Script Recognition—A Review

Debashis Ghosh, Tulika Dube, and Adamane P. Shivaprasad

Abstract—A variety of different scripts are used in writing languages throughout the world. In a multiscript, multilingual environment, it is essential to know the script used in writing a document before an appropriate character recognition and document analysis algorithm can be chosen. In view of this, several methods for automatic script identification have been developed so far. They mainly belong to two broad categories—structure-based and visual-appearance-based techniques. This survey report gives an overview of the different script identification methodologies under each of these categories. Methods for script identification in online data and video-texts are also presented. It is noted that the research in this field is relatively thin and still more research is to be done, particularly in the case of handwritten documents.

Index Terms—Document analysis, optical character recognition, script identification, multiscript document.

1 Introduction

One interesting and challenging field of research in pattern recognition is Optical Character Recognition (OCR). Optical character recognition is the process in which a paper document is optically scanned and then converted into computer processable electronic format by recognizing and associating symbolic identity with every individual character in the document.

With the increasing demand for creating a paperless world, many OCR algorithms have been developed over the years [1], [2], [3], [4], [5], [6]. However, most OCR systems are script specific in the sense that they can read characters written in one particular script only. *Script* is defined as the graphic form of the writing system used to write statements expressible in language. This means that a script class refers to a particular style of writing and the set of characters used in it. Languages throughout the world are typeset in many different scripts. A script may be used by only one language or may be shared by many languages, sometimes with slight variations from one language to other. For example, Devnagari is used for writing a number of Indian languages like Sanskrit, Hindi, Konkani, Marathi, etc., English, French, German, and some other European languages use different variants of the Latin alphabet, and so on. Some languages even use different scripts at different point of time and space. One good example for this is Malay, which uses the Latin alphabet nowadays, replacing the previously used Jawi. Another example is Sanskrit, which is mainly written in Devnagari in India but is also written in Sinhala script in Sri Lanka. Therefore, in this multilingual and multiscript world, OCR systems need to be capable of recognizing characters irrespective of the script in which they are written. In general, recognition of different script characters in a single OCR module is difficult. This is because the features necessary for character recognition depend on the structural properties, style, and nature of writing, which generally differs from one script to another. For example, features used for recognition of English alphabets are, in general, not good for recognizing Chinese logograms.

Another option for handling documents in a multiscript environment is to use a bank of OCRs corresponding to all different scripts expected to be seen. The characters in an input document can then be recognized reliably by selecting the appropriate OCR system from the OCR bank. Nevertheless, this will require knowing a priori the script in which the input document is written. Unfortunately, this information may not be readily available. At the same time, manual identification of the documents' scripts may be tedious and time-consuming. Therefore, automatic script recognition techniques are necessary to identify the script in the input document and then redirect it to the appropriate character recognition module, as illustrated in Fig. 1.

A script recognizer is also useful in reading multiscript documents in which different paragraphs, text blocks, textlines, or words in a page are written in different scripts. Fig. 2 shows several examples of multiscript documents. Analysis of such documents works in two stages—identification and separation of different script regions in the document, followed by reading of each individual script region using corresponding OCR system.

Script identification also serves as an essential precursor for recognizing the language in which a document is written. This is necessary for further processing of the document, such as routing, indexing, or translation. For scripts used by only one language, script identification itself accomplishes language identification. For scripts shared by many languages, script recognition acts as the first level of classification, followed by language identification within the script.

Script recognition also helps in text area identification, video indexing and retrieval, and document sorting in

- *D. Ghosh is with the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247 667, India. E-mail: ghoshfec@iitr.ernet.in.*
- *T. Dube is with the Indian Institute of Management Ahmedabad, Dorm 2, Room 31, Vastrapur, Ahmedabad, Gujrat 380 015, India. E-mail: 9tulikad@iimahd.ernet.in.*
- *A.P. Shivaprasad is with the Department of Electronics and Communication Engineering, Sambhram Institute of Technology, 915/33, 7th Cross, 13th Main, Mathikere, Bangalore, Karnataka 560 054, India. E-mail: apshivaprasad@yahoo.com.*

Manuscript received 6 July 2007; revised 16 Apr. 2008; accepted 22 July 2009; published online 21 Jan. 2010.
Recommended for acceptance by D. Lopresti.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-07-0408.
Digital Object Identifier no. 10.1109/TPAMI.2010.30.

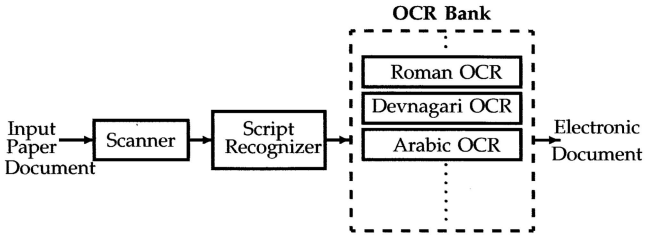
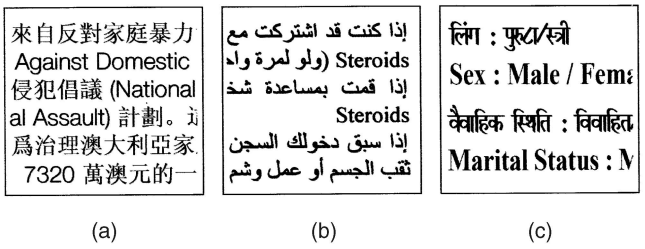


Fig. 1. Stages of document processing in a multiscript environment.



(a) (b) (c)

Fig. 2. Examples of multiscript document images: (a) a government report in China containing a mix of Chinese and English words, (b) a medical report in Arabic containing words in English that do not have an exact Arabic equivalent, (c) a portion of an official application form in India containing different script lines typeset in Hindi and English.

digital libraries when dealing with a multiscript environment. Text area detection refers to either segmenting out text blocks from other nontextual regions, like halftones, images, line drawings, etc., in a document image, or extracting text printed against textured backgrounds and/or embedded in images within a document. To do this, the system takes advantage of script-specific distinctive characteristics of text which make it stand out from other nontextual parts in the document. Text extraction is also required in images and videos for content-based browsing. One powerful index for image/video retrieval is the text appearing in them. Efficient indexing and retrieval of digital image/video in an international scenario therefore requires text extraction followed by script identification and then character recognition. Similarly, text found in documents can be used for their annotation, indexing, sorting, and retrieval. Thus, script identification plays an important role in building a digital library containing documents written in different scripts.

In short, automatic script identification is crucial to meet the growing demand for electronic processing of volumes of documents written in different scripts. This is important for business transactions across Europe and the Orient, and has great significance in a country like India, which has many official state languages and scripts. Due to this, there has been a growing interest in multiscript OCR technology during recent years. A brief survey on methods for script recognition was reported earlier in [7], with emphasis on script identification in Indian multiscript documents but little insight into the script recognition methods for non-Indian scripts. A review of script identification research for Indian documents is also available in [8]. A report on the key technologies in multilingual OCR and their application in building a multilingual digital library can also be found in [9].

In this paper, we present a comprehensive survey of different script recognition techniques developed mainly for identification of certain major scripts of the world, viz., Chinese, Japanese, Korean, Arabic, Hebrew, Latin, Cyrillic, and the Brahmic family of Indian scripts. To begin with, in Section 2, we give a brief description of different script types, highlighting their main discriminating features. Methods for script recognition in document images are described in Section 3, giving comparative analysis among them. Section 4 discusses several methods for script recognition in the realm of pen computing. As said before, script identification in video text is also important. However, not much research has been done on this topic. The only work that we have found on this is outlined in Section 5. Section 6 raises issues related to performance evaluation of multiscript OCR systems. Finally, we state our concluding remarks in Section 7, including some insights on the recent trends and future scope of work in this field.

2 Writing Systems and Scripts of the World

In the context of script recognition, it may be worth studying the characteristics of various writing systems and the structural properties of the characters used in certain major scripts of the world. In Fig. 3, we draw a tree diagram showing different classes of writing systems. As said in [10], [11] and depicted in the tree diagram, there are six prominent writing systems. Major scripts that follow each of these writing systems are also shown in the tree diagram and are described below.

2.1 Logographic System

A logogram, also called an *ideogram*, refers to a symbol that graphically represents a complete word. Accordingly, the number of characters in a script for an ideographic writing system generally runs into thousands. This makes recognition of logographic characters a difficult but interesting problem.

An example of logographic script is *Han*, which is mainly associated with Chinese. Japanese and Korean writings also include Han modified as *Kanji* and *Hanja*, respectively. Han characters are generally composed of multiple short strokes, giving them a complex and dense look, distinctly different from other Western and Asian scripts. Accordingly, character optical density and certain other visual appearance-based features have been utilized by many researchers in distinguishing Han from other scripts. Another interesting property of Han is its directionality—words in a textline are written either from left to right or from top to bottom.

2.2 Syllabic System

In a syllabic system, every written symbol represents a phonetic sound or syllable, as used in Japanese. The symbols representing the Japanese syllables are known as *Kanas*, which are of two types—*Hirakana* and *Katakana*. As indicated in Fig. 3, Japanese script uses a mix of logographic Kanji and syllabic Kanas. Hence, it is visually similar to Chinese, but less dense due to the presence of simpler Kanas in between the logograms.

2.3 Alphabetic System

An alphabet is a set of characters representing phonemes of a spoken language. Examples of scripts following this system

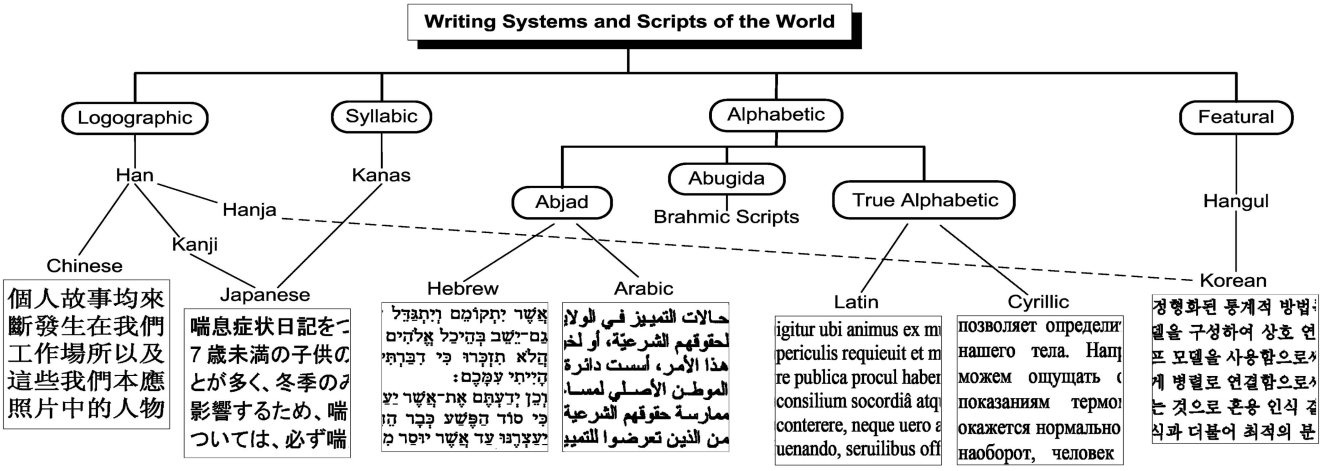


Fig. 3. Tree diagram showing broad classification of prominent writing systems and scripts of the present world.

are Greek, Latin, Cyrillic, and Armenian. The *Latin* script, also called Roman script, is used by many languages throughout the world with varying degrees of modifications from one language to another. It is used for writing many European languages like English, Italian, French, German, Portuguese, Spanish, etc., and has been adopted in many Amerindian and Austronesian languages, including the modern Malay, Vietnamese, and Indonesian languages. Fig. 4 shows a few such variants of the Latin script. Compared to other scripts, classical Latin characters are simple in structure, mainly composed of a few lines and arcs. The other major script under the alphabetic system is *Cyrillic*. This script is used by some languages of Eastern Europe, Asia, and Slavic regions that include Bulgarian, Russian, Macedonian, Ukrainian, Mongolian, etc. The basic properties of this script are somewhat similar to that of Latin except that it uses a different alphabet set. Some characters in the Cyrillic alphabet are also borrowed from Latin and Greek, modified with cedillas, crosshatches, or diacritical marks. This induces recognition ambiguity among Cyrillic, Latin, and Greek.

2.4 Abjads

The Abjad system of writing is similar to the alphabetic system, but has symbols for consonantal sounds only. Unlike most other scripts in the world, Abjads are written from right to left within a textline. This unique feature is particularly useful for identifying Abjad-based scripts in pen computing.

Two important scripts under this category are *Arabic* and *Hebrew*. A typical Arabic character is formed of a long main stroke along with one to three dots. The characters in a word are generally conjoined, giving an overall cursive appearance to the written text. This provides an important clue for the recognition of Arabic script. The same applies to some other scripts of Arabic origin, such as Farsi (Persian), Urdu, Sindhi, Jawi, etc. On the other hand, character strokes in Hebrew are more uniform in length and the letters in a word are generally discrete.

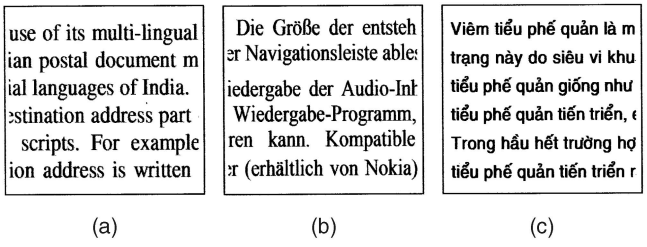


Fig. 4. Examples of some languages using the Latin alphabet with different modifications. (a) English. (b) German. (c) Vietnamese.

2.5 Abugidas

Abugida is another alphabetic-like writing system used by the *Brahmic* family of scripts that originated from the ancient Indian Brahmi script and includes nearly all of the scripts of India and southeast Asia. In Fig. 5, we draw a tree diagram to illustrate the evolution of major Brahmic scripts in India and southeast Asia. The northern group of Brahmic scripts (e.g., Devnagari, Bengali, Manipuri, Gurumukhi, Gujrati, and Oriya) bears a strong resemblance to the original Brahmi script. On the other hand, scripts in south India (Tamil, Telugu, Kannada, and Malayalam) as well as in southeast Asia (e.g., Thai, Lao, Burmese, Javanese, and Balinese) are derived from Brahmi through many changes and so look quite different from the northern group. One important characteristic of Devnagari, Bengali, Gurumukhi, and Manipuri is that the characters in a word are generally written together without spaces so that the top bar is unbroken. This results in the formation of a headline, called *shirorekha*, at the top of each word. Accordingly, these scripts can be separated from other script types by detecting the presence of a large number of horizontal lines in the textual portions of a document.

2.6 Featural System

The last significant form of writing system is the featural system in which the symbols or characters represent the features that make up the phonemes. One prominent script of this sort is the Korean *Hangul*. As indicated in Fig. 3, the Korean script is formed by mixing logographic Hanja with featural Hangul. However, modern Korean contains more of Hangul than Hanja. Consequently, the Korean script is relatively less complex and less dense compared to the Chinese and Japanese, containing more circles and ellipses.

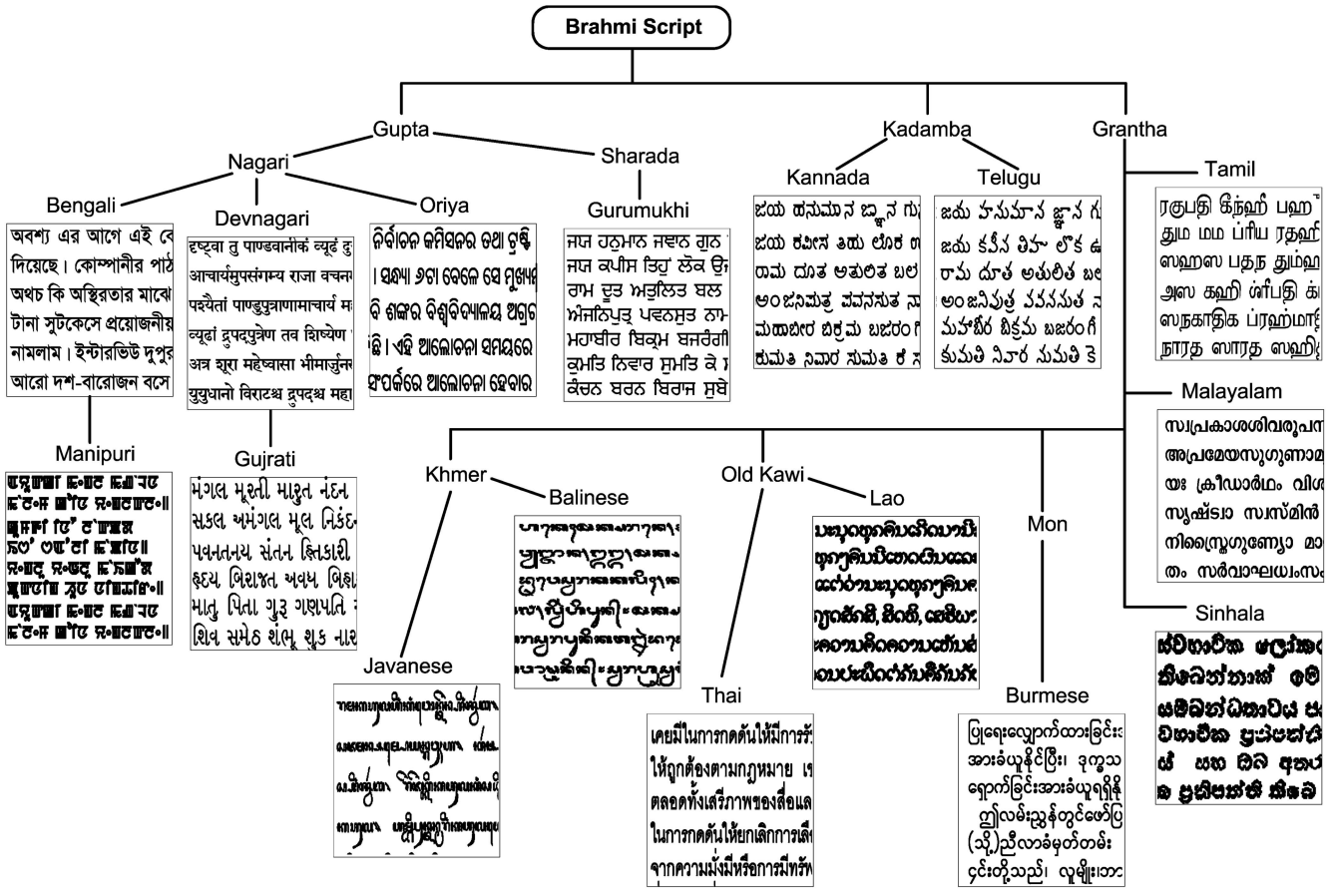


Fig. 5. The Brahmic family of scripts used in India and southeast Asia.

3 Script Recognition Methodologies

Script identification relies on the fact that each script has unique spatial distribution and visual attributes that make it possible to distinguish it from other scripts. So, the basic task involved in script recognition is to devise a technique to discover these features from a given document and then classify the document's script accordingly. Based on the nature of approach and features used, these methods may be divided into two broad categories—structure-based and visual appearance-based methods. Script recognition techniques in each of these two categories may be further classified on the basis of the level at which they are applied inside a document image, viz., pagewise, paragraphwise, textlinewise, and wordwise. The application mode of a method depends on the minimum size of the text from which the features proposed in the method can be extracted reliably. Various algorithms under each of these categories are summarized below.

3.1 Structure-Based Script Recognition

In general, script classes differ from each other in their stroke structure and connections and the writing styles associated with the character sets they use. One approach to script recognition may be to extract connected components (continuous runs of pixels) in a document [12] and then analyze their shapes and structures so as to reveal the intrinsic morphological characteristics of the script used in the document. In machine-printed Latin, Greek, Han, etc., every individual character or part of a character is a connected component. On the other hand, in cursive handwritten documents, the characters in a word or part of a word can touch each other to form one single connected component. Likewise, in scripts like Devnagari, Bengali, Arabic, etc., a word or a part of a word forms a connected component. Script identification methods that are based on extraction and analysis of connected components fall under the category of structure-based methods.

3.1.1 Pagewise Script Identification Methods

A script identification method that relies on the spatial relationship of character structures was developed by Spitz for differentiating Han and Latin scripts in machine-printed documents. In his first work on this topic [13], he used character optical density for classifying individual textlines in a document as being either English or Japanese. In another paper, Spitz used vertical distribution of upward concavities in characters for discriminating Han from Latin with 100 percent success in continuous production use [14]. Later, he developed a two-stage classifier in [15] by combining these two features. In the first stage, Latin is separated from Han-based scripts by comparing the variances of their upward concavity distributions. Further classification within the Han-based scripts is performed by analyzing the distribution of optical density in the text image. The system also has provisions for language identification within documents using the Latin alphabet by observing the most frequently occurring character shape codes. A schematic diagram showing the flow of information in the process is given in Fig. 6.

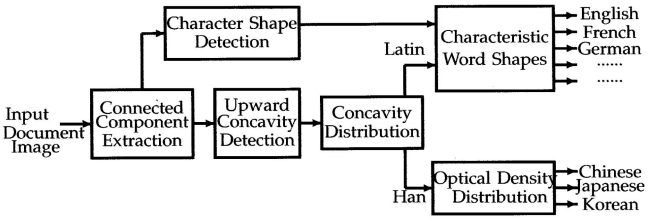


Fig. 6. Spitz's method of script identification.

The above works by Spitz were extended by Lee et al. [16] and Waked et al. [17] by incorporating some additional features. In [16], the script of a printed document is identified via textlinewise script recognition, followed by a majority vote of the already decided textline classification results. The features used are character height distribution and the top and bottom profiles of character bounding boxes, in addition to upward concavity distribution and optical density features. Experimental results showed that these features can separate Han-based (Chinese and Japanese) documents from Latin-based (English, French, German, Italian, and Spanish) documents in 98.16 percent of cases. In [17], Waked et al. used bounding box size distribution, character density distribution, and horizontal projections for classifying printed documents written in Han, Latin, Cyrillic, and Arabic. These statistical features are more robust compared to the structural features proposed by Spitz and Lee et al. However, Waked et al. achieved an accuracy rate of only 91 percent when tested on documents of varying kinds, diverse formats, and qualities. This drop in recognition accuracy is mainly due to the misclassification between Latin and Cyrillic scripts, which are similar looking under this measure. Also, some test documents of extremely poor quality account for this degradation in performance.

Script identification in machine-printed documents using statistical features has also been explored by Lam et al. [18]. In a first level of classification, documents are classified as Latin, Chinese, Japanese, or Korean using horizontal projection profiles, height distributions of connected components, and enclosing structure of connected components. Non-Latin documents that cannot be recognized in this stage are classified in a second level of recognition using structural features like character complexity, presence of circles, ellipses, and vertical strokes. In the process, more than 95 percent correct recognition was achieved.

The fact that every script class is composed of some "textual symbols" of unique characteristic shapes had been exploited by Hochberg et al. in identifying the script of a printed document [19]. First, textual symbols obtained from documents of a known script are resized and clustered to generate template symbols for that script class, as depicted in Fig. 7. Textual symbols include character fragments, discrete characters, adjoined characters, and even whole words. During classification, textual symbols extracted from the input document are compared to the template symbols using Hamming distance and then scored against every script class on the basis of their distances from the best match template symbols in that script class. The script class with the best average score is chosen as the script of the document. Hochberg et al. tested their method on as many as 13 scripts, viz., Arabic, Armenian, Burmese, Chinese,

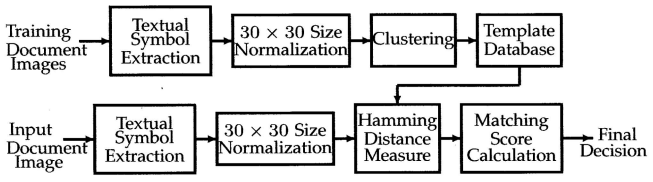


Fig. 7. Hochberg et al.'s method of script identification in printed documents.

Cyrillic, Devanagari, Ethiopic, Greek, Hebrew, Japanese, Korean, Latin, and Thai, and obtained 96 percent accuracy.

Hochberg et al. [20] proposed a feature-based approach for script identification in handwritten documents and achieved 88 percent accuracy in distinguishing Arabic, Chinese, Cyrillic, Devnagari, Japanese, and Latin. In their method, a handwritten document is characterized in terms of mean, standard deviation, and skew of five features, which are relative vertical centroid, relative horizontal centroid, number of holes, sphericity, and aspect ratio, of the connected components in a document page. A set of Fisher Linear Discriminants (FLDs), one FLD for every pair of script classes, is used for classification. The document is finally assigned to the script class to which it is classified most often. A schematic diagram showing different stages of the system is given in Fig. 8.

A novel approach to script identification using fractal features was proposed in [21] and had been utilized for discriminating printed Chinese, Japanese, and Devnagari scripts. Fractal features are obtained by computing fractal signatures for the patterns extracted from a document image. The fractal signature is determined by the area of the surface onto which a gray-level function corresponding to the document image is mapped.

A method for script identification in printed document images based on morphological reconstruction was proposed in [22]. In this method, morphological erosion and opening by reconstruction is carried out on the document image in horizontal, vertical, right, and left diagonal directions using line structuring elements. The average pixel distributions in the resulting images give the measures of horizontal, vertical, 45, and 135 degree slanted lines present in the document page. Finally, script identification is carried out using nearest neighbor classification. The method showed robustness with respect to noise, font sizes, and styles, and an average classification accuracy of 97 percent was achieved when applied for classification of four script classes, viz., Latin, Devnagari, Urdu, and Kannada.

3.1.2 Script Identification at Paragraph and Text Block Level

The script identification methods discussed above require large blocks of input text so that sufficient information is

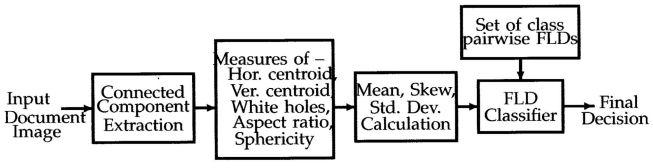


Fig. 8. Hochberg et al.'s method of script identification in handwritten documents.

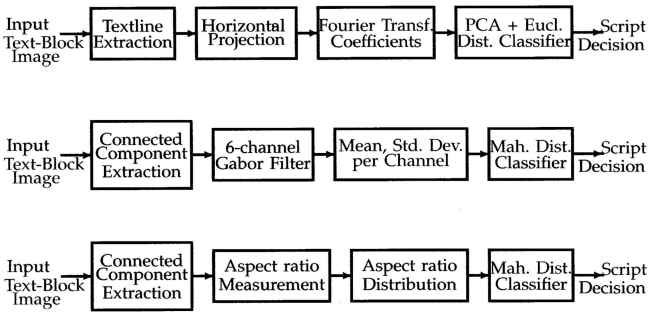


Fig. 9. Chaudhury and Sheth's three methods of script identification.

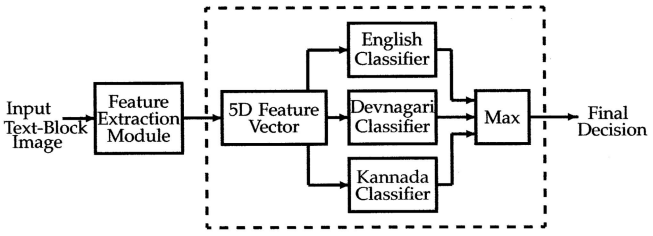


Fig. 10. Neural network-based architecture for script identification proposed by Patil and Reddy.

available to bring out the characteristics of the script. They offer good performance when used for script identification at the page level, but may not retain their performance when applied on a smaller block of text. In multiscript documents, it is necessary to identify and separate different script regions like paragraph, textline, word, or even character in the document page. This is particularly important in a country like India that hosts a variety of scripts like Devnagari, Bengali, Tamil, Telugu, Kannada, Malayalam, Gujrati, Gurumukhi, Oriya, Manipuri, Urdu, Sindhi, and Latin. In view of this, several multiscript OCR systems involving more than one Indian script in a single unit have been developed [8]. Multiscript OCR systems that perform script recognition at the paragraph level are now described.

Fig. 9 shows three different strategies developed by Chaudhury and Sheth [23] to recognize the script of a text block in a printed document. In the first technique, the script of the text block is described in terms of the Fourier coefficients of the horizontal projection profile. Subsequent classification is based on euclidean distance in the eigenspace. The other two schemes are based on features derived from connected components in text blocks—one using the means and standard deviations of the outputs for a six-channel Gabor filter and the other using distribution of the width-to-height ratio of the connected components present in the document. Classification in both of these cases is accomplished using Mahalanobis distance. The average recognition rate obtained with these methods, when tested on Latin, Devnagari, Telugu, and Malayalam scripts, was approximately 85, 95, and 89 percent, respectively.

In [24], a neural network-based architecture was developed for identification of printed Latin, Devnagari, and Kannada scripts. It consists of a feature extractor followed by a modular neural network, as shown in Fig. 10. In the feature extraction stage, a feature vector corresponding to pixel distributions along specified directions is obtained via morphological operations. The modular neural network structure consists of three independently trained feed-forward neural networks, one for each of the three scripts under consideration. The input is assigned to the script class of the network, which produces maximum output. It was seen that such a system can classify English and Kannada with 100 percent accuracy, while the rate is slightly lower (97 percent) in recognizing Devnagari.

Script recognition using feed-forward neural network was also performed in [25]. The network is trained to classify an input printed text block into Han or Latin directly, without performing any feature extraction. The network consists of four layers with 49 nodes in the input layer, 15 and 20 nodes in the hidden layers, and two nodes in the output layer that correspond to the two script classes. The nodes in the input layer are fed with pixel values in a block of size 7×7 pixels. A number of sample blocks are randomly extracted from the input text block, and the script of the text block is then determined by a simple majority vote among the sampling blocks. Experiments on a number of mixed-type document images showed the effectiveness of the proposed system, yielding 92.3 and 95 percent accuracy in determining the Chinese and English texts, respectively.

A method for Arabic and Latin text block differentiation in both printed and handwritten scripts was proposed in [26]. This method is based on morphological analysis at the text block level and geometrical analysis at textline and connected component levels. Experimental evaluation of the method was carried out on two different data sets containing 400 and 335 text blocks, and the results obtained were quite promising.

In an attempt to build automatic letter sorting machines for Bangladesh post offices, an algorithm for Bengali/English script identification was developed recently [27]. The method is designed for application to both machine-printed and handwritten address blocks on envelope images. The two scripts under consideration are recognized on the basis of the aggregate distance of the pixels in the topmost and the bottommost profiles of the connected components—an English text image has these two distance measures almost equal, whereas their difference in Bengali text image is quite large. It was observed in the experiments that the accuracy of this script identification method is quite high for printed text (98 and 100 percent for English and Bengali, respectively) and, for handwritten text, the proposed approach can achieve a satisfactory accuracy of about 95 percent.

3.1.3 Textlinewise Script Identification

The earliest work we have found on textlinewise script identification in Indian documents was reported by Pal and Chaudhuri in [28]. The method uses projection profile, statistical and topological features, and stroke features for decision-tree-based classification of printed Latin, Urdu, Devnagari, and Bengali script lines. Later, they proposed an automatic system for the identification of Latin, Chinese, Arabic, Devnagari, and Bengali textlines in printed documents [29]. As depicted in Fig. 11, the headline ("shirorekha") information is used first to separate Devnagari and Bengali script lines from Latin, Chinese, and Arabic script

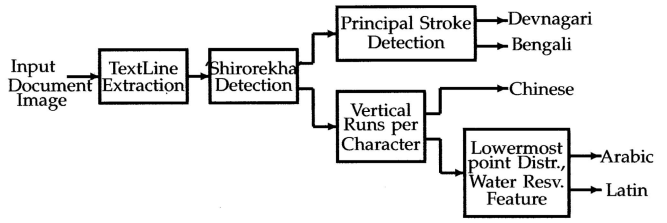


Fig. 11. Pal and Chaudhuri's method for script line separation from multiscript documents in India.

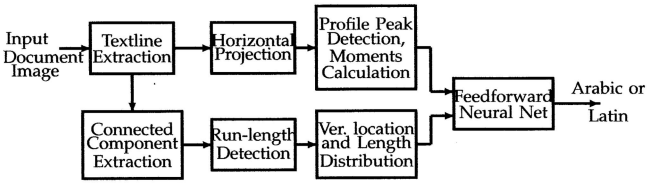


Fig. 12. Elgammal and Ismail's technique for script identification in Arabic-English documents.

lines. Next, Bengali script lines are distinguished from Devnagari by observing the presence of certain script-specific principal strokes. Similarly, Chinese textlines are identified by checking the existence of characters with four or more vertical runs. Finally, Latin (English) textlines are separated from Arabic using statistical as well as water reservoir-based features. Statistical features include the distribution of lowermost points in the characters—the lowermost points of characters in a printed English textline lie only along the baseline and the bottom line, while those in Arabic are more randomly distributed. Water reservoir-based features give a measure of the cavity regions in a character. Based on all of these structural characteristics, the identification rates obtained were, respectively, 97.32, 98.65, 97.53, 96.05, and 97.12 percent for Latin, Chinese, Arabic, Devnagari, and Bengali scripts, with an overall accuracy of 97.33 percent.

A more generalized scheme for script line identification in printed multiscript documents that can classify as many as 12 Indian scripts, viz., Devnagari, Bengali, Latin, Gujrati, Kannada, Kashmiri, Malayalam, Oriya, Gurumukhi, Tamil, Telugu, and Urdu, is available in [30]. Features chosen in the proposed method are headlines, horizontal projection profile, water reservoir-based features, left and right profiles, and feature based on jump discontinuity, which refers to the maximum horizontal distance between two consecutive border pixels in a character pattern. Experimental results show an average script line identification accuracy of 97.52 percent.

A method for discriminating Arabic text and English text using connected component analysis was proposed by Elgammal and Ismail in [31]. They tested their method on several machine-printed documents containing a mix of these two languages and achieved a recognition rate as high as 99.7 percent. Features used for distinguishing Arabic from Latin are the number of peaks and the moments in the horizontal projection profile, and the distribution of run-lengths over the location-length space. The horizontal projection profile of an Arabic textline generally has a single peak, while that of an English textline has two major peaks. Thus, Arabic script can be distinguished from Latin by detecting the number of peaks in the horizontal projection profile. The other features they used for discriminating Arabic and Latin scripts are the third and fourth central moments of the horizontal projection profiles. The third moment measures the skew, while the fourth moment measures the kurtosis that describes how flat the profile is. It is seen that the horizontal projection profile for English is more symmetric and flat compared to that of the Arabic. Therefore, the moments in the case of English text are generally smaller than those of the Arabic text. Script classification using these features is done in a two-layer feed-forward network. The basic steps of processing in this method are illustrated in Fig. 12. The algorithm was also applied for script identification at the word level and a recognition rate of 96.8 percent was achieved.

Script identification using character component n-grams was recently patented by Cumbee [32]. First, character segments extracted from training documents of a known script are clustered using K-means clustering and then replaced by their corresponding cluster identification number. Thus, every line of text is converted into a sequence of numbers. This sequence of numbers is then analyzed to determine all the n-grams present in it and a weight corresponding to the frequency of occurrence is defined for each n-gram. During recognition, n-grams are generated in a similar fashion by comparing character segments in the input textline to the K-means cluster centroids of a known script. These are then compared to the n-grams present in the training documents of that script. The input is subsequently scored against that script class by adding the weights of the best-match n-grams. The script of the input textline is determined to be the script against which it scores the highest.

3.1.4 Script Identification at Word/Character Level

Compared to the paragraph and textline-level identifications, script recognition at the word level in a multiscript document is generally more difficult. This is because the information available from only a few characters in a word may not be sufficient for the purpose. This has motivated many researchers to take up this challenging problem in script identification. Some have even attempted to do script identification at the character level. However, script recognition at the character level is generally not required in practice. This is because the script usually changes only from one word to the next and not from one character to another within a word.

In one of the earliest works on script identification at the character level, Lee and Kim tried to solve the problem using self-organizing networks [33]. The network is able to determine the script of every individual character in a machine-printed multiscript document and classify them into four groups—Latin, Chinese, Korean, and mixed. Characters in the mixed group which cannot be classified in the network with full confidence are classified in the next level of fine classification using learning vector quantization. In order to evaluate the performance of the proposed scheme, experiments with 3,367,200 characters were carried out and a recognition rate of over 98.27 percent was obtained.

An extension of Hochberg et al.'s work in [19] includes separation of different script regions in a machine-printed

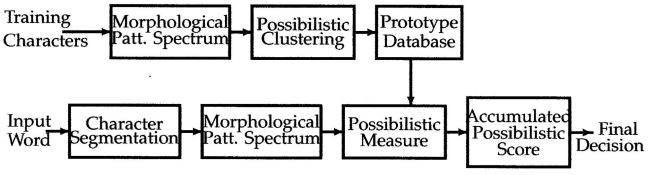


Fig. 13. Ghosh and Shivaprasad's method of script identification for handwritten characters/words using pattern spectrum and possibilistic measure.

multiscript document [34]. In this work, every textual symbol (character, word, or part of a word) in a document is matched to a set of template symbols, as in [19], and is classified to the script class of the best matching template symbol. It was observed that the method offers good separation in all cases except in the case of visually similar scripts, such as Latin/Cyrillic and Latin/Greek. The best separation was observed in visually distinct script pairs like Latin/Arabic, Latin/Japanese, and Latin/Korean.

Methods that employ clustering for generating script-specific prototype symbols, much like the procedure by Hochberg et al., were proposed in [35], [36]. In both of these methods, classification algorithms are not based on direct shape matching, as in Hochberg's method, but use matching of shape description features of connected components and/or characters. The shape description features used in [35] are the pattern spectrum coefficients of every individual character in a string of isolated handwritten characters. During training, prototype symbols for each script class are obtained via possibilistic clustering [37]. In the recognition phase, the algorithm calculates the degree to which every character in a string belongs to each of the script classes using the possibilistic measure defined in [37]. The character string is classified to that script class for which the accumulated possibilistic measure is maximum. The basic structure of the proposed system is shown in Fig. 13. The method was tested on several artificially generated [38] strings of handwritten numeric characters in four different scripts, viz., Arabic, Devnagari, Bengali, and Kannada, and a recognition rate as high as 96 percent was achieved. Ablavasky and Stevens reported a similar work [36], but for machine-printed documents. The algorithm processes a stream of connected components and assigns a script label when enough evidence has been accumulated to make the decision. The method uses geometric properties like Cartesian moments and compactness for shape description. The likelihood of every input textual symbol belonging to each of the script classes is calculated using K-nearest neighbor (KNN) classification. This approach was shown to be quite efficient, yielding 97 percent success rate in discriminating similar looking Latin and Cyrillic scripts.

In another structural approach to script identification, stroke geometry has been utilized for script characterization and identification [39]. Another new approach for identifying the script type of character images in printed documents was proposed in [40]. Individual character images in a document are classified either by applying prototype classification or by using support vector machine. Both of the methods were implemented successfully in classifying characters into Latin, Chinese, and Japanese.

Extraction of Arabic words from among a mix of printed Arabic-English words has gained attention in recent times [41], [42]. The method proposed in [41] is based on recognition of Arabic characters or character segments in a word. First, a database containing templates of Arabic character segments is generated through training. A word is supposed to be Arabic if the percentage of matching character segments in the word exceeds a user-defined value. Otherwise, the word is considered to be written in English (Latin). Experimental results showed 100 percent recognition accuracy on 30 text blocks containing a total of 478 words. The method in [42] is also based on recognition of Arabic characters in the document, but via feature matching. Features used are morphological and statistical features such as overlapping and inclusion of bounding boxes, horizontal bar, low diacritics, height and width variation of connected components, etc. Recognition accuracy achieved with this method was 98 percent.

Wordwise script identification using character shape codes was proposed by Tan et al. [43] and Lu et al. [44]. In [43], shape codes generated using basic document features like elongation of bounding boxes of character cells and the position of upward concavities are used to identify Latin, Han, and Tamil in printed document images. The method in [44] captures word shapes on the basis of local extremum points and horizontal intersections. For each script under consideration, a word shape template is first constructed based on a word shape coding scheme. Identification is then accomplished using Hamming distance between the word shape code of a query image and the previously constructed templates. Experimental tests demonstrated 99 percent recognition accuracy in discriminating eight Latin-based scripts/languages.

As noted before, multiscript document processing is important in a multiscript country such as India. Consequently, script recognition at the word level involving Indian scripts is an important topic of research for the OCR community. Indian scripts are, in general, of two types—one that has headlines ("shirorekha") on top of the characters (e.g., Devnagari, Bengali, and Gurumukhi) and the other that does not carry headlines (e.g., Gujrati, Tamil, Telugu, Malayalam, and Kannada). Based on this, a bilingual OCR for printed documents was developed in [45] that identifies Devnagari and Telugu scripts by observing the presence and absence of shirorekha. The classification result is further supported with context information; if the previous word is Devnagari (or Telugu), the next word is also in Devnagari (Telugu) unless a strong clue suggests otherwise. The proposed method was tested extensively on several Hindi-Telugu documents with recognition accuracies that vary in the range from 92.3 to 99.86 percent.

The script line identification techniques in [29], [30] were modified in [46], [47] for script word separation in printed Indian multiscript documents by including some new features, in addition to the features considered earlier. The features used are headline feature, distribution of vertical strokes, water reservoir-based features, shift below headline, left and right profiles, deviation feature, loop, tick feature, and left inclination feature. Tick feature refers to the distinct "tick"-like structure, called *telakattu*, present at the top of many Telugu characters. This helps in separating Telugu script from other scripts. Fig. 14 shows a few Telugu

Fig. 14. Examples of Telugu characters having the tick feature.

characters having this feature. The overall accuracy in script word separation using this proposed set of features was about 97.92 percent when applied to five script pairs, viz., Devnagari/Bengali, Bengali/Latin, Malayalam/Latin, Gujrati/Latin, and Telugu/Latin. Finally, based on this script word separation algorithm, systems for recognizing English, Devnagari, and Urdu [48], and English and Tamil [49] have been developed in recent years. In this context, a script word discrimination system proposed by Padma and Nagabhushan [50] also deserves mentioning. The system uses several discriminating structural features for identification and separation of Latin, Hindi, and Kannada words in Indian multiscript documents in a manner similar to the above.

The basic system for blockwise script identification in [24] was modified further so as to accomplish script recognition at the word level. The modified system architecture consists of a preprocessor that separates out individual words in a machine-printed document, followed by a modified feature extractor and a probabilistic neural network classifier. The probabilistic network is a two-layered structure composed of a radial basis layer followed by a competitive layer. Experiments yielding 98.89 percent classification accuracy demonstrate the effectiveness of such a script classification system.

A neural network structure employing script recognition at the character level in printed documents was presented in [51]. Script separation at the word level can also be achieved by combining the outputs of the character-level classification using Viterbi algorithm. The algorithm was tested on five scripts commonly used in India, namely, Latin, Devnagari, Bengali, Telugu, and Malayalam, and an average recognition accuracy of 97 percent was achieved.

MLP neural networks have also been employed for script identification in Indian postal automation systems developed by Roy et al. [52], [53], [54], [55], [56]. In India, people generally tend to write addresses either in English only or English mixed with the local language/script. This calls for script identification at the word and character levels. In their earliest work [52], they developed a method for locating the address block and extracting postal code from the address. In [53], [54], a two-stage neural network-based general classifier is used for the recognition of postal code digits written in Arabic or Bengali numerals. Since there exist shape similarities between some Arabic and Bengali numerals, the final assignment of script class is done in a second stage using majority voting. It was noted that the accuracy of the classifier was 98.42 percent in printed and about 89 percent in handwritten postcodes. Methods for wordwise script recognition in postal addresses using features like the water reservoir concept, headline ("shirorekha"), etc., in a tree classifier were proposed in [55]. Based on this, a two-stage MLP network was constructed in [56] that accomplishes wordwise script recognition in Indian postal addresses at more than 96 percent accuracy.

3.2 Appearance-Based Script Recognition

Script types generally differ from each other by the shape of individual characters and the way they are grouped into words, words into sentences, etc. This gives different scripts distinctively different visual appearances. Therefore, one natural way of identifying the script in which a document is written may be on the basis of its visual appearance as seen at a glance by a casual observer, without really analyzing the character patterns in the document. Accordingly, several features that describe the visual appearance of a script region have been proposed and used for script identification by many researchers, as described below.

3.2.1 Pagewise Script Identification Methods

One early attempt to characterize the script of a document without actually analyzing the structure of its constituent connected components was made by Wood et al. [57]. They proposed using vertical and horizontal projection profiles of document images for determining scripts in machine-generated documents. They argued that the projection profiles of document images are sufficient to characterize different scripts. For example, Roman script shows dominant peaks at the top and bottom of the horizontal projection profile, while Cyrillic script has a dominant midline and Arabic script has a strong baseline. On the other hand, the Korean characters usually have a peak on the left of the vertical projection profile. However, the authors did not suggest how these projection profiles can be analyzed automatically for script determination without any user intervention. Also, they did not present any recognition results to substantiate their argument.

Since visual appearance is often related to texture, a block of text corresponding to each script class forms a distinct texture pattern. Thus, the problem of script identification essentially boils down to a texture analysis problem and one may employ any available texture classification algorithm to perform the task. In accordance with this, Tan developed Gabor function-based texture analysis for machine-printed script identification that yielded an accuracy as high as 96.7 percent in discriminating printed Chinese, Latin, Greek, Russian, Persian, and Malayalam script documents [58]. In the first step of this method, a uniform text block on which texture analysis can be performed is produced from the input document image via the method given in [59]. Texture features are then extracted from the text block using a 16-channel Gabor filter with channels at a fixed radial frequency of 16 cycles/sec and at 16 equally spaced orientations. The average response of every channel provides a characteristic measure for the script that is robust to noise but rotation dependent. In order to achieve invariance to rotation, Fourier coefficients for this set of 16 channel outputs are calculated. During classification, a feature vector generated from the input text block is compared to the class-representative feature vectors using weighted (variance normalized) euclidean distance measure, as depicted in Fig. 15. A representative feature vector for a script class is obtained by computing the mean feature vector obtained from a large set of training documents written in that script.

One drawback with the above method is that the text blocks extracted from the input documents do not necessarily

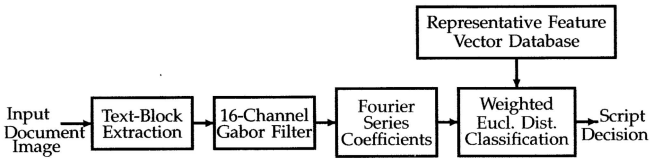


Fig. 15. Tan's script identification system using Gabor function-based rotation-invariant features.

have uniform character spacing. In view of this, Peake and Tan extended this work in [60], where they used some simple preprocessings to obtain uniform text blocks from the input-printed document. These include textline location, outsized textline removal, spacing normalization, and padding. Documents are also skew compensated so that it is not necessary to generate rotation-invariant features. For the purpose of feature extraction, gray-level co-occurrence matrices (GLCMs) and multichannel Gabor filter are used in independent experiments. GLCMs represent pairwise joint statistics of the pixels in an image and have long been used as a means for characterizing texture [61]. In Gabor filter-based feature extraction, a 16-channel filter with four frequencies at four orientations is used. These two approaches for texture feature extraction were applied to machine-printed documents written in seven different scripts (adding Korean to the six scripts used earlier in [58]). Script identification was then performed using KNN classification. It was seen that GLCM approach yields only 77.14 percent accuracy at best while Gabor filter approach yields accuracy rate as high as 95.71 percent.

One problem encountered in Gabor filter-related applications is the high computational cost due to the frequent image filtering. In order to reduce the cost of computation, script identification in machine-printed documents using steerable Gabor filters was proposed in [62]. The method offers twofold advantages. First, the steerability property of Gabor filter is exploited to reduce the high computational cost. Second, the Gabor filter bank is appropriately designed so that the extracted rotation-invariant features can discriminate scripts containing characters that are similar in shape and even share many characters. In this paper, a 98.5 percent recognition rate was achieved in discriminating the Chinese, Japanese, Korean, and Latin scripts, while the number of image filtering operations was significantly reduced by 40 percent.

Although the above Gabor function-based script recognition schemes have shown good performance, their application is limited to machine-printed documents only. Variations in writing style, character size, and interline and interword spacings make the recognition process difficult and unreliable when these techniques are applied directly on handwritten documents. Therefore, it is necessary to preprocess the document images prior to the application of the Gabor filter so as to compensate for the different variations present. This has been addressed in the texture-based script identification scheme proposed in [63]. In the preprocessing stage, the algorithm employs denoising, thinning, pruning, m-connectivity, and text size normalization in sequence. Texture features are then extracted using a multichannel Gabor filter. Finally, different scripts are classified using fuzzy classification. In this proposed system, an overall accuracy of 91.6 percent was achieved in classifying handwritten documents written in four different scripts, namely, Latin, Devnagari, Bengali, and Telugu.

Another visual attribute that has been used in many image processing applications is histogram statistics, which reflects spatial distribution of gray levels in an image. In a recent work [64], Cheng et al. proposed using normalized histogram statistics for the purpose of script identification in documents typeset in Latin, Chinese, Cyrillic, or Japanese. In this work, every line of text in an input document is divided into three zones—ascender zone between top line and x-line, x-zone between x-line and baseline, and descender zone between baseline and bottom line. Then, a horizontal projection is obtained for each textline that gives zonewise distribution of character pixels in a textline. It is observed that Latin and Cyrillic characters mainly distribute in the x-zone with two significant peaks located on the x-line and baseline. The baseline peak is higher than the x-line peak in Latin, while they are almost equal in Cyrillic. The Chinese characters, on the other hand, have relatively random distribution, without any peak in the profile. The Japanese characters also have the same random distribution, but the average height of the profile is significantly lower. Thus, it is possible to separate out every script from other scripts by analyzing the distribution of character pixels in different zones inside a document.

3.2.2 Script Identification at Paragraph and Text Block Level

The use of texture features in script identification was considered by Jain and Zhong for discriminating printed Chinese and English documents [65]. This paper in fact proposed a texture-based language-free page segmentation algorithm which automatically extracts text, halftone, and line-drawing regions from input gray-scale document images. An extension of this page segmentation procedure provides for further segmentation of the text regions into different script regions. First, a set of optimal texture discrimination masks is created through neural network training. Next, texture features are obtained by convolving the trained masks with the input image. These features are then used for classification.

The use of other texture features for script classification, other than GLCM and Gabor energy features, has been explored by Busch et al. [66]. The features that they used are wavelet energy features, wavelet log mean deviation features, wavelet co-occurrence signatures, wavelet log co-occurrence features, and wavelet scale co-occurrence signatures. They tested these features on a database containing eight different script types—Latin, Han, Japanese, Greek, Cyrillic, Hebrew, Devnagari, and Farsi. In their experiments, machine-printed document images of size 64×64 pixels were first binarized, skew corrected, and text block normalized, in line with the work done by Peake and Tan in [60]. In order to reduce the dimensionality of the feature vectors while improving classification accuracy, Fisher linear discriminant analysis technique is applied. Classification is performed using a Gaussian Mixture Model (GMM) classifier, which models each script class as a combination of Gaussian distributions. The GMM classifier

is trained using a version of the expectation maximization (EM) algorithm. In order to create a more stable and global script model, a maximum a posteriori (MAP) adaptation-based method was also proposed. It was seen that the wavelet log co-occurrence outperforms all other texture features for script classification (only 1 percent classification error) while GLCM features yielded the worst overall performance (9.1 percent classification error). This indicates that pixel relationships at small distances are insufficient to characterize the script of a document image appropriately.

However, a single model per script class is useful only when every script is written using only one font or using only visually similar fonts. On the contrary, there typically exist a large number of fonts, often of widely varying appearance, within a given script. Because of such variations, it is unlikely that a model trained on one set of fonts will correctly identify an image of a previously unseen font of the same script. For example, classification error increases from 1 and 9.1 percent in [66] to 15.9 and 13.2 percent in cases of wavelet log co-occurrence and GLCM features, respectively. In view of this, Busch proposed characterizing multiple fonts within a single script more adequately by using multiple models per script class [67]. This is done by partitioning each script class into 10 subclasses, each subclass corresponding to one font included within that script class. This is followed by linear discriminant analysis and classification using the modified MAP-GMM classifier as above. Such a classification system provides significant improvement when compared to the results obtained using a single model—classification error reduces to 2.1 and 12.5 percent for the above two cases, respectively.

Script identification in Indian printed documents using oriented local energy features was performed in [68]. Local energy is defined as the sum of squared responses of a pair of conjugate symmetric Gabor filters. In an earlier work, Chan and Coghill [69] derived a set of descriptors from oriented local energy and demonstrated their utility in script classification. In line with human perception, the features chosen are energy distribution, the ratio of energies for two nonadjacent channels, and the horizontal projection profile. The distribution of energy across differently oriented channels of a Gabor filter differs from one script to other. While this feature captures the global differences among scripts, a closer analysis of the energy distribution may be necessary to reveal finer differences between similar looking scripts. This is provided by the ratios between energies at the output of nonadjacent channel pairs. Finally, there are certain scripts which are distinguishable only by the stroke structures used in the upper part of the words. For example, Devnagari and Gurumukhi differ in the shape of the *matra* present above the headline ("shirorekha"). Horizontal projection is used to discover this information. One major advantage with these features is that it is not necessary to perform analysis at multiple frequencies but at only one optimal frequency. This helps in reducing the computational cost. Again, filter response can be enhanced by increasing filter bandwidth at this optimal frequency. Accordingly, the filters employed in [68] are log-Gabor filters designed for one empirically determined optimal

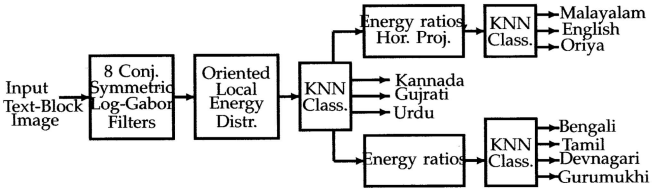


Fig. 16. Classification hierarchy in Joshi et al.'s script identification scheme.

frequency and at eight equispaced orientations. For an input text block of size 100×100 pixels, the aforementioned features are calculated and then classified into different script classes using a KNN classifier. The scheme was tested on 10 different scripts commonly used in India and an overall classification accuracy of 97.11 percent was achieved. The scripts used included Devnagari, Bengali, Tamil, Kannada, Malayalam, Gurumukhi, Oriya, Gujrati, Urdu, and Latin. Fig. 16 illustrates how these 10 different Indian scripts are classified using these features in two levels of hierarchy.

3.2.3 Script Identification at Word/Character Level

While all of the texture-based script identification methods described above work on a document page or a text block, script identification at the word level was successfully implemented in [70], [71], [72], [73], [74], [75], [76]. In the works by Ma et al. [70], [71], Gabor filter analysis is applied to each word in a bilingual document to extract features characterizing the script in which that particular word is written. Subsequently, a 2-class classifier system is used to discriminate the two different scripts contained in the input document. Different classifier architectures based on SVM, KNN, weighted euclidean distance, and GMM are considered. A classifier system consisting of a single classifier may consist of any of the above four architectures, while a multiple classifier system is built by combining two or more of them. In a multiple classifier system, the classification scores from each of the different component classifiers are combined using sum-rule to arrive at the final decision. In their papers, Ma et al. considered bilingual documents containing combinations of one Latin-based language (mainly English) and one non-Latin language (e.g., Arabic, Chinese, Hindi, or Korean). It was observed that while the performance for English-Hindi documents was quite good (97.51 percent recognition rate using KNN classifier), script identification in English-Arabic documents had the lowest performance (90.93 percent using SVM classifier). Moreover, it was established that multiple classifier system can consistently outperform the single classifier systems (98.08 and 92.66 percent in case of English-Hindi and English-Arabic documents, respectively, using a combination of KNN and SVM classifiers).

A visual-appearance-based approach has also been applied to identify and separate script words in Indian multiscript documents. In [72], [73], two different approaches to script identification at the word level in printed bilingual (Latin and Tamil) documents are presented. The first method structures words into three distinct spatial zones and utilizes the information about the spatial spread of the words in these zones. The second technique analyzes

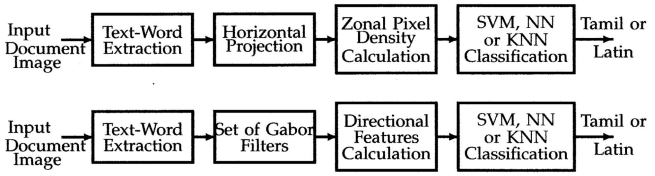


Fig. 17. Dhanya et al.'s two approaches to script identification in Tamil-English documents.

the directional energy distribution of words using Gabor filters with suitable frequencies and orientations. The algorithms are based on the observations as follows:

1. The spatial spread of Roman characters mostly covers the middle and upper zones; only a few lowercase characters spread to the lower zone.
2. The Roman alphabet contains more vertical and slanted strokes.
3. In Tamil, the characters mostly spread to the upper and lower zones.
4. There is a dominance of horizontal and vertical strokes in Tamil.
5. The aspect ratio of Tamil characters is generally more than that of the Roman characters.

These suggest that the features that may play a major role in discriminating Roman and Tamil script words are the spatial spread of the words and the direction of orientation of the structural elements of the characters in the words. The spatial feature is obtained by calculating zonal pixel concentration, while the directional features are available as responses of Gabor filters. The extracted features are classified using SVM, Nearest Neighbor, or KNN classifiers. A block schematic diagram of the system is presented in Fig. 17. It was observed that the directional features possess better discriminating capabilities than the spatial features, yielding as high as 96 percent accuracy in an SVM classifier. This may be attributed to the fact that Gabor filters can take into account the general nature of scripts better.

Dhanya and Ramkrishnan also attempted to recognize and separate out different script characters in printed Tamil-Roman documents using zonal occupancy information along with some structural features [74]. For this, they proposed a hierarchical scheme for extracting features from characters and classify them accordingly. Based on the zonal occupancy of characters, the scheme divides the combined alphabet set into four groups—characters that occupy all three zones (Group 1), characters that occupy the middle and lower zones (Group 2), characters that occupy the middle and upper zones (Group 3), and characters that occupy the middle zone only (Group 4). Groups 3 and 4 are further divided on the basis of presence or absence of loop structure in the character. This is followed by feature extraction, feature transformation, and finally, nearest neighbor classification. Features that may be extracted from a character are geometric moments, DCT coefficients, or DWT coefficients. Feature space transformation is required for dimension reduction while enhancing class discrimination. Three methods are proposed for the purpose—PCA, FLD, or maximization of divergence. The whole process is explained pictorially in Fig. 18. The proposed scheme

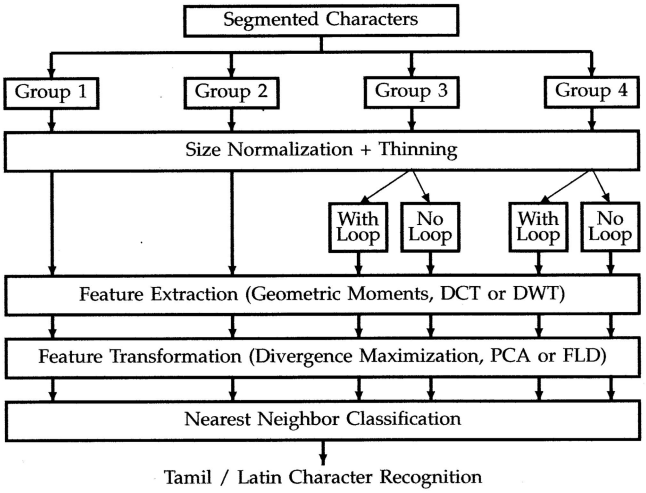


Fig. 18. Stages of character classification in a printed Tamil-Latin document.

yielded recognition accuracies of 94 percent and above when tested on 20 document samples, each containing a minimum of 300 characters.

In [75], a Gabor function-based multichannel directional filtering approach is used for both text area separation and script identification at the word level. It may be assumed that the text regions in a document are predominantly high-frequency regions. Hence, a filter-bank approach may be useful in discriminating text regions from nontext regions. The script classification system using a Gabor filter with four radial frequencies and four orientations showed a high degree of classification accuracy (minimum 96.02 percent and maximum 99.56 percent) when applied to bilingual documents containing Hindi, Tamil, or Oriya along with English words. In an extended version of this work [76], the method was applied to documents containing three scripts and five scripts. In this filter bank approach to script recognition, the Gabor filter bank uses three different radial frequencies and six different angles of orientations. For decision making, two different classifiers are considered—linear discriminant classifier and the commonly used nearest neighbor classifier. It was observed in several experiments that both of the classifiers perform well with Gabor feature vectors, although in some cases, the nearest neighbor classifier performs marginally better—the average accuracy obtained in case of triscript documents was 98.4 and 98.7 percent with linear discriminant and nearest neighbor classifiers, respectively. The highest recognition accuracy obtained was 99.7 percent using the nearest neighbor classifier in a biclass problem, while the lowest attained recognition rate was 97.3 percent.

3.3 Comparative Analysis

Table 1 summarizes some of the benchmark work in script recognition. Various script features used by different researchers are listed in this table. However, the results they reported, although quite encouraging on most occasions, were obtained using only a selected number of script classes in their experiments. This leaves a question that how these script features will perform when applied to scripts other than those considered in their works. Therefore, it is important to investigate the discriminative power of each script identification feature proposed in the literature before

TABLE 1
Script Recognition Methods

Researchers	Method		Script types classified	Scope of application		Best recog. reported
	Features	Classifier				
Structure-based script recognition methods						
Spitz [15]	Upward concavity distribution	Var. comparison	Latin, Han	Printed	Page-wise	100%
	Optical density	LDA + Eucl. Dist.	Chinese, Japanese, Korean			
Lam, Ding, Suen [18]	Hor. proj., height distribution	Stat. classifier	Latin, Oriental scripts	Printed	Page-wise	95%
	Circles, ellipses, ver. strokes	Freq. of occurr.	Chinese, Japanese, Korean			
Hochberg et al [19]	Textual symbols	Hamming Dist. classifier	Arabic, Armenian, Devnag., Chinese, Cyrillic, Burmese, Ethiopic, Japanese, Hebrew, Greek, Korean, Latin, Thai	Printed	Page-wise	96%
Hochberg et al [34]	Textual symbols	Hamming Dist. classifier		Printed	Word-wise	NA*
Hochberg et al [20]	Hor./ver. centroids, sphericity, aspect ratio, white holes	FLD	Arabic, Chinese, Cyrillic, Devnagari, Japanese, Latin	Hand-written	Page-wise	88%
Pal et al [29]	Headline, strokes, ver. runs, lowermost pt., water resv.	Freq. of occurr.	Devnag., Bengali, Chinese, Arabic, Latin	Printed	Line-wise	97.33%
Elgammal et al [31]	Hor. proj. peak, moments, run-length distribution	Feedforward NN	Arabic, Latin	Printed	Line-wise	96.8%
Moalla et al [41]	Arabic character segments	Template match	Arabic, Latin	Printed	Word-wise	100%
Jawahar et al [45]	Headline, context info.	PCA + SVM	Devnagari, Telugu	Printed	Word-wise	92.3% to 99.86%
Chanda et al [47]	Headline, ver. strokes, tick left/right profiles, water resv., deviation, loop, left incline.	Freq. of occurr.	Devnagari, Bengali, Latin, Malayalam, Gujrati, Telugu	Printed	Word-wise	97.92%
Visual appearance-based script recognition methods						
Wood et al [57]	Horizontal / vertical proj.	—	Arabic, Cyrillic, Korean, Latin	Printed	Page-wise	NA*
Jain et al [65]	Texture feature using discriminating masks	MLP	Latin, Chinese	Printed	Para-wise	NA*
Tan [58]	Gabor filter-based texture feature	Weighted Euclid. Dist.	Chinese, Greek, Malayalam, Latin, Russian, Persian	Printed	Page-wise	96.7%
Peake et al [60]	GLCM features	KNN Classifier	Chinese, Greek, Malayalam, Latin, Russ., Persian, Korean	Printed	Page-wise	77.14%
	Gabor filter					95.71%
Singhal et al [63]	Gabor filter-based texture feature	Fuzzy Classifier	Devnagari, Bengali, Telugu, Latin	Hand-written	Page-wise	91.6%
Busch et al [66]	GLCM features	LDA + GMM	Latin, Chinese, Japanese, Cyrillic, Greek, Devnagari, Hebrew, Persian	Printed	Para-wise	90.9%
	Gabor energy					95.1%
	Wavelet energy					95.4%
	Wavelet Log Mean Dev.					94.8%
	Wavelet Co-occurrence					98%
	Wavelet Log Co-occurrence					99%
	Wavelet Scale Co-occurrence					96.8%
Joshi et al [68]	Gabor Energy distribution, horizontal projection profile, energy ratios	KNN Classifier	Devnag., Latin, Gurumukhi, Kannada, Malayalam, Urdu, Tamil, Gujrati, Oriya, Beng.	Printed	Para-wise	97.11%
Ma et al [70], [71]	Gabor filter-based texture feature	KNN + SVM Multi-classifier	Latin, Devnagari	Printed	Word-wise	98.08%
			Latin, Arabic			92.66%
Dhanya et al [73]	Gabor filter-based directional feature	SVM	Tamil, Latin	Printed	Word-wise	96%

*NA: Not available – recognition result not given in terms of numeric value.

one may use it for the purpose. In view of this, a comparative analysis between different methods and script features is desirable.

One important structural feature for script recognition used by Spitz and some others is the character optical density. This is the measure of character pixels inside a character bounding box, which is distinctly very high in scripts using complex ideographic characters. Structurally simple Arabic characters, on the other hand, are low in density. All other scripts across Europe and Asia show more or less the same medium character density. Therefore, while this feature may be good in separating out Han, on one hand, and Arabic, on the other, it does not help much in bringing out the difference between moderately complex scripts like Latin, Cyrillic, Brahmic scripts, etc. The second discriminating feature that Spitz used is the location of upward concavities in characters. An upward concavity is formed when a run of character pixels spans the gap between two white runs just above it. As a result, upward concavities in a character are observed at points where two or more character strokes join. Accordingly, ideograms composed of multiple strokes show many more upward concavities per character compared to that in other scripts. As observed by Spitz [77], there are usually at most two or three upward concavities in a single Latin character while Han characters have many more upward concavities per character that are evenly distributed along the vertical axis. However, we observe that most other scripts also show two or three upward concavities, the same as in the Latin script. So, upward concavity is good for separating Han from

others but not good for discrimination among non-Han scripts, except perhaps for Cyrillic, which contains a few more upward concavities compared to other non-Han scripts. Another problem with these two features is that they highly depend on document quality. Broken character segments may result in detection of false upward concavity, while noise contributes to optical density measure. Non-Han documents tend to be misclassified as Han-based Oriental ones if the document quality is poor, because many characters are either broken or noisy. In order to cope with such situations, features like character height distribution, character bounding box profiles, horizontal projections, and several other statistical features were proposed in [16], [17], [18]. These features do not depend on the document quality and resolution but on the overall size of the connected components. However, these features are not invariant to character size and font and offer high performance only in separating distinctly different Oriental scripts from other non-Han scripts.

Several different structural features, like character geometry, occurrence of certain stroke structures and structural primitives, stroke orientations, measure of cavity regions, side profiles, etc., that directly relate to the character shape have also been used for script characterization. However, while some features show marked difference between two scripts, measures of other features may be the same between that script pair. For example, while Devnagari and Gujrati can be easily identified using "shirorekha" and water reservoir-based features, character aspect ratio and character moments do not show much difference. This is because many Gujrati letters are exactly same as their Devnagari counterpart with the headline ("shirorekha") removed. Again, there are features that are optimal in one script pair but not in another pair. For example, the presence of "shirorekha" may be a good feature for discriminating Latin and Devnagari, but not at all useful in separating Devnagari and Bengali. Therefore, in order to separate out a script from all other scripts, one may need to check a large pool of structural features before any decision can be taken. This may result in the curse of dimensionality. So, a better option may be to do the classification using different sets of features at different levels of hierarchy, as proposed in some of the works above. Another option is to learn the script characteristics in a neural network, as in [25], without bothering about the features to be used for classification. However, a larger network with a greater number of hidden units may be necessary for reliable recognition as more and more script classes are included.

Compared to the above, Hochberg et al.'s method is more versatile. The method is based on discovering frequent characters/symbols in every script class and storing them in the system database for matching during classification. Therefore, in principle, the method can identify any number of scripts of varied nature and font as long as they are included in the training set. It is possible to apply the method in a common framework to scripts containing discrete and connected characters, alphabetic and nonalphabetic scripts, and so on, as demonstrated in [19], [34]. However, it is not difficult to realize that the classification error due to ambiguity will increase if the system includes script classes that use similar looking characters or even share many common characters. Therefore, Hochberg's method may not be suitable in a multiscript country like India, where most scripts have the same line of origin. Nevertheless, it offers invariance to font size and computational simplicity. This is because textual symbols are size-normalized and the algorithm uses simple binary shape matching without any feature value calculation.

Another important feature proposed by Wood et al. and used by many researchers is the horizontal projection. This gives a measure of the spatial spread of the characters in a script that provides an important clue to script identification. Some scripts can be identified by detecting the peaks in the projection profile, e.g., Arabic scripts having a strong baseline show peak at the bottom of the profile while Brahmic scripts with "shirorekha" show peak at the top, and so on. However, this feature also is not good for separating scripts of similar nature and structure. For example, Devnagari, Bengali, and Gurumukhi will show the same peak in the profile due to "shirorekha"; Arabic, Urdu, and Farsi have the same lower peak. Hence, this feature has not been used alone but mostly in combination with other structural features.

A better approach to script identification is via texture feature extraction using multichannel Gabor filter that provides a model for human vision system. This means that Gabor filter offers a powerful tool to extract out visual attributes from a document. This has motivated many researchers to employ Gabor filter for script determination. Since texture feature gives the general appearance of a script, it can be derived from any script class of any nature. Accordingly, this feature may be considered a universal one. The discriminating power of a multichannel Gabor filter can be varied by having more channels with different radial frequencies and closely spaced orientation angles. Thus, this system is flexible compared to all other methods and can be effectively used in discriminating scripts that are quite close in appearance. The main criticism with this approach is that it cannot be applied with confidence to small text regions as in wordwise script recognition. Also, Gabor filters are not capable of handling variations in script size and font, interline spacings, etc.

Table 1 also lists recognition rates, as reported in the literature. Since the experiments were conducted independently using different data sets, however, they do not reflect the comparative performance of these methods. To have a proper measure of their relative script separation power, these methods need to be applied on a common data set. Script recognition performance of some of the above-mentioned features, when applied to a common data set, is given in Table 2. The data set contains printed documents typeset in 10 different scripts, including six scripts used in India. In the absence of any standard database, we created our own database by collecting document samples from books and magazines. Some documents were also available from the World Wide Web, which we printed using a laser printer. All of the documents were scanned in black-and-white mode at 300 dpi and then rescaled to have a standard textline height in all documents while maintaining the

TABLE 2
Script Recognition Results (in Percentage)

Script Features Used	Latin	Cyrillic	Arabic	Urdu	Chinese	Korean	Devnagari	Bengali	Gujrati	Tamil
Optical Density [13]	75.4	84.6	89.1	87.2	96.3	93.7	76.2	73.4	74.0	83.8
Textual Symbol [19]	97.2	92.3	93.7	90.1	97.2	94.3	95.3	97.8	87.1	98.9
Hor. Projection Profile [23]	89.7	91.2	94.3	92.9	87.5	90.2	92.1	90.0	94.6	76.8
Gabor Coefficients [60]	95.2	92.7	97.2	94.3	93.3	89.9	95.8	91.3	87.8	96.2

character aspect ratio. Script recognition was performed at the text block level. Homogeneous text blocks of size 256×256 pixels were extracted from document pages in such a way that page margins and nontextual parts were excluded. A total of 120 text blocks were generated per script, each block containing 10 to 12 textlines. The print quality of the documents, and hence, the quality of the document images was reasonably good containing very little noise.

We observe that the optical density feature is capable of identifying Chinese and Korean and also Arabic and Urdu to some extent. For other script classes, the recognition rate was well below the acceptable level. This is because the optical density feature is not good enough to discriminate among scripts of similar complexity. The same argument holds for other script features. The Gabor filter method shows relatively better discriminating power in comparison. We noticed that the classification error was mainly due to the misclassification between script pairs like Arabic and Urdu, Chinese and Korean, Devnagari and Bengali, and Devnagari and Gujrati. These pairs of script classes have characters of the same nature and complexity, and even share some common characters. This leads to ambiguity, and hence, the classification error. So, on the whole, we may say that every proposed script identification method and script feature works well only when applied within a small set of script classes. Classification accuracy falls significantly when more scripts of similar nature and origin are included.

As observed in Table 1, almost all works on script recognition are targeted toward machine-printed documents. They have not been tested for script recognition in handwritten documents. In view of the large amount of handwritten documents that need to be processed electronically nowadays, script identification in handwritten documents turns out to be an important research issue. Unfortunately, the script features proposed for printed documents may not be always effective in case of handwritten documents. Variations in writing style, character size, and interline and interword spacings make the recognition process difficult and unreliable when these techniques are applied to the handwritten documents. Variation in writing across a document can be taken care of by using certain statistical features, as proposed in [20]. Textual symbol-based method can also be used but with certain modifications—some shape descriptor features can be derived from the text symbols and the prototypes can be generated through clustering. We demonstrated this approach in an earlier paper [35]. Also, a script class may be represented by multiple models to account for variation in writing from one person to another.

Based on our discussion above, we see that script features are extracted either from a list of connected components like textline, word, and character in a document or from a patch of text that may be a complete paragraph, a text block cropped from the input document, or even the whole document page. Script identification methods that use segmentwise analysis of character structure may hence be regarded as local approach. On the other hand, visual appearance-based methods that are designed to identify script by analyzing the overall look of a text block may be regarded as a global approach.

As discussed before, many different structural features and methods for script characterization have been proposed over the years. In each of these methods, the features were chosen keeping in view only those script types that were considered therein. Therefore, while these features have been proven to be efficient for script identification within a given set of scripts, they may not be good in separating a wider variety of script classes. Again, structural features cannot effectively discriminate between scripts having similar character shapes, which otherwise may be distinguished by their visual appearances. Another disadvantage with structure-based methods is that they require complex preprocessing involving connected component extraction. Also, extraction of structural features is highly susceptible to noise and poor-quality document images. The presence of noise or significant image degradation adversely affects the location and segmentation of these features, making them difficult or sometimes impossible to extract.

In short, the choice of features in local approach to script classification depends on the script classes to be identified. Further, the success of classification in this approach depends on the performance of the preprocessing stage, which includes denoising and extraction of connected components. Ironically, document segmentation and extraction of connected components sometimes require the script type to be known a priori. For example, an algorithm that is good for segmenting ideograms in Han may not be equally effective in segmenting alphabetic characters in the Latin script. This presents a paradox in that, for determining the script type, it is necessary to know the script type beforehand. In contrast, text block extraction in visual appearance-based global approaches is simpler and can be employed irrespective of the document's script. Since here it is not necessary to extract individual script components, such methods are better suited to degraded and noisy documents. Also, global features are more general in nature and can be applied to a broader range of script classes. They have practical importance in script-based retrieval systems because they are relatively fast and reduce the cost of document handling. Thus, visual appearance-based methods prove to be better than structure-based script identification methods in many ways, as listed in

TABLE 3
Local versus Global Approaches for Script Identification

	Local approaches	Global approaches
Preprocess.	Line, word, character segmentation	Text-block extraction
	Complex and script dependent	Simple and script independent
Scope of application	Page-wise, Para-wise, Line-wise, Word-wise	Page-wise, Para-wise
	Limited script types	Wider variety of scripts
Robustness	Sensitive to noise	Less prone to noise
	Moderately robust to skew, font size / type	Moderately robust to skew, font size and type

Table 3. However, local approach is useful in applications involving textlinewise, wordwise, and even characterwise script identification, which otherwise are generally not possible through global approach. Since local methods extract features from elemental structures present in a document, in principle they can be applied at all levels within the document. Nonetheless, some structure-based methods demand a minimum size of the text to arrive at some conclusive decision. For example, Spitz's two-stage script classifier [15] requires at least two lines of text in the first level of classification and at least six lines in the second stage. Likewise, at least 50 textual symbols need to be verified for acceptable classification in [19]. The same applies to methods in which the script class decision is based on statistics taken across the input document. We also note that methods developed for pagewise script identification can also be used for script recognition in a paragraph or a text block as long as the document size is big enough to provide necessary information.

4 Online Script Recognition

The script identification techniques described earlier are for offline script recognition and are, in general, not applicable to online data. With the advancement of pen computing technology in the last few decades, many online document analysis systems have been developed in which it is necessary to interpret the written text as it is input by analyzing the spatial and temporal nature of the movement of the pen. Therefore, as in the case of OCR systems for offline data, an online character recognizer in a multiscript environment must be preceded by an online script recognition system. Unfortunately, in comparison to offline script recognition, not much effort has been dedicated toward the development of online script recognition techniques. As of today, only a few methods are available for online script recognition, as described below.

One of the earliest works on online script recognition was reported in [78] by Lee et al. Later, they extended their work in [79]. Their method is based on the construction of a unified recognizer for the entire set of characters incorporated from more than one script, and an approach using HMM network is proposed for recognizing sequences of words in multiple languages. Viewing handwritten script as an alternating sequence of words and interword ligatures, a hierarchical HMM is constructed by interconnecting HMMs for ligatures and words in multiple languages. These component HMMs are, in turn, modeled by a network of interconnected character and ligature models. Thus, basic characters of a language, language network, and intermixed use of language are modeled with hierarchical relations. Given such a construction, recognition corresponds to finding the optimal path in the network using the Viterbi algorithm. This approach can be used for recognizing freely handwritten text in more than one language and can be applied to any combination of phonetic writing systems. Results of word recognition tests showed that Hangul words can be recognized with about 92 percent accuracy while English words can be recognized correctly only 84 percent of the time. It was also observed that by combining multiple languages, recognition accuracy drops negligibly but speed is slowed substantially. Therefore, a more powerful search method and machine are needed to use this technique in practice.

The basic principle behind online character recognition is to capture the temporal sequence of strokes. A stroke is defined as the locus of tip of the pen from pen-down to the next pen-up position. For script recognition, therefore, it may be useful to check the writing style associated with each script class. For example, Arabic and Hebrew scripts are written from right to left, Devnagari script is characterized by the presence of "shirorekha," a Han character is composed of several short strokes, and so on. An online system can capture such information and be used for script identification. In [80], Namboodiri and Jain proposed nine measures that may be used to quantify the characteristic writing style of every script. They are:

1. horizontal interstroke direction defining the direction of writing within a textline,
2. average stroke length,
3. "shirorekha" strength,
4. "shirorekha" confidence,
5. stroke density,
6. aspect ratio,
7. reverse direction defined as the distance by which the pen moves in the direction opposite to the normal writing direction,
8. average horizontal stroke direction, and
9. average vertical stroke direction.

Their proposed classification system, based on the above spatial and temporal features of the strokes, attained classification accuracies in between 86.5 and 95 percent in different experimental tests. Later, they added two more features in [81], viz., vertical interstroke direction and variance of stroke length, and achieved around 0.6 percent improvement in the classification accuracy.

A unified syntactic approach to online script recognition was presented in [82] and was applied for classifying Latin, Devnagari, and Kanji scripts by analyzing their characteristic properties that include the fuzzy linguistic descriptors to describe the character features. The fuzzy pattern description language Fuzzy Online Handwriting Description Language (FOHDEL) is used to store fuzzy feature values for every character of a script class in the form of fuzzy rules. For example, the character "b" in the Roman alphabet may be described as consisting of two fuzzy linguistic terms—*very straight vertical line at the beginning*

followed by *an almost circular curve at the end*. These fuzzy rules aid in decision making during classification.

5 Script Recognition in Video Text

Script identification is not only important for document analysis but also for text recognition in images and videos. Text recognition in images and videos is important in the context of image/video indexing and retrieval. The process includes several preprocessing steps like text detection, text localization, text segmentation, and binarization before an OCR algorithm may be applied. As with documents in an multiscript environment, image/video text recognition in an international environment also requires script identification in order to apply suitable algorithm for text extraction and recognition. In view of this, an approach for discriminating between Latin and Han script was developed in [83]. The proposed approach proceeds as follows: First, the text present in an image or video frame is localized and size normalized. Then, a set of low-level features is extracted from the edges detected inside the text region. This includes mean and standard deviation of edge pixels, edge pixel density, energy of edge pixels, horizontal projection, and Cartesian moments of the edge pixels. Finally, based on the extracted features, the decision about the type of the script is made using a KNN classifier. Experimental results have demonstrated the efficiency of the proposed method by identifying Latin and Han scripts accurately at the rate of 85.5 and 89 percent, respectively.

6 Issues in Multiscript OCR System Evaluation

In connection with research in script recognition, it is useful and important to develop benchmarks and methodologies that may be employed to evaluate the performance of multiscript OCR systems. Some aspects of this problem have been reported in [84], and are discussed below.

The OCR evaluation approaches are broadly classified into two categories: black box evaluation and white box evaluation. In black box evaluation, only the input and output are visible to the evaluator. In a white box evaluation procedure, outputs of different modules comprising the system may be accessed and the total system is evaluated stage by stage. Nevertheless, the primary issues related to both types of evaluation are recognition accuracy and processing speed. The parameters that can be varied for the purpose of evaluation are content, font size and style, print and paper quality, scanning resolution, and the amount of noise and degradation in the document images.

Needless to say, the overall performance of a multiscript OCR greatly depends on the performance of the script recognition algorithm used in the system. As with any OCR system, the efficiency of a script recognizer is mainly assessed on the basis of accuracy and speed. Another important performance criterion is the minimum size of the document necessary for the script recognizer to perform reliably. This is to measure how the recognizer performs with varying document size.

In a multiscript system, another issue of consideration is the writing system adopted by a script, script complexity, and the size of the character set. Since some scripts are simple in nature and some are quite complex, a relative comparison of performance across scripts is a difficult task. For example, Latin is generally simpler in structure and is based on an alphabetic system. A script identifier that is good in recognizing Latin scripts may not be so in the case of complex nonalphabetic scripts like Arabic, Han, and Devnagari. Therefore, in order to evaluate various systems, a standard set of data should be used so that the evaluation is unbiased. However, it is generally difficult to find document data sets in different languages/scripts that are similar in content and layout. To address this problem, Kanungo et al. introduced the *Bible* as a data set for evaluating multilingual and multiscript OCR performance [85]. Bible translations are closely parallel in structure, relevant with respect to modern day language, widely available, and inexpensive. These make the Bible attractive for controlling document content while varying language and script. The document layout can also be controlled by using synthetically generated page image data. Other holy books, whose translation has similar properties, like the *Quran* and the *Bhagavad Gita*, have also been suggested by some researchers.

One major concern with most of the reported works in script recognition is the lack of any comparative analysis of the results. Experimental results given for every proposed method have not been compared with other benchmark works in the field. Moreover, the data sets used in experiments are all different. This is mainly due to the lack of availability of a standard database for script recognition research. Consequently, it is hard to assess the results reported in the literature. Hence, a standard evaluation testbed containing documents written in only one script type as well as multiscript documents with a mix of different scripts within a document is necessary. One important consideration in selecting the data set for a script class is that it should reflect the global probability of occurrence of the characters in texts written in that particular script. Another problem of concern is for languages that constantly undergo spelling modifications and graphemic changes over the years. As a result, if an old document is chosen as the corpus, then it may not be suitable for evaluating a modern OCR system. On the other hand, a database of modern documents may not be useful if the goal of the OCR is to process historic documents. This suggests that the data set should include all different forms of the same language that evolved with time, with full coverage of the script alphabet of different languages, and it should be large enough to reflect the statistical occurrence probability of the characters.

7 Conclusion

This paper presents a comprehensive survey on the developments in script recognition technology, which is an important issue in OCR research in our multilingual multiscript world. Researchers have attempted to characterize different scripts either by extracting their structural features or by deriving some visual attributes. Accordingly, many different script features have been proposed over the years for script identification at different levels within a document—pagewise, paragraphwise, textlinewise, wordwise, and even characterwise. Textlinewise and wordwise script

identifications are particularly important for use in a multi-script document. However, compared to the large arsenal of literature available in the field of document analysis and optical character recognition, the volume of work on script identification is relatively thin. The main reason is that most research in the area of OCR has been directed at solving issues within the scope of the country where the research is conducted. Since most countries in the world use only one language/script, OCR research in these countries need not bother determining the script in which a document is written. For instance, the US postal department spent a lot in developing system for automatic reading of postal addresses, but under the assumption that all letters originating or arriving in US will carry addresses written in English only. Script recognition is important only in an international environment or in a country that uses more than one script.

Nonetheless, with recent economic globalization and increased business transactions across the globe, there had been increased awareness of automatic script recognition among the OCR community. That is why the majority of the reported works are dated only during the last decade. However, it is noted that most of these script recognition methods have been tested on machine-printed documents only, and their performance on handwritten documents is not known. In view of this, it will be not wrong to say that script recognition in handwritten documents is still in its early stage of research. Since the present thrust in OCR research is in handwritten document analysis, parallel research on script identification in handwritten documents is in demand. Also, not many of these script recognition techniques have addressed font variation within a script class. Hence, we can conclude that script recognition technology still has a way to go, especially for handwritten document analysis. Therefore, there is an urgent need to work on script recognition of handwritten documents and in developing font-independent script recognizers.

As is evident from our analysis, development in script recognition technology lacks a generalized approach to the problem that can handle all different types of scripts under a common framework. While a particular script feature proves to be efficient within a set of scripts, it may not be useful in other scripts. To some extent, texture features can be used universally but cannot be applied reliably at word and character levels within a document.

Finally, we need to create a standard data set for research in this field. This is necessary to evaluate different script recognition methodologies under the same conditions. The creation of standard data resources will undoubtedly provide a much needed resource to researchers working in this field.

REFERENCES

[1] C.Y. Suen, M. Berthod, and S. Mori, "Automatic Recognition of Handprinted Characters—The State of the Art," *Proc. IEEE,* vol. 68, no. 4, pp. 469-487, Apr. 1980.

[2] J. Mantas, "An Overview of Character Recognition Methodologies," *Pattern Recognition,* vol. 19, no. 6, pp. 425-430, 1986.

[3] V.K. Govindan and A.P. Shivaprasad, "Character Recognition—A Review," *Pattern Recognition,* vol. 23, no. 7, pp. 671-683, 1990.

[4] S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proc. IEEE,* vol. 80, no. 7, pp. 1029-1058, July 1992.

[5] H. Bunke and P.S.P. Wang, *Handbook of Character Recognition and Document Image Analysis.* World Scientific Publishing, 1997.

[6] N. Nagy, "Twenty Years of Document Image Analysis in PAMI," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 38-62, Jan. 2000.

[7] U. Pal, "Automatic Script Identification: A Survey," *J. Vivek,* vol. 16, no. 3, pp. 26-35, 2006.

[8] U. Pal and B.B. Chaudhuri, "Indian Script Character Recognition: A Survey," *Pattern Recognition,* vol. 37, no. 9, pp. 1887-1899, Sept. 2004.

[9] L. Peng, C. Liu, X. Ding, and H. Wang, "Multilingual Document Recognition Research and Its Application in China," *Proc. Int'l Conf. Document Image Analysis for Libraries,* pp. 126-132, Apr. 2006.

[10] A. Nakanishi, *Writing Systems of the World: Alphabets, Syllabaries, Pictograms.* Charles E. Tuttle Co., 1980.

[11] F. Coulmas, *The Blackwell Encyclopedia of Writing Systems.* Blackwell Publishers, 1996.

[12] C. Ronse and P.A. Devijver, *Connected Components in Binary Images: The Detection Problem.* John Wiley & Sons, 1984.

[13] A.L. Spitz, "Multilingual Document Recognition," *Proc. Int'l Conf. Electronic Publishing, Document Manipulation, and Typography,* pp. 193-206, Sept. 1990.

[14] A.L. Spitz and M. Ozaki, "Palace: A Multilingual Document Recognition System," *Proc. IAPR Workshop Document Analysis Systems,* pp. 16-37, Oct. 1994.

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

[16] D.-S. Lee, C.R. Nohl, and H.S. Baird, "Language Identification in Complex, Unoriented, and Degraded Document Images," *Proc. IAPR Workshop Document Analysis Systems,* pp. 76-98, Oct. 1996.

[17] B. Waked, S. Bergler, C.Y. Suen, and S. Khoury, "Skew Detection, Page Segmentation and Script Classification of Printed Document Images," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* vol. 5, pp. 4470-4475, Oct. 1998.

[18] L. Lam, J. Ding, and C.Y. Suen, "Differentiating between Oriental and European Scripts by Statistical Features," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 12, no. 1, pp. 63-79, Feb. 1998.

[19] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic Script Identification from Document Images Using Cluster-Based Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 176-181, Feb. 1997.

[20] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly, "Script and Language Identification for Handwritten Document Images," *Int'l J. Document Analysis and Recognition,* vol. 2, nos. 2/3, pp. 45-52, Dec. 1999.

[21] Y. Tho and Y.Y. Tang, "Discrimination of Oriental and Euramerican Scripts Using Fractal Feature," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 1115-1119, Sept. 2001.

[22] B.V. Dhandra, P. Nagabhushan, M. Hangarge, R. Hegadi, and V.S. Malemath, "Script Identification Based on Morphological Reconstruction in Document Images," *Proc. IEEE Int'l Conf. Pattern Recognition,* vol. 2, pp. 950-953, Aug. 2006.

[23] S. Chaudhury and R. Sheth, "Trainable Script Identification Strategies for Indian Languages," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 657-660, Sept. 1999.

[24] S.B. Patil and N.V. Subbareddy, "Neural Network Based System for Script Identification in Indian Documents," *Sadhana,* vol. 27, no. 1, pp. 83-97, Feb. 2002.

[25] Z. Chi, Q. Wang, and W.-C. Siu, "Hierarchical Content Classification and Script Determination for Automatic Document Image Processing," *Pattern Recognition,* vol. 36, no. 11, pp. 2483-2500, Nov. 2003.

[26] S. Kanoun, A. Ennaji, Y. Lecourtier, and A.M. Alimi, "Script and Nature Differentiation for Arabic and Latin Text Images," *Proc. Int'l Workshop Frontiers in Handwriting Recognition,* pp. 309-313, Aug. 2002.

[27] L. Zhou, Y. Lu, and C.L. Tan, "Bangla/English Script Identification Based on Analysis of Connected Component Profiles," *Proc. Int'l Workshop Document Analysis Systems,* pp. 243-254, Feb. 2006.

[28] U. Pal and B.B. Chaudhuri, "Script Line Separation from Indian Multi-Script Documents," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 406-409, Sept. 1999.

[29] U. Pal and B.B. Chaudhuri, "Identification of Different Script Lines from Multi-Script Documents," *Image and Vision Computing,* vol. 20, nos. 13/14, pp. 945-954, Dec. 2002.

[30] U. Pal, S. Sinha, and B.B. Chaudhuri, "Multi-Script Line Identification from Indian Documents," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 880-884, Aug. 2003.

[31] A. Elgammal and M.A. Ismail, "Techniques for Language Identification for Hybrid Arabic-English Document Images," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 1100-1104, Sept. 2001.

[32] C.S. Cumbee, *Method of Identifying Script of Line of Text,* US Patent 7020338, Mar. 2006.

[33] S.-W. Lee and J.-S. Kim, "Multi-Lingual, Multi-Font, Multi-Size Large-Set Character Recognition Using Self-Organizing Neural Network," *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 1, pp. 28-33, Aug. 1995.

[34] J. Hochberg, M. Cannon, P. Kelly, and J. White, "Page Segmentation Using Script Identification Vectors: A First Look," *Proc. Symp. Document Image Understanding Technology,* pp. 258-264, Apr./May 1997.

[35] D. Ghosh and A.P. Shivaprasad, "Handwritten Script Identification Using Possibilistic Approach for Cluster Analysis," *J. Indian Inst. of Science,* vol. 80, pp. 215-224, May/June 2000.

[36] V. Ablavsky and M.R. Stevens, "Automatic Feature Selection with Applications to Script Identification of Degraded Documents," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 750-754, Aug. 2003.

[37] R. Krishnapuram and J.M. Keller, "A Possihilistic Approach to Clustering," *IEEE Trans. Fuzzy Systems,* vol. 1, no. 2, pp. 98-110, May 1993.

[38] D. Ghosh and A.P. Shivaprasad, "An Analytic Approach for Generation of Artificial Handprinted Character Database from Given Generative Models," *Pattern Recognition,* vol. 32, no. 6, pp. 907-920, June 1999.

[39] D.W. Muir and T. Thomas, *Automatic Language Identification by Stroke Geometry Analysis,* US Patent 6064767, May 2000.

[40] Y.-H. Liu, C.-C. Lin, and F. Chang, "Language Identification of Character Images Using Machine Learning Techniques," *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 2, pp. 630-634, Aug./Sept. 2005.

[41] I. Moalla, A. Elbaati, A.M. Alimi, and A. Benhamadou, "Extraction of Arabic Text from Multilingual Documents," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* http://ieeexplore.ieee.org/iel5/8325/26298/01173266.pdf?arnumber=1173266, Oct. 2002.

[42] I. Moalla, A.M. Alimi, and A. Benhamadou, "Extraction of Arabic Words from Multilingual Documents," *Proc. Conf. Artificial Intelligence and Soft Computing,* http://www.actapress.com/PDFViewer.aspx?paperId=18567, Sept. 2004.

[43] C.L. Tan, P.Y. Leong, and S. He, "Language Identification in Multi-Lingual Documents," *Proc. Int'l Symp. Intelligent Multimedia and Distance Education,* pp. 59-64, Aug. 1999.

[44] S. Lu, C.L. Tan, and W. Huang, "Language Identification in Degraded and Distorted Document Images," *Proc. Int'l Workshop Document Analysis Systems,* pp. 232-242, Feb. 2006.

[45] C.V. Jawahar, M.N.S.S.K. Pavan Kumar, and S.S. Ravi Kiran, "A Bilingual OCR for Hindi-Telugu Documents and Its Applications," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 408-412, Aug. 2003.

[46] S. Sinha, U. Pal, and B.B. Chaudhuri, "Word-Wise Script Identification from Indian Documents," *Proc. IAPR Int'l Workshop Document Analysis Systems,* pp. 310-321, Sept. 2004.

[47] S. Chanda, S. Sinha, and U. Pal, "Word-Wise English Devnagari and Oriya Script Identification," *Speech and Language Systems for Human Communication,* R.M.K. Sinha and V.N. Shukla, eds., pp. 244-248, Tata McGraw-Hill, 2004.

[48] S. Chanda and U. Pal, "English, Devnagari and Urdu Text Identification," *Proc. Int'l Conf. Cognition and Recognition,* pp. 538-545, Dec. 2005.

[49] S. Chanda, R.K. Roy, and U. Pal, "English and Tamil Text Identification," *Proc. Nat'l Conf. Recent Trends in Information Systems,* pp. 184-187, July 2006.

[50] M.C. Padma and P. Nagabhushan, "Identification and Separation of Text Words of Kannada, Hindi and English Languages through Discriminating Features," *Proc. Nat'l Conf. Document Analysis and Recognition,* pp. 252-260, July 2003.

[51] R. Kumar, V. Chaitanya, and C.V. Jawahar, "A Novel Approach to Script Separation," *Proc. Int'l Conf. Advances in Pattern Recognition,* pp. 289-292, Dec. 2003.

[52] K. Roy, U. Pal, and B.B. Chaudhuri, "Address Block Location and Pin Code Recognition for Indian Postal Automation," *Proc. Workshop Computer Vision, Graphics, and Image Processing,* pp. 5-9, Feb. 2004.

[53] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, and A. Belaid, "A System for Indian Postal Automation," *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 2, pp. 1060-1064, Aug./Sept. 2005.

[54] K. Roy, D. Pal, and U. Pal, "Pin-Code Extraction and Recognition for Indian Postal Automation," *Proc. Nat'l Conf. Recent Trends in Information Systems,* pp. 192-195, July 2006.

[55] K. Roy and U. Pal, "Word-Wise Hand-Written Script Separation for Indian Postal Automation," *Proc. Int'l Workshop Frontiers in Handwriting Recognition,* pp. 521-526, Oct. 2006.

[56] K. Roy, U. Pal, and B.B. Chaudhuri, "Neural Network Based Word-Wise Handwritten Script Identification System for Indian Postal Automation," *Proc. Int'l Conf. Intelligent Sensing and Information Processing,* pp. 240-245, Jan. 2005.

[57] S.L. Wood, X. Yao, K. Krishnamurthi, and L. Dang, "Language Identification for Printed Text Independent of Segmentation," *Proc. Int'l Conf. Image Processing,* vol. 3, pp. 428-431, Oct. 1995.

[58] T.N. Tan, "Rotation Invariant Texture Features and Their Use in Automatic Script Identification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 7, pp. 751-756, July 1998.

[59] L. O'Gorman and R. Kasturi, *Document Image Analysis.* IEEE CS Press, 1995.

[60] G.S. Peake and T.N. Tan, "Script and Language Identification from Document Images," *Proc. Asian Conf. Computer Vision,* pp. 97-104, Jan. 1998.

[61] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 3, no. 6, pp. 610-621, Nov. 1973.

[62] W.M. Pan, C.Y. Suen, and T.D. Bui, "Script Identification Using Steerable Gabor Filters," *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 2, pp. 883-887, Aug./Sept. 2005.

[63] V. Singhal, N. Navin, and D. Ghosh, "Script-Based Classification of Hand-Written Text Documents in a Multilingual Environment," *Proc. Int'l Workshop Research Issues in Data Eng.—Multi-Lingual Information Management,* pp. 47-54, Mar. 2003.

[64] J. Cheng, X. Ping, G. Zhou, and Y. Yang, "Script Identification of Document Image Analysis," *Proc. Int'l Conf. Innovative Computing, Information, and Control,* vol. 3, pp. 178-181, Aug./Sept. 2006.

[65] A.K. Jain and Y. Zhong, "Page Segmentation Using Texture Analysis," *Pattern Recognition,* vol. 29, no. 5, pp. 743-770, May 1996.

[66] A. Busch, W.W. Boles, and S. Sridharan, "Texture for Script Identification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 11, pp. 1720-1732, Nov. 2005.

[67] A. Busch, "Multi-Font Script Identification Using Texture-Based Features," *Proc. Int'l Conf. Image Analysis and Recognition,* pp. 844-852, Sept. 2006.

[68] G.D. Joshi, S. Garg, and J. Sivaswamy, "Script Identification from Indian Documents," *Proc. IAPR Int'l Workshop Document Analysis Systems,* pp. 255-267, Feb. 2006.

[69] W. Chan and G.G. Coghill, "Text Analysis Using Local Energy," *Pattern Recognition,* vol. 34, no. 12, pp. 2523-2532, Dec. 2001.

[70] H. Ma and D. Doermann, "Gabor Filter Based Multi-Class Classifier for Scanned Document Images," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 968-972, Aug. 2003.

[71] S. Jaeger, H. Ma, and D. Doermann, "Identifying Script on Word-Level with Informational Confidence," *Proc. Int'l Conf. Document Analysis and Recognition,* vol. 1, pp. 416-420, Aug./Sept. 2005.

[72] D. Dhanya, A.G. Ramkrishnan, and P.B. Pati, "Script Identification in Printed Bilingual Documents," *Sadhana,* vol. 27, no. 1, pp. 73-82, Feb. 2002.

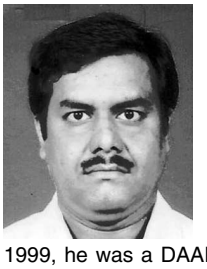
[73] D. Dhanya and A.G. Ramkrishnan, "Script Identification in Printed Bilingual Documents," *Proc. IAPR Int'l Workshop Document Analysis Systems,* pp. 13-24, Aug. 2002.

[74] D. Dhanya and A.G. Ramkrishnan, "Optimal Feature Extraction for Bilingual OCR," *Proc. IAPR Int'l Workshop Document Analysis Systems,* pp. 25-36, Aug. 2002.

[75] P.B. Pati, S. Sabari Raju, N. Pati, and A.G. Ramakrishnan, "Gabor Filters for Document Analysis in Indian Bilingual Documents," *Proc. Int'l Conf. Intelligent Sensing and Information Processing,* pp. 123-126, Jan. 2004.

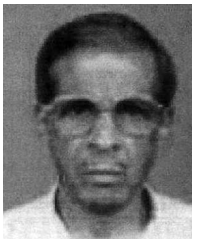
[76] P.B. Pati and A.G. Ramakrishnan, "HVS Inspired System for Script Identification in Indian Multi-Script Documents," *Proc. Int'l Workshop Document Analysis Systems,* pp. 380-389, Feb. 2006.

[77] A.L. Spitz, "Script and Language Determination from Document Images," *Proc. Ann. Symp. Document Analysis and Information Retrieval,* pp. 229-235, Apr. 1994.
[78] J.J. Lee, B.K. Sin, and J.H. Kim, "On-Line Mixed Character Recognition Using an HMM Network," *Proc. KISS Ann. Conf.,* vol. 20, no. 2, pp. 317-320, Oct. 1993.
[79] J.J. Lee, J.H. Kim, and M. Nakajima, "A Hierarchical HMM Network-Based Approach for On-Line Recognition of Multi-Lingual Cursive Handwritings," *IEICE Trans. Information and Systems,* vol. E81-D, no. 8, pp. 881-888, Aug. 1998.
[80] A.M. Namboodiri and A.K. Jain, "Online Script Recognition," *Proc. Int'l Conf. Pattern Recognition,* vol. 3, pp. 736-739, Aug. 2002.
[81] A.M. Namboodiri and A.K. Jain, "Online Handwritten Script Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 124-130, Jan. 2004.
[82] A. Malaviya and L. Peters, "Fuzzy Handwriting Description Language: FOHDEL," *Pattern Recognition,* vol. 33, no. 1, pp. 119-131, Jan. 2000.
[83] J. Gllavata and B. Freisleben, "Script Recognition in Images with Complex Backgrounds," *Proc. IEEE Int'l Symp. Signal Processing and Information Technology,* pp. 589-594, Dec. 2005.
[84] B.B. Chaudhuri, "On Multi-Script OCR System Evaluation," *Proc. Int'l Workshop Performance Evaluation Issues in Multi-Lingual OCR,* http://www.kanungo.com/workshop/abstracts/chaudhuri.html, Sept. 1999.
[85] T. Kanungo, P. Resnik, S. Mao, D.-W. Kim, and Q. Zheng, "The Bible and Multilingual Optical Character Recognition," *Comm. ACM,* vol. 48, no. 6, pp. 124-130, June 2005.



Debashis Ghosh received the BE degree in electronics and communication engineering from M.R. Engineering College, Jaipur, India, in 1993, and the MS and PhD degrees in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1996 and 2000, respectively. He is currently an associate professor in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee. From April 1999 to November 1999, he was a DAAD research fellow at the University of Kaiserslautern, Germany. In November 1999, he joined the Indian Institute of Technology Guwahati, as an assistant professor of electronics and communication engineering. He spent the 2003-2004 academic year as a visiting faculty member in the Department of Electrical and Computer Engineering at the National University of Singapore. Between 2006 and 2008, he was a senior lecturer with the Faculty of Engineering and Technology, Multimedia University, Malaysia. His teaching and research interests include image/video processing, computer vision, and pattern recognition.

Tulika Dube received the BTech degree in electronics and communication engineering from the Indian Institute of Technology Guwahati in 2006. Soon after her graduation, she joined the Indian Division of British Telecom at Bangalore, and later moved to Ibibo Web Pvt. Ltd., Gurgaon, India, as a software engineer. Between 2007 and 2009, she worked as a senior software engineer with Infovedics Software Pvt. Ltd., Noida, India. She received a search developer certification from FAST University, Norway, in 2007. She is currently working toward the management degree at the Indian Institute of Management, Ahmedabad.



Adamane P. Shivaprasad received the BE, ME, and PhD degrees in electrical communications engineering from the Indian Institute of Science, Bangalore, in 1965, 1967, and 1972, respectively. He is currently a guest professor in the Department of Electronics and Communication Engineering, Sambhram Institute of Technology, Bangalore, India. He was a member of the academic staff of the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, from 1967 until he retired as a professor in 2006. His research interests include design of micropower VLSI circuits, intelligent instrumentation, communication systems, and pattern recognition.