to classify the underlying script in the text block.

Features Used for Classification. The oriented energy distribution characterises a script texture as it indicates the dominance of individual subpatterns (lines of different orientation). For instance, the Hindi script is characterised by the dominance of horizontal lines, whereas this is not true for Malayalam (see Fig. 2). Hence, we extract such features that are relevant to the problem in hand.

Oriented local energy responses: The oriented local energy is computed as given in equation (4). A dominance of lines at a specific orientation θ is signalled by a peak in $\widetilde{E}(\theta)$. This is computed for text blocks (extracted as discussed in Sec. 2.2) using log-Gabor filters designed at 8 equi-spaced orientations ($0\,^\circ, 22.5\,^\circ, 45\,^\circ, 77.5\,^\circ, 90\,^\circ, 112.5\,^\circ, 135.5\,^\circ$ and $180\,^\circ$) and at an empirically determined optimal scale. The energy values are normalised for a reliable classification and can be derived as

$$E(\theta_i) = \left\{ \frac{\widetilde{E}(\theta_i)}{max\left\{\widetilde{E}(\theta_j) | j = 1, \cdots, 8\right\}} | i = 1, \cdots, 8 \right\} \tag{5}$$

Here, index i denotes the corresponding orientation ($0\,^\circ, 22.5\,^\circ, \cdots, 180\,^\circ$). We have dropped r_o for convenience, as we computed energy in only one scale. Several used features are extracted from this normalised energy. We describe these features and their method of computation next. The features are presented in the order of their saliency in the final classifier.

1. **Statistical features:** The energy profile for all the ten Indian scripts can be seen in Fig. 3. The shape of energy profiles differ from each other based on the underlying script. The energy in some scripts, like Devanagari which contain more linear patterns, is concentrated more in fewer channels with less spread into the neighbouring channels. On the other hand, energy is distributed more or less evenly amongst neighbouring channels for scripts which have curved shape,like Oriya. To capture such variation in the energy profile, we can use the relative strength in $E(\theta)$ for adjacent orientation channels. This is derived by finding the first difference in $E(\theta)$ as follows

$$\Delta E_i = \begin{cases} E(\theta_i) - E(\theta_{i+1}) \text{ if } i = 1, \cdots, 7 \\ E(\theta_8) - E(\theta_1) \quad i = 8 \end{cases} \tag{6}$$

These eight feature values provide enough discriminant information to perform a first level classification of scripts. Furthermore, the choice features

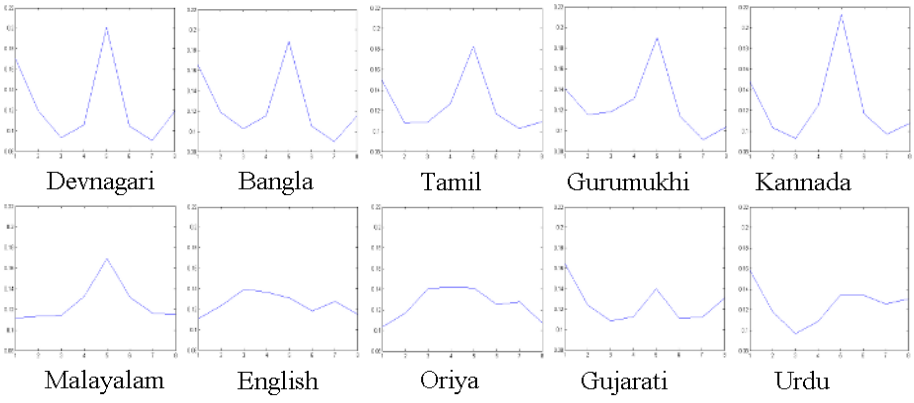

Fig. 3. Energy plots for each Indian language script

also makes our scheme invariant to font size since the $E(\theta_i)$ will proportionally change in each orientation with the change in the font size while the ΔE is less susceptible to change. In order to make the classification robust to skew, it can be observed that skew in the script image results in a shift in the values of ΔE to their neighbouring orientations according to the skew angle but it would not make any change in these average values. Hence, we extract two more features as follows:

$$\overline{\Delta E} = \frac{1}{8}\sum_{i=1}^{8} \Delta E_i; \qquad \overline{E} = \frac{1}{8}\sum_{i=1}^{8} E(\theta_i) \tag{7}$$

2. **Local features:** The above features capture global differences among scripts. In order to capture the finer differences between similar scripts a set of local features are needed. For instance, Devanagari,Bangla, Tamil and Gurumukhi have similar scripts. A fine analysis is required for their further classification. It is observed that the similar scripts also have similar energy profiles (shape) which is captured in ΔE. However, these energy values $E(\theta_i)$ actually differ drastically in non-adjacent orientation channels. This can be a useful information and hence is captured in the following features. Here, the ratio of energies $E(\theta_i)$ is computed for two non-adjacent orientations θ_i.
3. **Horizontal profile:** Finally, there are some scripts which are distinguishable only by strokes used in the upper part of the words. For instance, Devanagari and Gurumukhi scripts both use a headline but differ in the strokes above the headline. This difference can be captured from the horizontal projection profiles of a whole text block (not of individual text lines as in [6, 5, 4]). The profiles of these scripts' text blocks are shown in Fig. 4. It can be seen that region above *headline* (signalled by three high peak in each profile) differ in both scripts. The average value of peaks in that region is higher for Gurumukhi script.

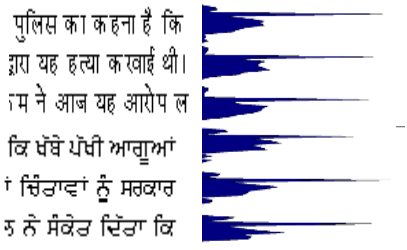

Fig. 4. Devanagari and Gurumukhi scripts with their corresponding horizontal profile

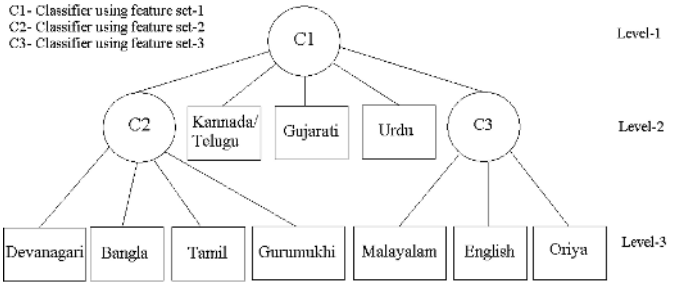

Fig. 5. Classification scheme for Indian language scripts

2.4 Script Classification Scheme

We now propose a hierarchical script classifier for the Indian scripts using the *globally* extracted features listed in the previous section. These features capture discriminating information among the scripts and get more script specific in the successive levels of the classifier. In the highest level, gross information is used for a broad categorisation, whereas in the lower levels categorisation is performed using finer analysis of the underlying script.

The proposed hierarchical classifier uses a two-level, tree based scheme (shown in Fig. 5) in which different sets of principle features are used at the non-leaf

Table 1. Features used in classifier at different levels

Feature set	Features used	Classifier
1	8 ΔE_i values $\overline{\Delta E}$ $\overline{E}$	C1
2	$ratio(E_3, E_7)$ $ratio(E_3, E_1)$ $ratio(E_7, E_1)$	C2
3	$ratio(E_5, E_1)$ $ratio(E_8, E_1)$ Horizontal profile	C3

nodes. The root node classifies scripts into five major classes using feature set-1. In the second level of the scheme, there are five major classes in which three are single member classes. Rest of the two classes have four and three members, respectively. On the respective non-leaf nodes, different feature sets are used, based on their effectiveness in discriminating between members of that sub-class. In the third level, all the leaf nodes belong to a single member class. Table. 1 gives a complete list of feature sets used by each classifier.

3 Experiments and Results

3.1 Data Collection

At present, in India, standard databases of Indian scripts are unavailable. Hence, data for training and testing the classification scheme was collected from different sources. These sources include the regional newspapers available online [14] and scanned document images in a digital library [9].

3.2 Selection for the Best Classifier

In order to identify the most appropriate classifier for the problem at hand, we experimented with different classifiers. Matlab pattern recognition toolbox [11] was used to conduct these experiments. Well known classifiers based on different approaches were chosen for the experiments [13]. These were: k-nearest neighbor, Parzen density, quadratic Bayes, feed-forward neural net and support vector machine based classifiers.

Table. 2 compares the performance of the classification scheme when different classifiers are used at every node of the proposed classifier. It can be seen that the nonparametric classifiers (K-NN and Parzen window) perform the best among all classifiers. The best classification rate obtained is 97.11% with 10 different Indian scripts after testing on a large script test data set (2978 text blocks). This

Table 2. Error rate for different classifiers

Classifier	Remarks	Error rate
Quadratic Bayes normal classifier	Gaussian with full variance	37.34%
Neural network based classifier	Three hidden layers	4.84%
K-Nearest Neighbor Classifier	k is optimized using leave-one-out error estimation	2.89%
Support vector classifier	Polynomial kernel	34.96%
	Redial basis kernel	36.57%
	Exponential kernel	6.98%
Parzen density based classifier	Kernel width is optimized using leave-one-out error estimation	3.16 %

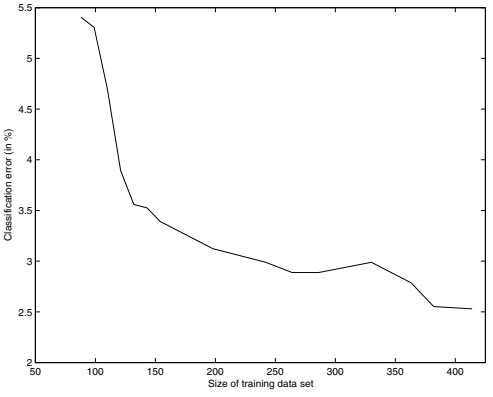

Fig. 6. Dependency of classification error on the training data set size

indicates the effectiveness of the proposed features. Since all these features were globally extracted, the good performance demonstrates the strength and effectiveness of global analysis based classification which is also computationally efficient. This is in contrast to the previous approaches to Indian script identification which use local analysis (of connected components) and achieve the same level of performance as the proposed approach. Due to the lack of standard (benchmark) Indian scripts data, it is not possible to directly compare the performance of the proposed scheme with previously reported script classification schemes.

3.3 Optimal Size of Training Data Set and Feature Set

In order to determine the optimal size of the training data set required for the best performance of the classifier we tested with varying number of training data set (see Fig. 6). We found that with the size of 264 data set block, our classifier attains best performance with a classification error of 2.89%. This size of data set can easily be collected. A possible reason for the size being small is that the extracted features best represent the discriminant information among scripts. Ten features are used at the root node of the classifier (as explained in section (2.3)). These features are the local energy computed at the output of 8 oriented filters with an orientation resolution of 22.5°. We examined the influence of the resolution on the classifier performance and our finding was that with a resolution reduction of 50% the performance degrades to 95% (from 97% with 8 oriented filters).

3.4 Performance Analysis

As mentioned earlier, based on testing the proposed scheme on 2978 individual text blocks, the classification accuracy obtained is 97.11%. It was found that a skew of upto 4 degree has no effect on the classifier performance. To improve this robustness further, more rotationally invariant features derived from the oriented energy reponses can be added [4]. A Confusion matrix for the proposed classification scheme is given in Table. 3. The major diagonal term indicates the number

Table 3. Confusion Matrix of the proposed script classifier for 10 different Indian scripts. (Here De=Devanagari, Ba=Bangla, Ta= Tamil, Gu= Gurumukhi, Ka= Kannada, Ma=Malayalam, Ro= Roman (English), Or= Oriya, Guj= Gujarati, Ur= Urdu.)

	Classified									
Actual	De	Ba	Ta	Gu	Ka	Ma	Ro	Or	Guj	Ur
De	203			1						
Ba		282		3						
Ta		1	283	23		3				
Gu	1		9	248						
Ka					596	3		3		
Ma						279		9		
Ro		7	6				264	2		
Or					1	5	8	231	1	
Guj									263	
Ur										243

of correctly classified testing samples while the off-diagonal term indicates the number of misclassified samples. From the matrix, it can be observed that the worst performance is only in the case of Tamil and Gurumukhi. It is interesting to see that both the scripts have similar energy profile (given in Fig. 3) even though they are perceptually different(can be viewed from Fig. 2). Thus it appears that the extracted global features are insufficient to discriminate such cases at present.

4 Conclusion and Future Work

Based on our observation human ability to classify unfamiliar scripts we have examined the possibility of using only global analysis of scripts for identifying them. We have presented a set of local energy based features for accomplishing classification in a hierarchical fashion extracted from oriented log-Gabor filters. These features have been used to develop a script classification scheme for Indian language scripts. The scheme is very simple and practical for a script based retrieval system. It requires a very simple preprocessing followed by a feature extraction process. Test results of the proposed classification scheme has revealed that good performance accuracy (97%) is obtainable using global analysis thereby illustrating its strength and utility. The scheme can be extended to multiple scales to handle scripts printed at a different resolution. The proposed scheme can be used for other language scripts as well with minimal modification.

References

1. A. Spitz., Determination of the script and language content of document images. IEEE Trans. Pattern Anal. Mach. Intell.**19(3)** (1997) 235–245.
2. J. Hochberg, L. Kerns, P. Kelly, and T. Thomas., Automatic script identification from images using cluster-based templates. IEEE Trans. Pattern Anal. Mach. Intell. **19(2)** (1997) 176–181.

3. S. L. Wood, X. Yao, K. Krishnamurthi, and L. Dang., Language identification for printed text independent of segmentation. Proceedings of International Conference on Image Processing **3** (1995) 428–431.
4. T. N. Tan., Rotation invariant texture features and their use in automatic script identification. IEEE Trans. Pattern Anal. Mach. Intell. **20(7)** (1998) 751–756.
5. U. Pal, S. Sinha, and B. B. Chaudhuri., Multi-script line identification from indian document.Seventh International Conference on Document Analysis and Recognition **2** (2003) 880–884.
6. S. Chaudhury and R. Sheth., Trainable script identification strategies for indian languages. Fifth International Conference on Document Analysis and Recognition (1999) 657–660.
7. W. Chan and J. Sivaswamy., Local energy analysis for text script classification. Proceedings of Image and Vision Computing New Zealand (1999) .
8. W. Chan and G. G. Coghill., Text analysis using local energy. Pattern Recognition **34(12)** (2001) 2523–2532.
9. Digital Library of India. http://dli.iiit.ac.in/.
10. M. C. Morrone, D. C. Burr, Feature detection in human vision: A phase-dependent energy model. Proceedings of the Royal Society, London Series B **235** (1988) 221–245.
11. PRTools: A Matlab Toolbox for Pattern Recognition. http://www.prtools.org/.
12. A. K. Jain and Y. Zhong., Page segmentation using texture analysis. Pattern Recognition **29** (1996) 743–770.
13. R. Duda, P. Hart, and D. Stork., Pattern Classification. second edition, New York: John Wiley and Sons (2001).
14. http://www.samachar.com/.
15. X. Zhitao, G. Chengming, Y. Ming, and L. Qiang, Research on log Gabor wavelet and its application in image edge detection. Sixth International Conference on Signal Processing, (2002).