International Journal of Document Analysis and Recognition (IJDAR) manuscript No.
(will be inserted by the editor)

Gopal Datt Joshi · Saurabh Garg · Jayanthi Sivaswamy

A generalised framework for script identification

Received: date / Accepted: date

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 model script identification as a texture classification problem and examine a global approach inspired by human visual perception. A generalised, hierarchical framework is proposed for script identification. A set of energy and intensity space features for this task is also presented. The framework serves to establish the utility of a global approach to the classification of scripts. The framework has been tested on two datasets: 10 Indian and 13 world scripts. The obtained accuracy of identification across the two datasets is above 94%. The results demonstrate that the framework can be used to develop solutions for script identification from document images across a large set of script classes.

Keywords Script Identification · Framework · Indian Documents · Multilingual Document Images · Global Approach · Log-Gabor Filter Bank.

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 old fragile documents in typically large databases. These documents have scientific or historical or artistic value and can be written in different scripts. 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) select appropriate script-specific OCRs and (iii) search online archives of document images for those containing a particular script.

Most of the existing literature on script identification either focus on the development of new approaches or on the improvement of existing approaches which work for some specific application or specific script classes. As a result, a generalised approach to the problem has not been considered which handles all flavours of problem under a common framework. We argue that there are some essential factors which need to be considered before choosing or designing a script identification scheme for any multi-lingual application. These factors are: (a) complexity in pre-processing, (b) complexity in feature extraction and classification, (c) computational speed of entire scheme, (d) sensitivity of the scheme to the variation in text in document (font style, font size and document skew), (e) performance of the scheme, and (f) range of applications in which the scheme could be used. Performance of the scheme includes accuracy reported and selection of testing data. Currently, individual approaches are designed such that they can effectively deal with some of the factors listed above (not all). None of them has showed their potential to become a generalised script identification scheme.

Existing script classification approaches can be classified into two broad categories, namely, local and global approaches. By local approach is meant an approach which analyses a document image at the level of a list of connected components (like line, word and character or LWC) and such components require segmentation of the image as a preprocessing step. By global approach is meant an approach which employs analysis of regions comprising at least two lines and do not require LWC type of fine segmentation.

Gopal Datt Joshi, Saurabh Garg and Jayanthi Sivaswamy
Centre for Visual Information Technology,
International Institute of Information Technology,
Gachibowli, Hyderabad - 500 032,
Andhra Pradesh, INDIA.
Tel.: +91-40-2300 1967, Extn: 139
Fax: +91-40-2300 1413
E-mail: gopal@research.iiit.ac.in, jsivaswamy@iiit.ac.in

In the category of local approaches, Spitz [1] proposed a method for discriminating Han (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, this method uses the optical density distribution in a character and the 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 and multi-script country, languages have scripts of their own, though some scripts may be shared by two or more languages. Some classification methods have been proposed for Indian language scripts as well[8, 7]. These use Gabor energy features extracted from connected components [8] or statistical and topological features [7]. In [8], 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 [7] uses horizontal profiles, statistical, topological and stroke based features. These features are chosen at a non-terminal node to get an optimum tree classifier. These features, however, are very fragile in the presence of noise.

Global approaches, in contrast, are designed to identify the script by analysing blocks of text extracted from the document image. Wood [4] proposes methods using Hough transforms and analysis of density profile resulting from projection along text lines. However, it is not clear from the description, how the projection profile is being analysed to determine the script. A texture based classification scheme in [5] uses the rotationally invariant features from Gabor filter responses. Since the texture images formed by different scripts patterns are found to be inconsistent, the text blocks are normalized to have equal height and width, with uniform spaces between the lines and the words. As a result of these processing steps, the scheme is reasonably expensive. Busch et al. [6] evaluated number of texture features including gray-level co-occurrence matrix , Gabor filter bank energies, and a number of wavelet transform-based features for script identification. Chan et al. [10, 9] 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. Table. 1 summarises the features of global and local approaches based on the factors discussed above.

In summary, when using a local approach for the script identification, the success of classification is dependent on the effectiveness of the pre-processing stage namely, accurate LWC segmentation. This presents a paradox, as LWC segmentation is best performed when the script class of the document, whose script is yet to be identified, is known [5]. Even when the script classes are known from the training data, testing requires performing LWC segmentation prior to script identification and it is difficult to find a common segmentation method that works well across all the script classes. For example, if the target script classes are Roman, Devanagari and Chinese, the LWC segmentation method required for Roman will not work well for Devanagari or Chinese. Due to this limitation, local approaches can not qualify as a generalised scheme. Whereas, global schemes show potential for a generalised approach to the script identification problem. In such schemes, it is not necessary to extract individual script components, making them ideal for degraded and noisy documents or situations. From the preceding discussion, we can infer that global approaches have potential to address script identification in more general way, as compared to local approaches. However, all reported global approaches have failed to exploit it as a generalised solution.

The rest of the paper is organised as follows. In Section 2, we show how the script identification problem can be reformulated as a texture classification problem, and call for a hierarchical strategy using global analysis. A framework for script identification is proposed in Section 3 and some features that can be used for classification are identified in Section 4. Sections 5 and 6 present the results of testing the proposed framework on an Indian script and world script datasets, respectively. We close the paper with a discussion of the obtained results and some conclusions in Section 7.

2 Background

A remarkable feature of the human visual system (HVS) is the ability to discriminate between unfamiliar scripts just based on a short visual inspection. This indicates that the HVS employs a global rather than a local approach to the discrimination task. Examination of the type of processing carried out at the pre-attentive (image data driven processing) level of the HVS reveals the presence of cells which extract oriented line features. These cells have been shown to be modelled by Gabor functions [15]. With this as a starting point, we consider script identification as a process of texture analysis and classification similar to [5]. Previous work on the texture based approaches pass images though filters which are designed at various scale and orientation and use their

Table 1 Table of comparison between local and global approaches for script identification (SI)

	Local approaches	Global approaches
Pre-processing	Complex and script dependant	Simple and script independent
Feature extraction process	requires LWC segmentation	LWC segmentation not required
Sensitivity in used features	Prone to document image noise	Less prone
Robustness to font size, type and skew in Document	Moderate	Moderate
Scope of application	Narrow	Wide
Potential to become a generalised approach for SI	Possibility is limited (due to the segmentation paradox)	High potential

responses to train a single classifier. These methods assume that the important features are implicitly encoded in the filters' responses. These filter responses contain ambiguity in their information content which results in a complex classifier and a large number of training data to attain good performance. No study has been carried out to understand the basic properties and nature of the captured features.

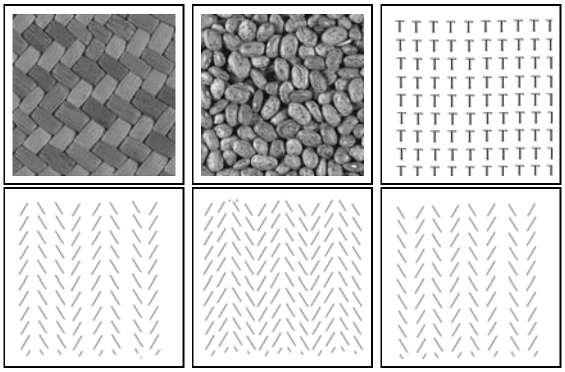


Fig. 1 Sample natural and synthetic textures

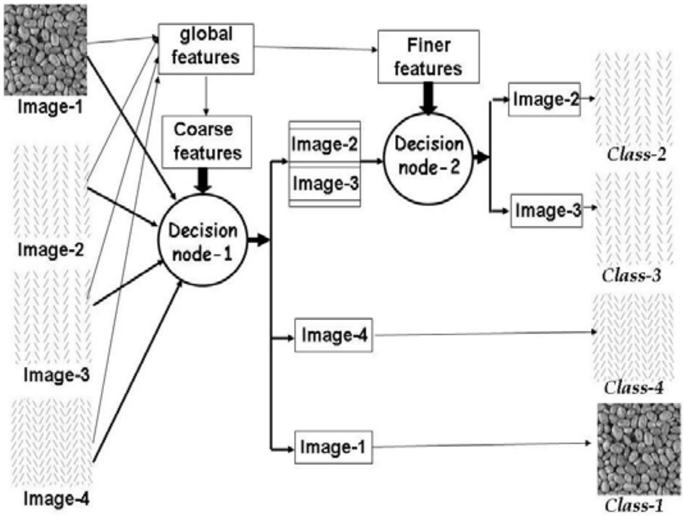


Fig. 2 A schematic overview of the framework development.

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 the first row of Fig.1 showing the cross section of a basket, a pile of seeds and a synthetic image. Each of these texture images has its own subpattern such as different size rectangles in the basket texture, different size ovals in the seed texture and 'T' shape patterns in the 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 careful 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 coarse analysis of the three images while the perception resulting from a careful examination is a result of a finer analysis of the images.

Based on the above observation, we assert that a general approach to script classification (identification) can be developed based on global features (pre-attentive) and do so by noting the following. Script patterns are textures formed by oriented linear/curvilinear subpatterns. Later, we will use these oriented line subpatterns as important features for script classification. Furthermore, any script (not a language) can be characterised by the distribution of linear subpatterns across different orientations. 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. The schematic development of the proposed approach is summarised in the Fig. 2. Global features are extracted from the input images to broadly classify textures into three groups. One group consists of *Image-2* and *Image-3* due to their similar perception. For further refinement in classification, local features are derived from global features in the next stage to help form single member (texture) classes. Here, global features are used for coarse classification whereas local features are used for finer classification. In the next section, we present a framework to address the script identification problem based on the above discussion.

3 A Generalised Framework to Script Identification

We propose a general framework to address the problem of script identification. The key features of this framework are: (i) a hierarchical strategy to help group given scripts into homogeneous script classes before attempting to classify them into single-script classes and (ii) the use of global analysis throughout. In the first step, the document image is processed to make it suitable for analysis. Next, in the feature extraction stage the document is sub-divided to fixed-size text blocks. *All* requisite features are then extracted from the text blocks. Since the features are extracted from a text block without any segmentation, they are global features. The features that are to be extracted are chosen to represent coarse to fine level of information about the text block. The coarse level features are passed to a base classifier which assigns the given script image to one of the broad classes. Each broad class has its own classifier having different input features. All these input features contain finer discriminating details of the scripts. They can help refine the classification of the given scripts and achieve a single script class. In this framework, finer features are to be derived based on prior knowledge (derivable from the training data set) of the target script classes, using global analysis. Fig. 3 shows a two-level hierarchical classification. This level can be further subdivided based on the problem at hand. In general, the number of levels is proportional to the number of homogeneous script classes present. In successive levels, a script class is separated from homogeneous classes by exploiting its unique fine(r) features.

4 Feature Extraction

4.1 Log-Gabor filtering

A traditional choice for texture analysis is to use Gabor filters at multiple scales. According to the formulation in Section 2, script patterns are textures formed by oriented linear/curvilinear subpatterns. Therefore, any script class can be characterised using linear/curvilinear subpatterns. We have selected oriented local energy features to capture these subpatterns, with the local energy defined to be the sum of the squared responses of a pair of conjugate symmetric filters. Local energy was proposed as a generalised feature detector capable of capturing edge and line information using a filter bank (multiple orientations and scales) [15]. However, given that the texture of interest in our case is made of only linear subpatterns, it suffices to carry out the local energy computation at a single scale with a sufficiently wide bandpass filter, thus reducing 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 larger bandwidths from 1 to 3 octaves which makes the features more effective, reliable and informative [3]. 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 (polar function) of a log-Gabor filter is expressed as

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

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

$$E_{\theta_o}^{r_o}(x,y) = \sqrt{(O_{\theta_o}^{r_o,even}(x,y))^2 + (O_{\theta_o}^{r_o,odd}(x,y))^2} \quad (2)$$

where $O_{\theta_o}^{r_o,even}(x,y)$, $O_{\theta_o}^{r_o,odd}(x,y)$ 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 the result transformed 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 the even and odd symmetric log-Gabor filters are expressed as:

$$O_{\theta_o}^{r_o,even} = Re(Z_{(r_o,\theta_o)}); \qquad O_{\theta_o}^{r_o,odd} = Im(Z_{(r_o,\theta_o)}) \quad (3)$$

The total energy over the entire image is computed as follows:

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

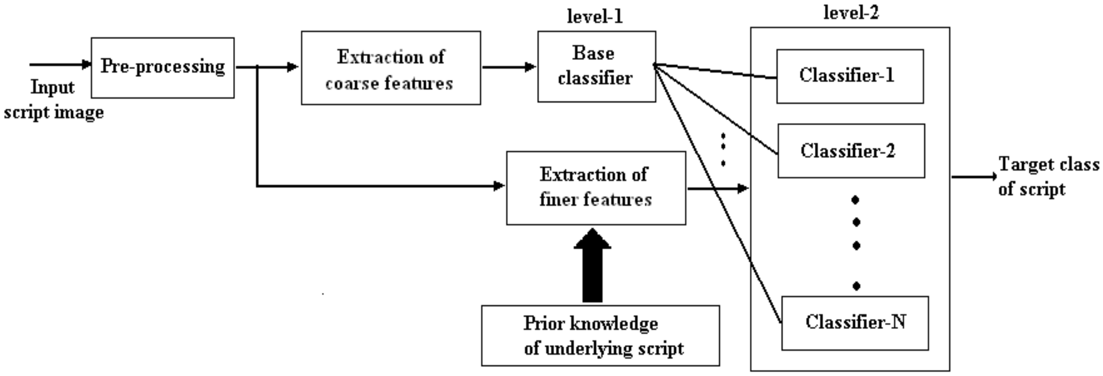


Fig. 3 A generalised framework for script identification

where $m \times n$ pixels is the size of the text block. This is nothing but the orientation 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.

4.2 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. 5). Hence, we extract such features that are relevant to the problem in hand.

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 Section 5.3) using log-Gabor filters designed at eight equi-spaced orientations (0 °, 22.5 °, 45 °, 67.5 °, 90 °,112.5 °, 135 ° and 157.5 °). The energy values are normalised to make them invariant to font size. It can be derived as

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

Here, index i denotes the corresponding orientation. We have dropped r_o for convenience, as we computed energy at only one scale. Several 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 classification.

1. **First level features (Coarse features)**: The energy profile, $E(\theta)$, for all the ten Indian scripts can be seen in Fig. 6. 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 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 and Armenian. To capture such variation in the energy profile, we can use the relative strength in adjacent orientation channels. This is derived by finding the first difference in $E(\theta)$ as follows

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

These eight feature values provide enough discriminant information to perform a first level broad classification of scripts. Furthermore, the choice of features also makes our scheme invariant to font size across text blocks, since the energy will proportionally change in each orientation with the change in the font size while the difference in energy 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 affect the 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. **Second level features (Finer features)**:
Ratios of normalised energies: The above features capture global differences among scripts. In order to capture the fine discrimination information between scripts, a set of fine 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 almost similar scripts also have similar energy profiles (shape) which is captured in ΔE. However, these energy values actually differ drastically in different orientation channels. This can be a useful information

and hence is captured by the ratio of energies $R(i,j)$ which is defined as:

$$R(i,j) = \frac{E(\theta_i)}{E(\theta_j)}; \qquad i,j = 1,\cdots,8 and i \neq j \tag{8}$$

The orientation pairs (i,j) are empirically determined in order to maximise the value of R and provide maximum discrimination among the script classes.

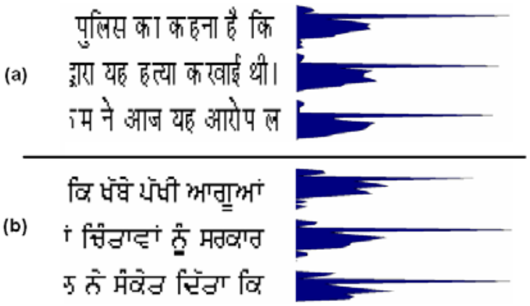


Fig. 4 (a) Devanagari and (b) Gurumukhi scripts with their corresponding horizontal profiles

Horizontal profile: All the features that were presented above are extracted from the energy profile of a given text block. The proposed framework permits the usage of global features from the intensity profile of the text block as well. The latter type of features are particularly useful to exploit the unique spatial arrangement of scripts, Examples are umlauts, matras, accents etc., that appear above or below the base lines. In Indian scripts, some scripts are distinguishable 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 information is available in the intensity profile of the text block. They can be extracted using horizontal projection of the *entire* text block. These profiles are shown in Fig. 4. From this figure, it can be seen that region above the *headline* (signalled by three high peak in each profile) differ in both scripts. The average value in that region is higher for Gurumukhi than the Devanagari script. These regions can be extracted using peak points and their next corresponding local minima points in the profile.

5 Experimental Results on Testbed Dataset

5.1 Indian language scripts

India has 18 official languages which includes Assamese, Bangla, English, Gujarati, Hindi, Konkanai, Kannada, Kashmiri, Malayalam, Marathi, Nepali, Oriya, Punjabi, Rajasthani, Sanakrit, Tamil, Telugu and Urdu. All the Indian languages do not have 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 text blocks of these images are shown in Fig. 5.

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 *headlines* and the characters of words in these scripts are generally connected by these headlines (two of them are shown in Fig. 4). Due to these properties, they can be differentiated 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 present a method to extract features through a global analysis of a given text document image and use them to classify the underlying script.

5.2 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] (using screen capture) and scanned document images (at 600 dpi) from a digital library [12].

5.3 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 [17] [19] could be readily 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. We do not attempt to homogenise these parameters.

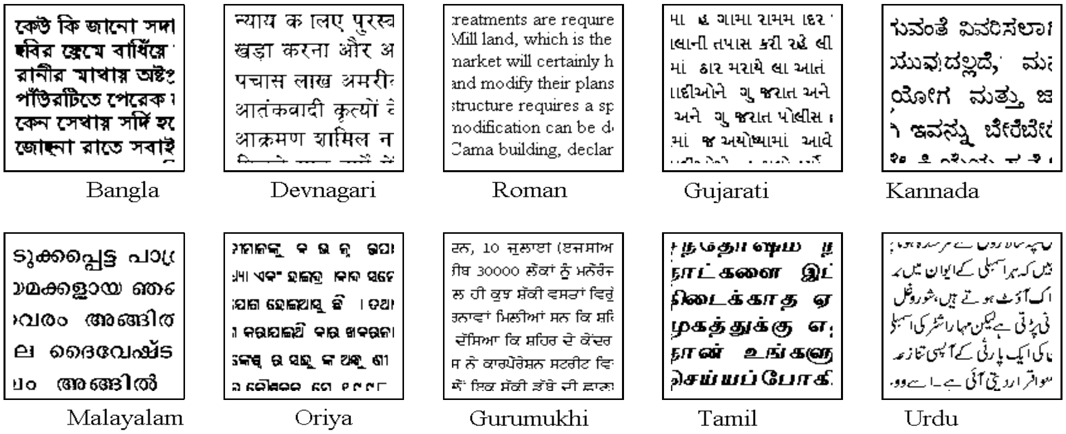

Bangla
Devnagari
Roman
Gujarati
Kannada
Malayalam
Oriya
Gurumukhi
Tamil
Urdu

Fig. 5 Sample text blocks of Indian language scripts

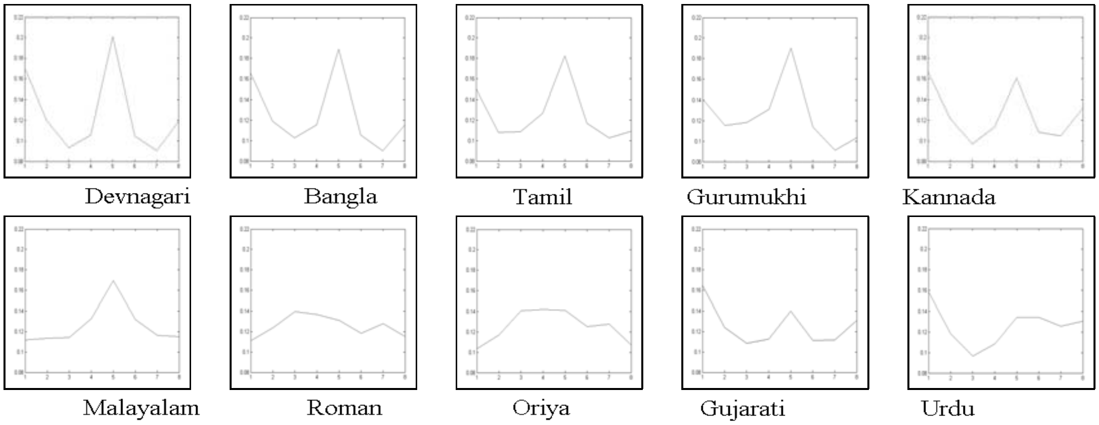


Fig. 6 Energy plots for each Indian language script

5.4 Script classification scheme: Classifier-T

We now present a hierarchical classifier for the Indian scripts using the *globally* extracted features listed in the previous section. These features capture coarse discriminating information among the scripts at the first level. In successive levels, they capture more script specific information. 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. 7) in which different sets of principle features are used at the non-leaf nodes. The root node classifies the 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. 2 gives a complete list of feature sets used by each classifier.

Table 2 Features used by classifiers at different levels

Feature set	Features used	Classifier
1	Eight $\Delta E_{\theta i}$ values, $\overline{\Delta E}$, $\overline{E}$	C1
2	R(5,1); R(8,1) Horizontal profile	C2
3	R(3,7); R(3,1); R(7,1)	C3

5.5 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 [16] was used to conduct these experiments. Well known classifiers based on different approaches were chosen for the experiments [18]. These were: k-nearest neighbor, Parzen density, quadratic Bayes, feed-forward neural net and support vector machine based classifiers.

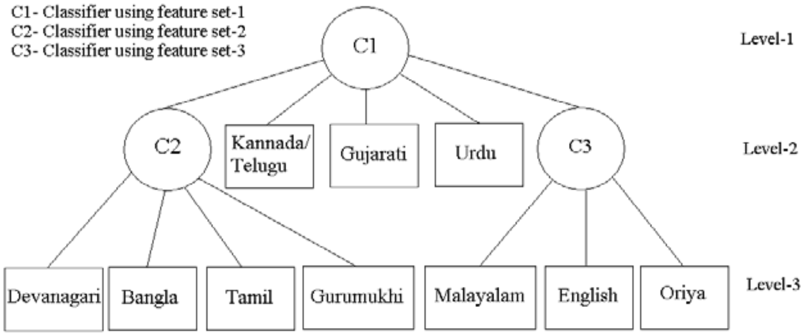


Fig. 7 Classification scheme for Indian language scripts

Table 3 Error rate for different candidate 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	5.43%
	Redial basis kernel	6.74%
	Exponential kernel	4.02%
Parzen density based classifier	Kernel width is optimized using leave-one-out error estimation	3.16 %

Table.3 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). The root classifier $C1$ contributes an error of 0.8% in the overall error (2.88%). This 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. The effect of skew was also studied. The error rates obtained for a skew of 4, 5 and 6 degrees were -2.88, -4.8 and -6.9, % respectively. Thus, a skew of up to -4 degrees is tolerated. Due to the lack of availability of standard Indian scripts data, it is not possible to directly compare the performance of the proposed scheme with previously reported script classification schemes.

5.6 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 varied the size of the training data set (see Fig. 8). 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. This attests to the discrimination power of the extracted features.

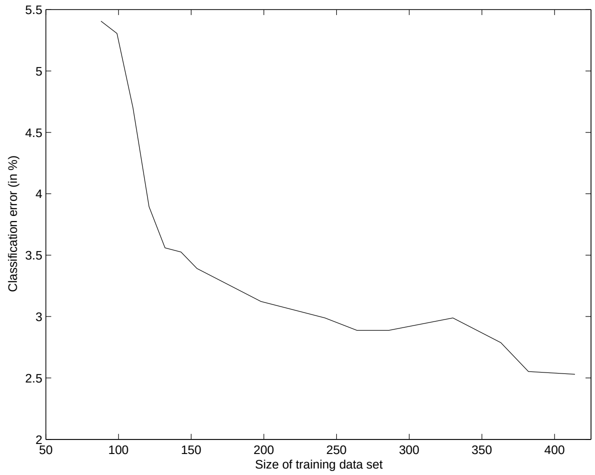


Fig. 8 Dependency of classification error on the training data set size (Indian scripts)

Ten features are used at the root node of the classifier (as explained in Section (4.2)). These features are based on 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 found that with a resolution reduction of 50% (four orientations, results 6 features) the perfor-

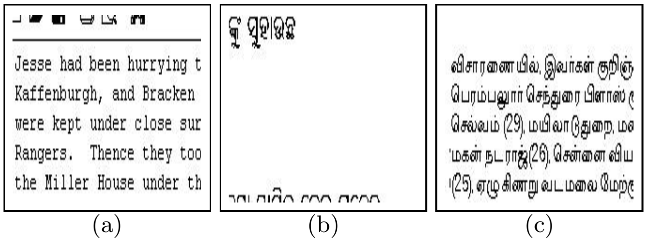

Jesse had been hurrying t
Kaffenburgh, and Bracken
were kept under close sur
Rangers. Thence they too
the Miller House under th
(a)
(b)
(c)

Fig. 9 Sample text blocks which are misclassified by the classifier-T

mance degrades to only 95% (from 97% with 8 oriented filters). This means that by reducing the filter bank size from 8 × 2 to 4 × 2 the loss in accuracy is only 2%.

Furthermore, we have tested root node classifier $C1$ to classify 10 Indian script classes using feature set-1 (given in Table 2). The root classifier having 10 target script classes was trained using these features. It was found that it achieves an accuracy of 91% which is less than the actual accuracy reported by hierarchical classifer. This demonstrates the advantage of employing a hierarchical strategy for classification.

5.7 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 up to −4 degrees 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 [5]. A confusion matrix for the proposed classification scheme is given in Table. 4. The major diagonal term indicates the number 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. 6) even though they are perceptually different(as seen from Fig. 5). Thus it appears that the extracted global features are insufficient to discriminate such cases at present. Some mis-classification occur due to the complexity present in the extracted text blocks. This complexity are namely, insufficient textual content (blank text blocks), the presence of horizontal/vertical lines and numerals in the extracted text blocks. Some sample misclassified text blocks are shown in Fig. 9.

6 Experimental Results on Benchmark Dataset

Next, we applied the proposed framework on the dataset used by Hochberg et al. [2] to test the generality of the proposed framework. According to [2] the scripts used in the dataset were chosen for their wide coverage of the world's languages and their ready availability. We have tested our proposed framework on this benchmark dataset using the same set of global features (expect horizontal profile) without re-tuning the filter parameters. The same set of first level (coarse) features were used for the classifier at the root node. At the second level, the optimal set of orientation channels (for finding the ratios of energy), were once again found as per the guidelines provided in Section 4.2. The following subsections present experimental details and results on benchmark dataset.

6.1 Description of benchmark dataset

The dataset consists of 195 document images of thirteen (13) scripts which represent world's languages: Arabic, Armenian, Burmese, Chinese, Cyrillic, Devanagari, Ethiopic, Greek, Hebrew, Japanese, Korean, Latin (Roman) and Thai. Sample text blocks of these languages are shown in Fig. 10. The scanned images were acquired from different sources namely, books, newspapers, magazines, and computer printouts. The images and a summary table are made available by the authors at [13]. These images include a range of font types, styles for each script, such as serif and sans-serif Roman. Following the method in [2] we have divided images into two different sets: 1) Training/test images consisting of 7 of available 15 images from each script class, and 2) Difficult images, based on the guideline given in the [2]. The difficult images have fonts not present in the training and test set and some contain multi-lingual text. They are used to assess the robustness of the script identification method.

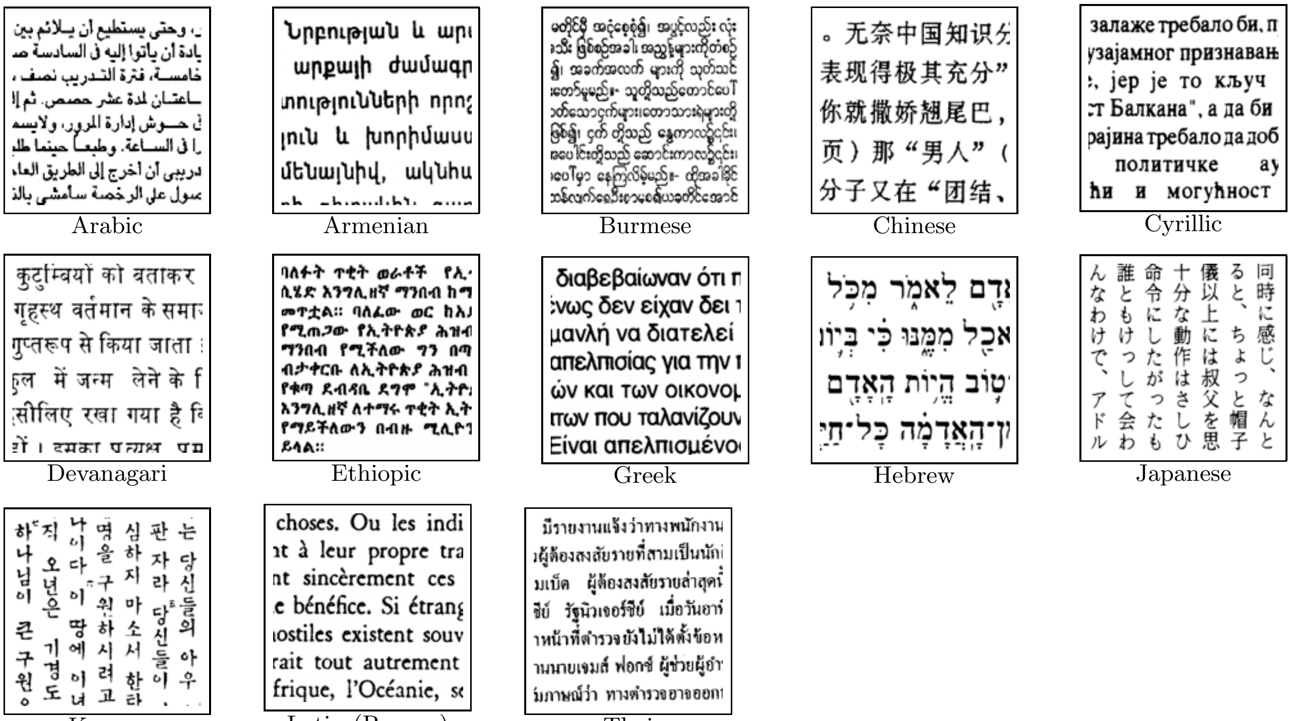
All document images in the dataset are typeset (not handwritten) and have black-on-white letters, except few difficult images which contain white-on-black text. The images were scanned at 400 dpi. Training and test image set contain only text in single script. In contrast, difficult images may include isolated foreign characters or words and illustrations. Some Japanese and Chinese images include horizontal or vertical lines in the text. The entire dataset includes images with overall rotation up to approximately 10 degree due to scanning process of the images. Particularly, in the difficult set, some images contain fragmented or conjoined characters, or extraneous material such as creases and stray lines. Fig. 13 shows a sample difficult image with lower/uppercase Greek letters of different size, numerals, blank regions and vertical lines.

6.2 Feature set

The images in the test (Indian scripts) and benchmark (world scripts) datasets are different, particularly in terms

Table 4 Confusion Matrix of the classifier-T. (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	Gur	Ka	Ma	Ro	Or	Guj	Ur
De	203			1						
Ba		282		3						
Ta		1	283	23		3				
Gur	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


Arabic

Armenian

Burmese

Chinese

Cyrillic

Devanagari

Ethiopic

Greek

Hebrew

Japanese

Korean

Latin (Roman)

Thai

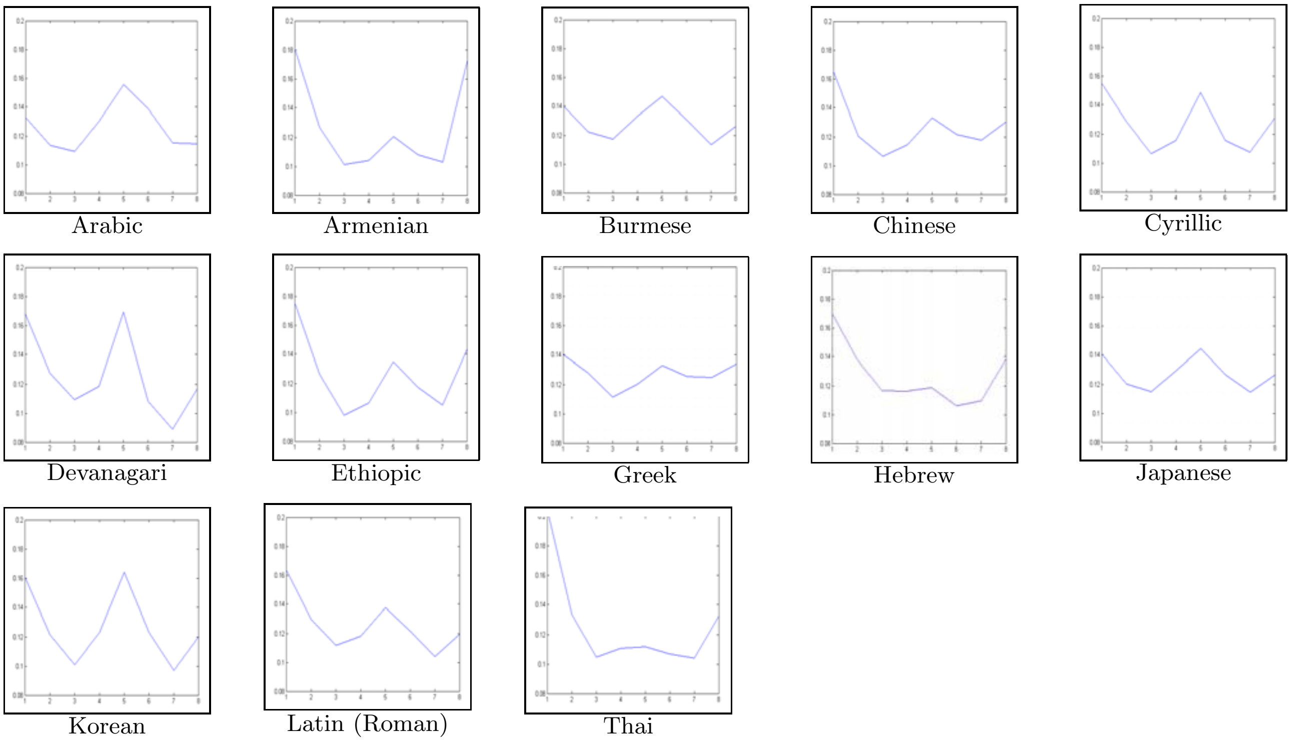
Fig. 10 Sample text blocks of world scripts from benchmark dataset

of font size. As a result, using the same size of text blocks for both datasets is unwise since it will result in capturing insufficient amount of text in a block. Hence, a text block size of 400 × 400 was used for feature extraction for the benchmark dataset. As mentioned earlier, the same set of global features were used for both datasets. The number of levels in the hierarchical solution to obtain 13 single script-class solution depends on the choice of the finer level features, which are to be determined based on the prior knowledge of the scripts (from training dataset). We had identified ratios of energy (R(i,j)) and horizontal profiles of text blocks as useful features at the finer level. The training set was used to determine the requisite pairs of orientations for the R(i,j) feature set that lead to single script-classes. The horizontal profiles of the benchmark dataset did not contain any useful information and hence was not used. A complete set of the features that were used are shown in the Table 5. In summary, the script identification task on the benchmark dataset required a total of 4 classifiers in two levels as shown in Fig. 12.

Table 5 Features used for classifier-B at various levels

Feature set	Features used	Classifier
1	Eight $\Delta E_{\theta i}$ values $\overline{\Delta E}$, $\overline{E}$	C1
2	R(1,5); R(8,3); R(5,8)	C2
3	R(5,1); R(5,8); R(2,7)	C3

4	R(2,6); R(5,7); R(8,6)
	R(3,7); R(3,1)
		C4



Arabic Armenian Burmese Chinese Cyrillic

Devanagari Ethiopic Greek Hebrew Japanese

Korean Latin (Roman) Thai

Fig. 11 Energy profiles of individual scripts from the benchmark dataset

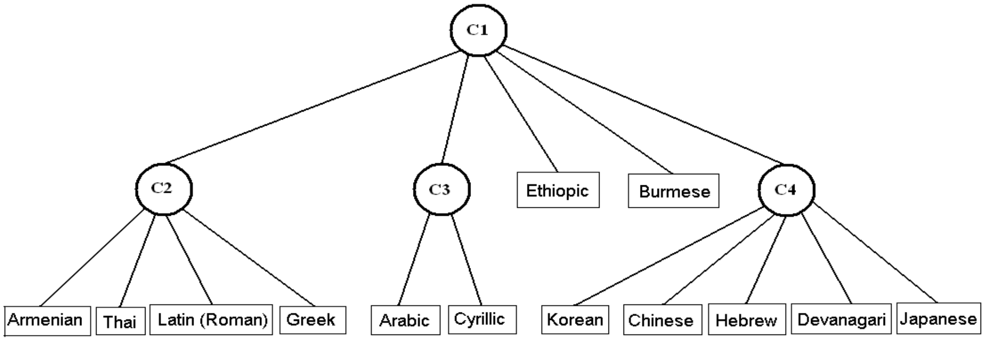


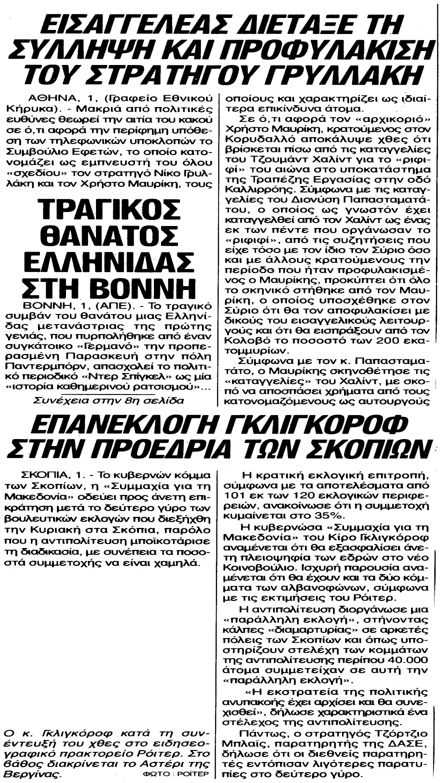
Fig. 12 Classifier-B for the benchmark dataset

6.3 Performance analysis on benchmark dataset

We have used k-nn classifier at the individual nodes of the classification scheme because of its superior performance on the test dataset (Indian script). A total of 355 text blocks (of size 400×400), selected at random from 7 images/script class, were used to train the four classifiers. Specifically, the root node classifier was trained with 355 text blocks, whereas, the classifiers at the second level were trained with a smaller subset($C2$: 50 ,$C3$: 110, and $C4$: 139). Testing on a total of 3791 text blocks of same size as for the training set, we obtained an accuracy of 91.6%.

It was found that most of the misclassified text blocks contained at best 5% text. Samples of mis-classified text blocks are shown in the Fig. 14. The paucity of textual content results in insufficient information in the energy profile. In the case of Indian scripts, there are very few such cases because blank regions along the borders of the image are manually removed. In contrast, the benchmark dataset images not only contain blank regions in the border but also in between text regions.

To address the above problem, we experimented with using a selection criterion for the blocks before inclusion in the training/testing data. A simple criterion, based on the density of the black pixels (black on white) in a given text block was used. Thus, if N_w and N_b are the number of white and black pixels in the block, respectively, the requirement is: $N_b/(N_w + N_b) > 0.05$. This factor is empirically chosen based on the obtained misclassified blocks. On imposing this selection, the accuracy of identification increased to 94.6%. The root clas-



ΕΙΣΑΓΓΕΛΕΑΣ ΔΙΕΤΑΞΕ ΤΗ ΣΥΛΛΗΨΗ ΚΑΙ ΠΡΟΦΥΛΑΚΙΣΗ ΤΟΥ ΣΤΡΑΤΗΓΟΥ ΓΡΥΛΛΑΚΗ

ΑΘΗΝΑ, 1, (Γραφείο Εθνικού Κήρυκα). - Μακριά από πολιτικές ευθύνες θεωρεί την αιτία του κακού σε ό,τι αφορά την περίφημη υπόθεση των τηλεφωνικών υποκλοπών το Συμβούλιο Εφετών, το οποίο κατονομάζει ως εμπνευστή του όλου «σχεδίου» τον στρατηγό Νίκο Γρυλλάκη και τον Χρήστο Μαυρίκη, τους οποίους και χαρακτηρίζει ως ιδιαίτερα επικίνδυνα άτομα.

Σε ό,τι αφορά τον «αρχικοριό» Χρήστο Μαυρίκη, κρατούμενος στον Κορυδαλλό αποκάλυψε χθες ότι βρίσκεται πίσω από τις καταγγελίες του Τζουμάντ Χαλίντ για το «ριφιφί» του αιώνα στο υποκατάστημα της Τραπέζης Εργασίας στην οδό Καλλιρρόης. Σύμφωνα με τις καταγγελίες του Διονύση Παπασταματάτου, ο οποίος ως γνωστόν έχει καταγγελθεί από τον Χαλίντ ως ένας εκ των πέντε που οργάνωσαν το «ριφιφί», από τις συζητήσεις που είχε τόσο με τον ίδιο τον Σύριο όσο και με άλλους κρατούμενους την περίοδο που ήταν προφυλακισμένος ο Μαυρίκης, προκύπτει ότι όλο το σκηνικό στήθηκε από τον Μαυρίκη, ο οποίος υποσχέθηκε στον Σύριο ότι θα τον αποφυλακίσει με δικούς του εισαγγελικούς λειτουργούς και ότι θα εισπράξουν από τον Κολοβό το ποσοστό των 200 εκατομμυρίων.

Σύμφωνα με τον κ. Παπασταματάτο, ο Μαυρίκης σκηνοθέτησε τις «καταγγελίες» του Χαλίντ, με σκοπό να αποσπάσει χρήματα από τους κατονομαζόμενους ως αυτουργούς

ΤΡΑΓΙΚΟΣ ΘΑΝΑΤΟΣ ΕΛΛΗΝΙΔΑΣ ΣΤΗ ΒΟΝΝΗ

ΒΟΝΝΗ, 1, (ΑΠΕ). - Το τραγικό συμβάν του θανάτου μιας Ελληνίδας μετανάστριας της πρώτης γενιάς, που πυρπολήθηκε από έναν συγκάτοικο «Γερμανό» την προπεπερασμένη Παρασκευή στην πόλη Παντερμπόρν, απασχολεί το πολιτικό περιοδικό «Ντερ Σπίγκελ» ως μία «ιστορία καθημερινού ρατσισμού»...

Συνέχεια στην 8η σελίδα

ΕΠΑΝΕΚΛΟΓΗ ΓΚΛΙΓΚΟΡΟΦ ΣΤΗΝ ΠΡΟΕΔΡΙΑ ΤΩΝ ΣΚΟΠΙΩΝ

ΣΚΟΠΙΑ, 1. - Το κυβερνών κόμμα των Σκοπίων, η «Συμμαχία για τη Μακεδονία» οδεύει προς άνετη επικράτηση μετά το δεύτερο γύρο των βουλευτικών εκλογών που διεξήχθη την Κυριακή στα Σκόπια, παρόλο που η αντιπολίτευση μποϊκοτάρισε τη διαδικασία, με συνέπεια τα ποσοστά συμμετοχής να είναι χαμηλά.

Ο κ. Γκλιγκόροφ κατά τη συνέντευξή του χθες στο ειδησεογραφικό πρακτορείο Ρόιτερ. Στο βάθος διακρίνεται το Αστέρι της Βεργίνας. ΦΩΤΟ : ΡΟΪΤΕΡ

Η κρατική εκλογική επιτροπή, σύμφωνα με τα αποτελέσματα από 101 εκ των 120 εκλογικών περιφερειών, ανακοίνωσε ότι η συμμετοχή κυμαίνεται στο 35%.

Η κυβερνώσα «Συμμαχία για τη Μακεδονία» του Κίρο Γκλιγκόροφ αναμένεται ότι θα εξασφαλίσει άνετη πλειοψηφία των εδρών στο νέο Κοινοβούλιο. Ισχυρή παρουσία αναμένεται ότι θα έχουν και τα δύο κόμματα των αλβανοφώνων, σύμφωνα με τις εκτιμήσεις του Ρόιτερ.

Η αντιπολίτευση διοργάνωσε μια «παράλληλη εκλογή», στήνοντας κάλπες «διαμαρτυρίας» σε αρκετές πόλεις των Σκοπίων και όπως υποστηρίζουν στελέχη των κομμάτων της αντιπολίτευσης περίπου 40.000 άτομα συμμετείχαν σε αυτή την «παράλληλη εκλογή».

«Η εκστρατεία της πολιτικής ανυπακοής έχει αρχίσει και θα συνεχισθεί», δήλωσε χαρακτηριστικά ένα στέλεχος της αντιπολίτευσης.

Πάντως, ο στρατηγός Τζόρτζιο Μπλαϊς, παρατηρητής της ΔΑΣΕ, δήλωσε ότι οι διεθνείς παρατηρητές εντόπισαν λιγότερες παρατυπίες στο δεύτερο γύρο.

Fig. 13 Sample document image from the difficult image set

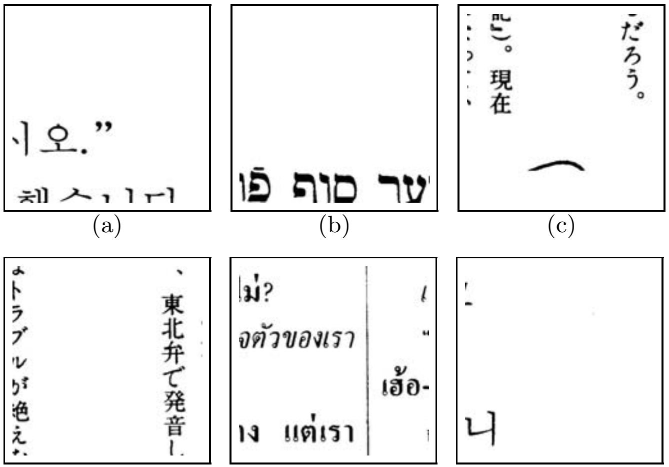


Fig. 14 Some misclassified samples. Misclassification is due to insufficient textual content

sifier $C1$ contributes an error of 2.8% in the overall error (5.4%). A confusion matrix for the proposed classification scheme is given in Table. 6. The diagonal term indicates the number 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 Burmese/Japanese and Korean/Chinese. The reason for this is not apparent from either the text block or their energy profiles and needs further investigation.

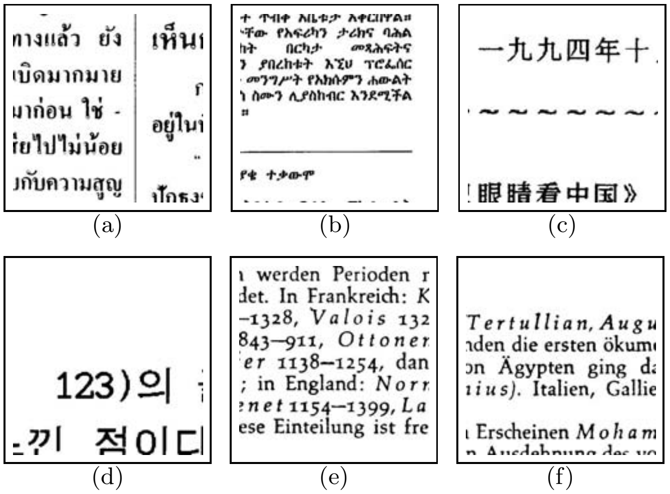


Fig. 15 Samples of misclassification after eliminating non-textual blocks.

After eliminating blocks containing insufficient text, mis-classifications were analysed again. Samples of such text blocks are shown in the Fig. 15. It was observed that error is mainly due to the presence of horizontal/vertical lines, extraneous characters such as numerals, tilde, etc., all of which distort the energy profile, a key feature for $C1$. The last two images in the second row indicate that mixed fonts pose a challenge for script identification.

The performance reported so far is only for text blocks. However, this can be extended to identify a script class from the given document as follows: a divide a given document image into a number of text blocks and assign each block to one script class using the proposed scheme; decide the final script class for the document image based on the majority script class assigned to the blocks. The effect of mis-classified blocks, due to blank regions in between the textual region and other illustrations, on the assignment of a script class for the entire document is thus reduced. Hence, reliable script identification of complex document images is also possible. We have tested this method on both the difficult and test image sets provided in the benchmark dataset. Of the 73 images in the former, 7 were misclassified. The images present in the difficult set were not part of the training set. A sample from the difficult image set can be seen in Fig. 13. The sources of errors were probably due to some text blocks, which were automatically extracted, containing mixed font sizes and the vertical lines. In general, difficult images are the representative set of images which pose a challenge to any script classification scheme. Of the 195 test document images, 13 were misclassified resulting in an accuracy of 93.3%. The above results have been obtained by using 355 textblocks for training. Since the benchmark data set consists of images from multiple

Table 6 Confusion Matrix for classifier-B. (Here, Eth=Ethiopic, Ara=Arabic, Cyr= Cyrillic, Kor= Korean, Chi= Chinese, Heb=Hebrew, Dev= Devanagari, Jap= Japanese, Arm= Armenian, Tha= Thai, Lat=Latin, Gre= Greek, Bur= Burmese.)

	Classified												
Actual	Eth	Ara	Cyr	Kor	Chi	Heb	Dev	Jap	Arm	Tha	Lat	Gre	Bur
Eth	131				1								
Ara		95		2	1						2		
Cyr		13	325									2	
Kor		2		259	15		1	1	2	1	1		
Chi		2		9	433			2		1	1	1	
Heb		4	5	3		213	4				1		
Dev							162						
Jap		3	7		7			204				1	
Arm									277				
Tha									4	342	3		
Lat		1	5						1	12	334	1	
Gre		1	5					1		3	6	252	
Bur								15			1		108

sources, we repeated the test after increasing the training set size to 605 textblocks covering multiple sources. As a result, the accuracy increased to 98.5% on test images. This illustrates the importance of selecting the right size of training dataset in classification.

It is difficult to compare the performance of our scheme with the performance reported in [2] since the available information on which images were used for the training versus the test sets, is insufficient. Furthermore, the reported template-based approach uses a screening process to pre-select the test data set using the reliability of match between an extracted symbol from a test image and the set of templates. This can potentially bias the results.

7 Discussion and Conclusion

Based on our observation of the human ability to classify unfamiliar scripts, we have presented a new representation for the script images in terms of line textures and have proposed a generalised framework for script identification. The approach helps address the basic problem of feature selection effectively with the use of global analysis and hierarchical classification. The use of global features, such as energy features, help broadly classify the scripts into homogeneous script classes. This can be further refined in successive levels using script-specific information. In general, selecting appropriate features for a large set of script classes is much more difficult than for a small set. The proposed approach aids in selecting features for small sets of script classes and hence is attractive.

The proposed framework has the capability to address factors involved in script identification such as feature selection, performance, scope for various applications etc. Some global features have been identified with discrimination capability across a large class of scripts and hence, are useful for script identification. Solutions for hierarchical classification of two different datasets (testbed and benchmark) were also presented. The testbed consisted of all Indian scripts whereas, the benchmark dataset contains representative scripts of Asia and Europe. The benchmark dataset is the only available multi-script dataset to our knowledge, and was chosen for its wide coverage of world (all continents) scripts. There were some notable differences between the two datasets: the world script dataset contained text oriented both horizontally and vertically (Japanese and Korean) unlike the Indian script dataset and the font size varied within a document image. Despite these differences, the obtained performance on the two datasets was good and comparable.

In real-life scenario, document images can be sourced in a variety of ways, each of which introduces some challenges to script identification. Some of these are image quality, skew, font variation, illustration etc. A common solution to handle these difficulties is via pre-processing. In our framework, pre-processing can be used to suppress illustrations and lines prior to feature extraction. For instance, texture based segmentation methods such as [17] [19] can be readily employed to perform automatic removal of illustrations. The horizontal or vertical lines, such as shown in Figures 14 (*e*) and 15 (*b*), can be eliminated from a *text block* by locating a peak point in the horizontal or vertical profiles. A skew correction method such as [20] can be used to correct the document images. The robustness of the framework to document script variations like font size and type depends on the extracted features. The global features that we have identified, namely, normalised energy are generally invariant to font size variations across text blocks (but not within) and to document noise. Font style variations like thickness due to bold font can be handled by applying a thinning operation before extracting features.

The given framework, based on a global approach, has scope for a wide range of applications such as selection of appropriate OCR module, archival and image retrieval systems. It does not require a common LWC segmentation method which works across large script classes; this is particularly useful in situations where a large number of multilingual document images have to be processed. Our experiments demonstrate that global approaches can handle the problem of script identification in an effective manner.

References

1. A. Spitz., Determination of the script and language content of document images. IEEE Trans. Pattern Anal. Mach. Intell.**19(3)** 235–245 (1997)
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)** 176–181 (1997)
3. X. Zhitao, G. Chengming, Y. Ming, and L. Qiang, Research on log Gabor wavelet and its application in image edge detection. Proc. of 6th International Conference on Signal Processing**1)** 592–595 (2002)
4. 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** 428–431 (1995)
5. T. N. Tan., Rotation invariant texture features and their use in automatic script identification. IEEE Trans. Pattern Anal. Mach. Intell. **20(7)** 751–756 (1998)
6. A. Busch, W. W. Boles, and S. Sridharan, Texture for script identification. IEEE Trans. Pattern Anal. Mach. Intell. **27(11)** 1720–1732 (2005)
7. U. Pal, S. Sinha, and B. B. Chaudhuri., Multi-script line identification from Indian document. 7th International Conference on Document Analysis and Recognition **2** 880–884 (2003)
8. S. Chaudhury and R. Sheth., Trainable script identification strategies for Indian languages. 5th International Conference on Document Analysis and Recognition 657–660 (1999)
9. W. Chan and J. Sivaswamy., Local energy analysis for text script classification. Proceedings of Image and Vision Computing New Zealand (1999)
10. W. Chan and G. G. Coghill., Text analysis using local energy. Pattern Recognition **34(12)** 2523–2532 (2001)
11. A. M. Namboodiri and A. K. Jain, Online handwritten script recognition. IEEE Trans. Pattern Anal. Mach. Intell. **26(1)** 124–130 (2004)
12. Digital Library of India. http://dli.iiit.ac.in/.
13. LIFI: Language Identification From Images. http://www.c3.lanl.gov/ kelly/LIFI/
14. Samachar. http://www.samachar.com/.
15. M. C. Morrone and D. C. Burr, Feature detection in human vision: A phase-dependent energy model. Proceedings of the Royal Society, London Series B **235** 221–245 (1988)
16. PRTools: A Matlab Toolbox for Pattern Recognition. http://www.prtools.org/.
17. A. K. Jain and Y. Zhong., Page segmentation using texture analysis. Pattern Recognition **29** 743–770 (1996)
18. R. Duda, P. Hart, and D. Stork., Pattern Classification. second edition, New York: John Wiley and Sons (2001)
19. T. Randen and J. H. Husy., Segmentation of text/image documents using texture approaches. Proceedings of NOBIM-Konferansen-94 60–67 (1994)
20. C. Sun and D. Si , Skew and slant correction for document images using gradient direction. Proceedings of Document Analysis and Recognition **1** 142–146 (1997)

Gopal Datt Joshi biography will be printed here if needed. Whether an author biography is included per author is set per journal. A photo is optional.

Saurabh Garg biography will be printed here if needed. Whether an author biography is included per author is set per journal. A photo is optional.

Jayanthi Sivaswamy biography will be printed here if needed. Whether an author biography is included per author is set per journal. A photo is optional.