Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2014, Article ID 896128, 12 pages
http://dx.doi.org/10.1155/2014/896128

Research Article

Script Identification from Printed Indian Document Images and Performance Evaluation Using Different Classifiers

Sk Md Obaidullah,[1] Anamika Mondal,[2] Nibaran Das,[3] and Kaushik Roy[2]

[1] *Department of Computer Science & Engineering, Aliah University, Kolkata, India*
[2] *Department of Computer Science, West Bengal State University, Barasat, India*
[3] *Department of Computer Science & Engineering, Jadavpur University, Kolkata, India*

Correspondence should be addressed to Sk Md Obaidullah; sk.obaidullah@gmail.com

Received 18 June 2014; Accepted 18 November 2014; Published 7 December 2014

Academic Editor: Erich Peter Klement

Identification of script from document images is an active area of research under document image processing for a multilingual/multiscript country like India. In this paper the real life problem of printed script identification from official Indian document images is considered and performances of different well-known classifiers are evaluated. Two important evaluating parameters, namely, AAR (average accuracy rate) and MBT (model building time), are computed for this performance analysis. Experiment was carried out on 459 printed document images with 5-fold cross-validation. Simple Logistic model shows highest AAR of 98.9% among all. BayesNet and Random Forest model have average accuracy rate of 96.7% and 98.2% correspondingly with lowest MBT of 0.09 s.

1. Introduction

Automatic script identification is an active area of research under document image processing. The work is particularly relevant for a multiscript country like India. Right now there are officially 22 languages and 13 scripts [1] are used to write those languages. With English the figure becomes 23. Automatic document processing helps conversion of physical real world document into digital text form, which can be very much useful for further processing like storing, retrieval, and indexing of large volume of data. In our country there are many languages which use the same script for writing. For example, Devanagari is a well-known script in India which is used to write languages like Hindi, Marathi, Sanskrit, and so forth whereas Bangla is another popular script and is used to write languages like Bangla, Assamese, and Manipuri. Multilingual document is very common in our daily life which includes postal document, pre-printed application form, and so forth. Optical Character Recognizer (OCR) for specific language will not work for such multilingual documents. Therefore, to make a successful multilingual OCR, script identification is very essential before running an individual OCR for a specific language. In this context, the problem of script identification is addressed here.

All the script identification techniques under printed category can be divided into four major groups, namely, (i) document level script identification, (ii) block level script identification, (iii) line level script identification, and (iv) word level script identification. Document level script identification is much faster than the other category because here the whole document is fed to the script identification system without performing fine segmentation into block, line, or word level. Ghosh et al. [2] presented a review where techniques developed by the document image processing researchers for script identification from printed and handwritten document are mentioned. Few works are reported in literature on printed Indic script identification. Sometimes non-Indic scripts are also considered in the database along with Indic scripts. Among those, Spitz [3] in his work identified Latin, Han, Chinese, Japanese, and Korean scripts by using features like upward concavity distribution, optical character density, and so forth. He carried out his work at

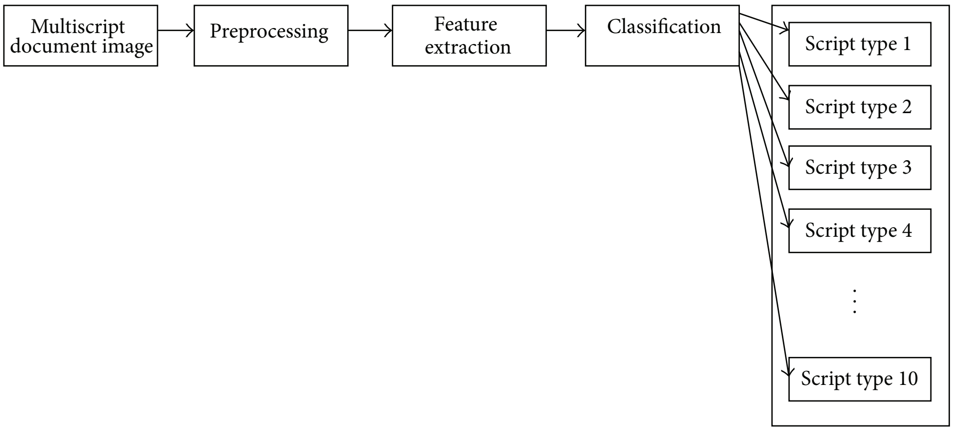


Figure 1: Block diagram of the present work.

document level. Lam et al. [4] identified some non-Indic scripts using horizontal projection profile, height distribution, presence of circles, ellipse, presence of vertical stroke, and so forth features. Hochberg et al. [5] identified six scripts, namely, Arabic, Armenian, Devanagari, Chinese, Cyrillic, and Burmese, using some textual symbol based features. Zhou et al. [6] identified Bangla and English scripts using connected component based features from both printed and handwritten document. Prasad et al. [7] identified printed Gujarati script using cluster based templates. Patil and Subbareddy [8] proposed a triscript identification technique on English, Kannada, and Hindi using neural network based classification technique. They performed their work at word level. Elgammal and Ismail [9] proposed a block level and line level script identification technique from Arabic and English scripts using horizontal projection profiles and run length histograms analysis. Dhandra et al. [10] proposed a word level script identification technique from Kannada, Hindi, English, and Urdu using morphological analysis. Chaudhuri and Pal [11] proposed line based script identification techniques from Roman, Bangle, and Devanagari scripts. Tan et al. [12] proposed mixed script identification techniques considering Chinese, Latin, and Tamil using upward concavity based features. Chaudhury and Sheth [13] proposed Gabor filter based script identification techniques from English, Hindi, Telugu, and Malayalam scripts. They performed the work at block level. In another work Padma and Vijaya [14] proposed a work using wavelet transform based feature considering seven Indic and non-Indic scripts, namely, English, Chinese, Greek, Cyrillic, Hebrew, Hindi, and Japanese. Using multichannel log Gabor filter based features Joshi et al. [15] proposed a block level script identification technique from English, Hindi, Telugu, Malayalam, Gujarati, Kannada, Gurumukhi, Oriya, Tamil, and Urdu scripts. Dhanya et al. [16] proposed a word level script identification technique from Roman and Tamil scripts using multichannel Gabor filters and discrete cosine transform (DCT) based feature.

Few works are reported in literature considering handwritten document images also. Among the pieces of work Zhou et al. [6] identified Bangla and English scripts using connected component profile based features. They performed the work at line, word, and character level. Singhal et al. [20] identified Roman, Devanagari, Bangla, and Telugu scripts from line level handwritten document images. They used texture classification algorithm for their work. Hochberg et al. [21] identified six Indic and non-Indic scripts, namely, Arabic, Chinese, Cyrillic, Devanagari, Japanese, and Latin, using some features like horizontal and vertical centroids, sphericity, aspect ratio, white holes, and so forth. They performed the work at document level. In another work Roy et al. [22] identified six popular Indic scripts, namely, Bangla, Devanagari, Malayalam, Urdu, Oriya, and Roman, using component based features, fractal dimension based features, circularity based features, and so forth. This is the first kind of work involving six Indic scripts altogether. In a block level script identification technique Basu et al. [23] identified Latin, Devanagari, Bangla, and Urdu handwritten numeral scripts using similar shaped digit pattern based features. Singhal et al. [24] proposed a technique to identify four Indic scripts, namely, Devanagari, Bangla, Telugu, and Latin. They used rotation invariant texture features using multichannel Gabor filtering and gray level cooccurrence matrix. Using fractal based features Moussa et al. [25] identified Arabic and Roman scripts from line level handwritten document. In a very recent work Hangarge et al. [26] proposed a word level scheme based on directional DCT based feature to identify six Indic scripts, namely, Roman, Devanagari, Kannada, Telugu, Tamil, and Malayalam. Rani et al. [27] provide a technique using Gabor filter and gradient based features and SVM classifier to identify Gurumukhi and Roman scripts. This work was done at character level.

All the above mentioned works are performed using only one standard model for classification purpose. It is also observed that till date no work is carried out considering all official Indic scripts. The present work is an attempt to identify any one of the ten popular Indic scripts and to evaluate the performance of different well-known classifiers with respect to different standard measuring parameters. Figure 1 shows a block diagram of the proposed model for the present work.

The paper is organized as follows: a brief overview about Indic languages and scripts is provided in Section 2. In Section 3 data collections and preprocessing are discussed. Section 4 deals with feature extraction techniques.

Experimental results are discussed in Section 5. Conclusion is in Section 6 and Acknowledgment section follows Conclusion section. Finally references are available in the last section.

2. Indic Languages and Scripts

India is a country with diversified culture, community, religion, and languages. There are total 22 official languages [1] in India and 13 scripts are used to write them. With English the count becomes 23. In the following section a brief outline about official Indic languages and scripts is provided. In Figure 2 a geographical map showing different languages and scripts for different states is provided.

2.1. Roman Script. It is used to write English language which is international language. This script is a descendant of the ancient Proto-Indo-European language family. About 328 million people in our country use this language as a communication medium.

It is also used to write Santali language, which is under Austro-Asiatic language family. About 6.2 million people living in different parts of eastern India mainly use this language.

2.2. Devanagari Script. Hindi is the one of the most popular languages in India which uses this script. This language is under Indo-European language family. In India about 182 million people mainly residing in northern part use this language as their communication medium.

Marathi language is under the Indo-European language family. About 68.1 million Indian people use this language. Marathi is the state language of Maharashtra.

About 0.5 million of Indian people live mainly in southern Assam, few areas of Manipur, Meghalaya, Jalpaiguri, Cooch Behar, and Darjeeling districts of West Bengal. This language is under Sino-Tibetan group of language family.

Konkani belongs to the Indo-European language family. About 7.6 million people use this language. It is the official language of Goa.

Sanskrit is under the family of Indo-European group of language. Sanskrit is mainly used as the liturgical language. About 0.03 million of Indian people use this language.

Sindhi language is under the Indo-European language family. Sindhi is used by many people residing in Madhya Pradesh, Andhra Pradesh, Uttar Pradesh, Gujarat, Tamil Nadu, Maharashtra, Rajasthan, Delhi, Bihar, and Orissa. About 21.4 million people use this language.

Nepali language belongs to the Indo-European language group. About 13.9 million people living in eastern part of India use this language as their communication medium.

Maithili has its origin in the Indo-European language family. The language is mostly used by people living in Bihar. About 34.7 million people use this language.

2.3. Bangla Script. Bangla language is one of the most popular languages in India. It is spoken by 181 million of population of India living mainly in the state of West Bengal. The language is originated from the Indo-European language family.

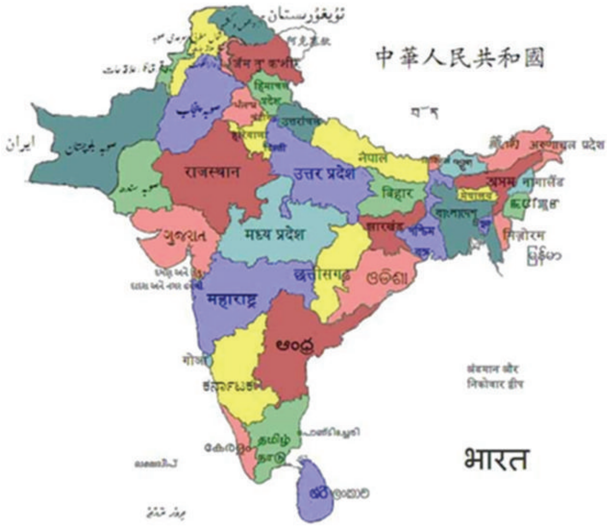


Figure 2: A map showing different languages and scripts for different states [17].

Assamese can be classified under the Indo-European group of languages. It is the state language of Assam. About 16.8 million of the Indian population use this language.

Manipuri originated from the Sino-Tibetan language group. People of Manipur mostly use the language. It is also used in different parts of Assam, Tripura, and so forth. About 13.7 million people use this language.

2.4. Telugu Script. This script is used by Telugu language and is classified under the Dravidian group of language family. It is the state language of Andhra Pradesh. About 69.8 million of Indian people residing in Andhra Pradesh and nearby states use this language as their communication medium.

2.5. Tamil Script. Tamil language uses Tamil script for writing. It is the state language of Tamil Nadu. This language is originated from the Dravidian group of languages. About 65.7 million of Indian population use Tamil as a medium of communication.

2.6. Urdu Script. Urdu script is used to write Urdu language. This language is originated from the Indo-European group of language family. It is the state language of Jammu and Kashmir. About 60.6 million of Indian population use this language as their medium of communication.

2.7. Gujarati Script. Gujarati language is one of the most popular languages in India. About 46.5 million of Indian people mainly residing in the areas like Gujarat, Maharashtra, Rajasthan, and Madhya Pradesh use this language as their medium of communication. Gujarati language falls under Indo-European language family group.

2.8. Malayalam Script. About 35.9 million of Indian people living mainly in the states of Kerala and nearby sides use this language. This language belongs to the Dravidian language family group.

2.9. Oriya Script. Oriya script is used by Oriya language. This language is originated from the Indo-European group of language family. Oriya is also state language of the state Orissa. About 31.7 million of population living mainly in eastern part of India use this language as their communication medium.

2.10. Kashmiri Script. Kashmiri language uses Kashmiri script for writing and it is originated from the Indo-European language family. About 5.6 million people living in Jammu and Kashmir, Kashmir Valley, Punjab, Delhi, and Uttar Pradesh use this language as a medium of communication.

2.11. Dogri Script. Dogri language falls under Indo-European language family. About 3.8 million of people living in the area of Jammu and Kashmir, Chandigarh, and West Bengal use this language as their medium of communication.

2.12. Kannada Script. This script is used by Kannada language. About 3.63 million of Indian people mainly residing in the states of Andhra Pradesh, Tamil Nadu, and Maharashtra use this language as their communication medium. This language belongs to the Dravidian group of language family.

2.13. Gurumukhi Script. This script is used by Punjabi language which belongs to Indo-European language group of language family. This is the state language of Punjab and about 1.05 million Indian people use this language as their communication medium (Table 1).

Table 1: Official language of India with total speaker and its script [1].

Serial number	Language	Speaker (M)	Script	Writer (M)
1	Assamese	16.8	Bangla	211.50
2	Bangla	181		
3	Manipuri	13.7		
4	Bodo	0.5	Devanagari	328.23
5	Hindi	182		
6	Konkani	7.6		
7	Maithili	34.7		
8	Marathi	68.1		
9	Nepali	13.9		
10	Sanskrit	0.03		
11	Sindhi	21.4		
12	Santhali	6.2	Roman	334.20
13	English	328		
14	Dogri	3.8	Dogri	03.80
15	Gujarati	46.5	Gujarati	46.50
16	Kannada	3.63	Kannada	03.63
17	Kashmiri	5.6	Kashmiri	05.60
18	Malayalam	35.9	Malayalam	35.90
19	Oriya	31.7	Oriya	31.70
20	Punjabi	1.05	Gurmukhi	01.05
21	Tamil	65.7	Tamil	65.70
22	Telugu	69.8	Telugu	69.80
23	Urdu	60.6	Urdu	60.60

3. Data Collection and Preprocessing

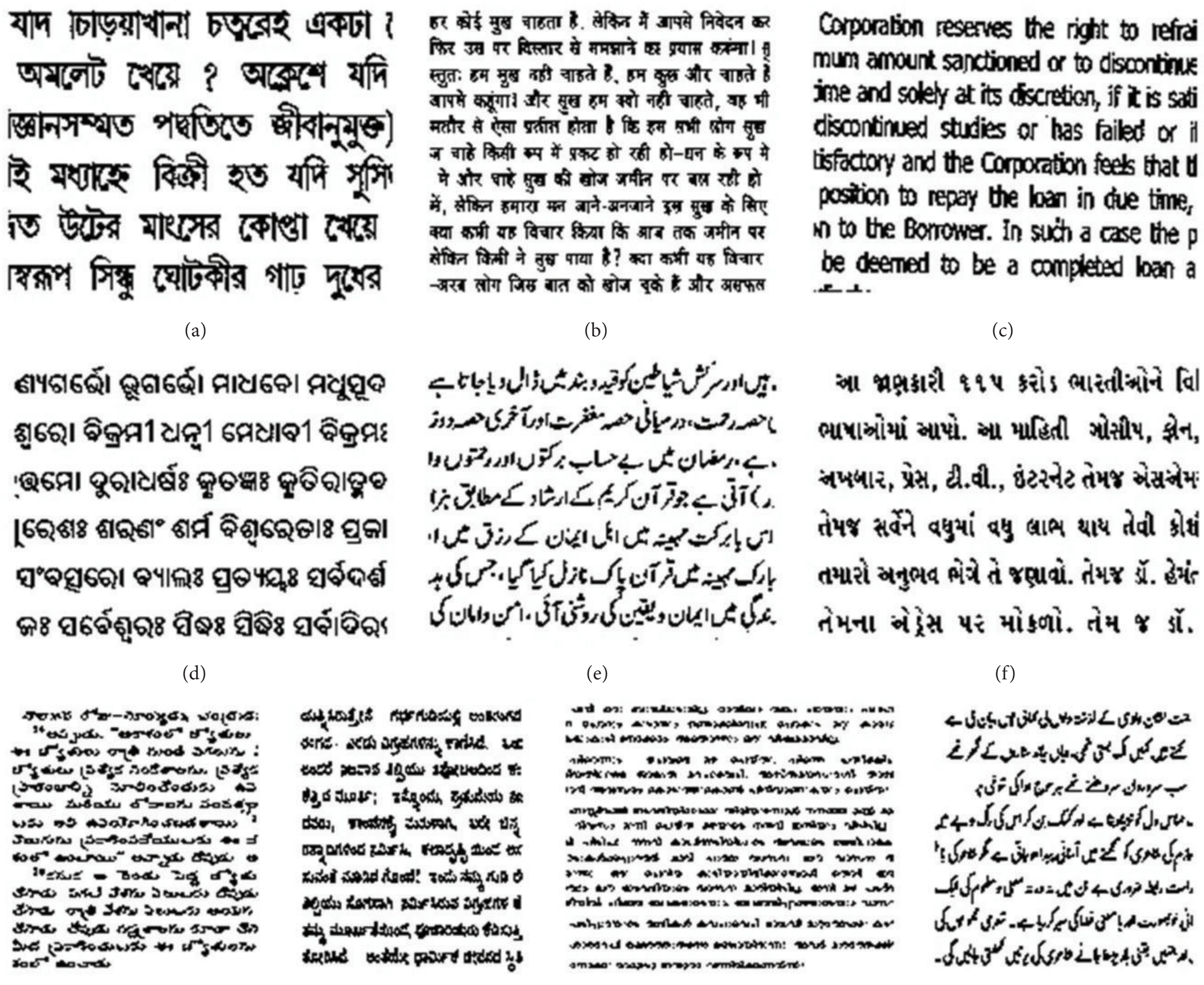
Availability of standard database is one of the most important issues for any pattern recognition research work. Till date no standard database is available for all official Indic scripts. Real life printed script data are collected from different sources like book pages, articles, and so forth. Total 459 printed document pages are collected from the above mentioned sources. Then collected documents are digitized using HP flatbed scanner. Out of 459 document pages 60 Bangla, 60 Devanagari, 60 Roman, 58 Gujarati, 20 Oriya, 60 Telugu, 60 Kannada, 22 Kashmiri, 29 Malayalam, and 30 Urdu script images are taken. Figure 3 shows sample script images from our database.

Initially the images are in gray tone and digitized at 300 dpi using a flatbed HP scanner. After digitization preprocessing was carried out. A two-stage based approach is used to convert the images into binary (0 and 1) or two tone images. At first stage prebinarization [18] is done using a local window based algorithm in order to get an idea of different region of interest (ROI). Then run length smoothing approach (RLSA) is applied on the prebinarized image. This will overcome the limitations of the local binarization method used. The stray/hollow regions created due to fixed window size are converted into a single component. Finally, using component labeling, each component is selected and mapped in the original gray image to get respective zones of the original image. The final binary image is obtained by applying a histogram based global binarization algorithm [18] to these regions/components of the original image.

After preprocessing feature extraction process is carried out to construct the feature vector. Major features considered for the present work are discussed in Section 4.

4. Feature Extraction

Feature extraction is the identification of appropriate and unique characteristics of the component of images. After the preprocessing of the input script images is done, next phase is to carry out the extraction and selection of different features. It is a very crucial phase for the recognition system. Computation of good features is really a challenging task. The term “good” signifies the set of features which are robust enough to capture the maximum variability among inter-classes and the minimum variability within the intra-classes and still computationally easy. First, visual observations are made on Indic scripts to study the nature of different graphemes of different scripts. Present work is done focusing on the features based on structure of the image components. Besides these few mathematical and morphological features are also computed observing their usefulness on Indic scripts.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3: Sample from our database of (a) Bangla, (b) Devanagari, (c) Roman, (d) Oriya, (e) Urdu, (f) Gujarati, (g) Telugu, (h) Kannada, (i) Malayalam, and (j) Kashmiri script documents.

Intel OpenCV (Open Source Computer Vision) library [19] has been used in this process to extract those features.

4.1. Structural Feature. Structure or shape analysis is a global measurement of an image component which is used as an important feature for present experiment. Initially inner and outer contours of the component are computed; then different structural features like circularity, rectangularity, convexity, chain code, and so forth are calculated for each component contour. Component analysis to find the presence of small component is very useful to identify scripts like Urdu, where number of small components prevails very much more than other scripts. Freeman chain code direction histogram is quite useful to distinguish scripts like Bangla, Devanagari due to their nature of using “Matra” or “Shirorekha” (Figure 4).

Circularity feature is very useful to identify some scripts like Malayalam, Oriya due to their circular nature. This feature can be computed globally on all the scripts to study which scripts are most circular and which are least. These global values are stored in the feature vector.

Some of these features are discussed in the following subsections.

4.1.1. Component Analysis. Dimensionality is an important measure in component analysis. In the present work image components are classified into three major categories, namely, (i) LC (large component), (ii) MC (medium component), and (iii) SC (small component). Different component sizes are computed based on these categories and these values are stored in the feature table (see Algorithm 1). An algorithm for computation of component dimensionality is provided in the following subsection where default threshold value is considered as 5.

It is observed that among the ten scripts considered for the present work Urdu script prevails as smaller component than other scripts considered.

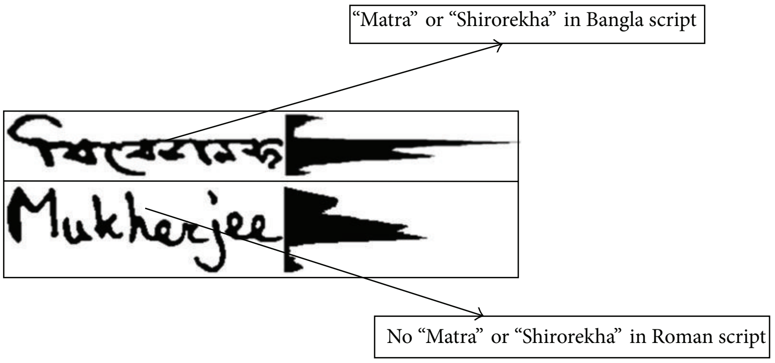

"Matra" or "Shirorekha" in Bangla script
Mukherjee
No "Matra" or "Shirorekha" in Roman script

Figure 4: Presence of "Matra" or "Shirorekha" in Bangla script: the same is absent in Roman [18].

```
Initially Set SC = 0;
Using component analysis each component is considered and pixel count is done.
        If Number Of Pixel (NOP) <= Predefined Threshold
        SC++;
End
```

Algorithm 1: Algorithm for computation of small component.

4.1.2. Chain Code Based Feature. Presence of different directional strokes (horizontal/vertical/left or right diagonal) in the scripts is important feature for identification. "Matra" or "Shirorekha" is a horizontal line present on the upper part of Bangla and Devanagari script which is a distinguishing feature. We have used cvFindContours() function in OpenCV [19] in CV_CHAIN_CODE mode for identifying these lines as a sequence of integers shown in Figure 5. A left to right directional horizontal line present in a script will generate a series of "2" by the cvFindContours() function. An example of chain code generated by OpenCV is shown in the same Figure.

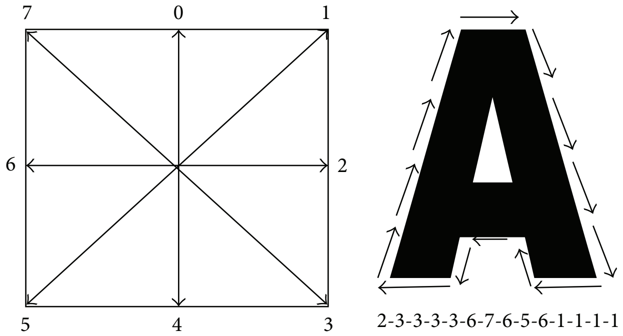


Figure 5: An example of 8-directional freeman chain code [19].

4.1.3. Component Circularity Based Feature. One of the key features in the structural category is circularity of a component. Graphemes of many scripts like Oriya, Malayalam, and so forth have more circular nature than others. Following is the algorithm for calculation of circularity of a component.

The following algorithm computes circularity of an image component.

(i) At first, minimum enclosing circle is drawn. This enclosing circle will cover the component minimally. The radius of the enclosing circle is stored in a variable say $R1$.

(ii) Then circle fitting is done. This operation will fit a circle in the component in as minimum manner as possible. Radius of the fitted circle is stored in a variable, say $R2$.

(iii) The differences of the two radiuses $R1$ and $R2$ are stored in a variable, say R. This value of R indicates the proximity of circularity of the component. In ideal case the value of R will be zero which means absolute circular component.

In fact the complete or almost complete circular components will have zero difference between the two radii or will have a difference tending to zero (Figure 6).

4.1.4. Component Rectangularity. Another dimensionality measurement is the rectangularity of a component for each script. The bounding rectangle is drawn on inner and outer contour for each of the image components and the ratio of the length of the height and width is measured to determine whether the component is square ($h/w = 1$), horizontal ($h/w < 1$), or vertical ($h/w > 1$). These global measurements are stored and used as an important feature (Figure 7).

4.1.5. Convex Hull. Convex hull was computed to comprehend the shape of the components. The hull is computed for every selected component's inner and outer contours in the

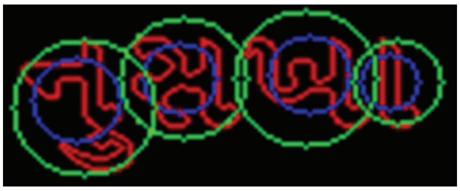


Figure 6: Computation of circularity of component on Gujarati script using fitted circles (blue: minimum encapsulating and green: best fitted).

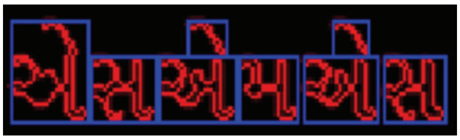


Figure 7: Computation of rectangularity of component on Gujarati script showing blue: rectangular box.

proposed method. Minimum and maximum of surrounding of inner and outer contour of the component was computed. Their average values and variance are also calculated. In OpenCV [19] the function cvConvexHull() takes an array of points and puts out indices of points that are convex hull vertices (Figure 8).

4.2. Other Important Features. For the present work emphasis was given to collect features based on the structure of the component. Besides the structural features few other features based on Gabor filter and morphological reconstruction are also computed. Gabor filter is proven successful for texture analysis whereas morphological reconstruction with different user defined kernels is used to capture different directional strokes presence in different scripts. Gabor features were extracted to construct a filter bank with varying orientations. The orientation angles are chosen experimentally. Morphological features are computed using different user defined kernels of horizontal and vertical and left and right diagonal types.

5. Experiments

One of the main objectives of the present work is to analyze the performance of different classifiers for a real life script identification problem. Weka [28] is a popular machine learning software used in the present work for classification of different scripts after computation of all the features. In this machine learning software a very simple easy to use GUI interface is provided to perform different tasks. It contains various tools for different applications like classification, clustering, data processing, regression analysis, and so forth. For present work Bayesian, functional, rule based, and tree based classifiers are considered. The main evaluating parameters considered for performance measurements of those classifiers are average accuracy rate (AAR) and model building time (MBT). Following section provides brief outlines about the classifiers considered.

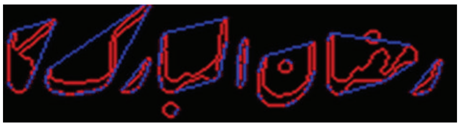


Figure 8: Computation of convex hull on Urdu script (blue: convex hull).

5.1. Experimental Protocol and Classifiers. During experimentation k-fold cross-validation is followed. Here 459 sample images are initially divided into k different subsets. Out of the k subsets, one subset is kept for the validation data for testing the model and the remaining subsets ($k-1$ number) are used as training data. This process is repeated then k times or k folds, where each of the $k-1$ subsets is used as the validation test data. In our experimentation the value of k was chosen empirically as 5 (Figure 9).

5.1.1. Bayesian Classifier

(i) BayesNet. Popular Bayesian classifier uses Bayes network learning using different search algorithms and quality parameters [29]. The base class of this classifier provides data structures (conditional probability distributions, network structure, etc.) and facilities common to Bayes network learning algorithms like K2 and B. Experimentally BayesNet gives an average accuracy rate of 96.7%. It is very fast to build the model at 0.09 s.

5.1.2. Functional Classifier

(i) LibLINEAR. LibLINEAR is a good linear classifier based on functional model for data with large number of instances or features. It has converged faster for our dataset than other classifiers of Weka. We have used the L2-loss support vector machine (dual) as the SVM type parameters of the LIBLINEAR and both the bias and cost parameters are set to 1.0. The EPS (the tolerance of the termination criterion) is 0.01. For more details see [30]. Average accuracy found by LibLINEAR is 97.6% and builds the model in 0.38 s.

(ii) MLP. MLP [18] is a classifier that uses backpropagation algorithm to classify instances. It is a layered feedforward network which can be represented by a DAG (directed acyclic graph). Each node of an MLP is termed as an artificial neuron. The weights/labels given in each directed arc represent the strength/capacity of synaptic connection between two neurons and the direction of the signal flow. In MLP there is an input and output layer. The number of neurons in input layer is the same as the number of features selected for the particular pattern recognition problem, whereas the number of output layers is the same as the number of target classes. The neurons in hidden and output layers compute the sigmoidal function on the sum of the products of inputs and weights of the corresponding connections to each neuron. Training process of an MLP involves tuning the strengths of its synaptic connections such that the MLP can respond properly to every input value taken from the training set. The

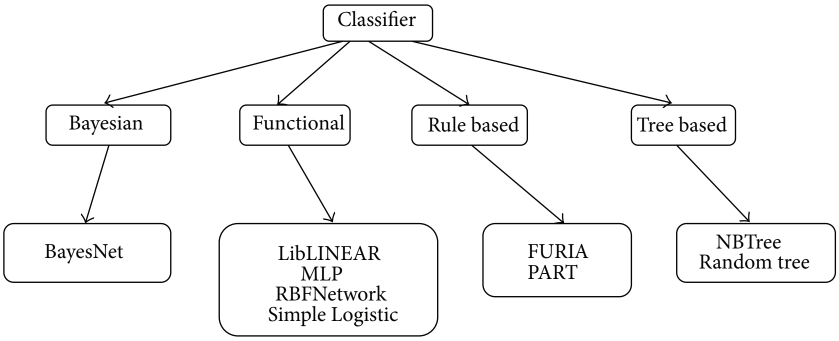


Figure 9: Classifier hierarchy considered for present work.

total number of hidden layers and the number of neurons present in each hidden layer should be determined during training process.

For this work, the feature is 62-dimensional and the number of scripts is 10, so the number of neurons in input layer and output layer is 62 and 10, respectively. The number of neurons for the hidden layer is chosen automatically (the default value) by the MLP classifier of the Weka [28]. The average accuracy rate found using MLP is 98.4% without any rejection. The time to build up for the classifier is a bit higher than others which is 35.3 seconds.

(iii) RBFNetwork. In radial basis function (RBF) networks for hidden layer processing elements the static Gaussian function has been used as the nonlinearity. The function works in a small centered region of the input space [31]. The implementation of the network depends on the centers of the Gaussian functions [32, 33]. The main functionality depends on how the Gaussian centers are derived and they act as weights of input to hidden layer. The widths of the Gaussians are calculated depending on the centers of their neighbors. The faster convergence criterion is one of the advantages of this network. This is because it only updates weights from hidden to output layer. All the parameters for RBF network classifier of Weka tool are set to its default values for this work like the MINSTDDEV (minimum standard deviation) has been set to 0.1. Average accuracy found by this classifier is 95.6%. The model builds in 5.9 s.

(iv) Simple Logistic. It is a classifier for building linear logistic regression model [14]. Here LogitBoost is used with simple regression functions as base learner for fitting the logistic model. The optimal number of LogitBoost iterations to perform is cross-validated here, which helps in the selection of automatic attribute. This classifier gives highest accuracy rate among all in our experiment which is 98.9% as it was found in some earlier work also [22]. Time taken to build the model is 8.36 s.

5.1.3. Rule Based Classifier

(i) FURIA. Fuzzy Unordered Rule Induction Algorithm (FURIA) is a fuzzy-rule-based classifier, used to obtain fuzzy rules. FURIA has recently been developed as an extension of the well-known RIPPER algorithm. Instead of conventional rules and rule lists it learns fuzzy rules and unordered rule sets. Furthermore it uses an efficient rule stretching scheme to deal with uncovered examples [34]. All the parameters for FURIA classifier of Weka tool are set to its default values for this work like the MINNO (minimum total weight of the instances in a rule) has been set to 2.0. Average accuracy found here is 93.8% and the model is built very fast in 0.69 s.

(ii) PART. It is a class for generating a PART decision list. It uses separate-and-conquer method. Then it builds a partial C4.5 decision tree during each iteration and makes the best leaf into a rule [28]. Average accuracy rate found using PART is 91.7 and the model takes time of 0.19 s.

5.1.4. Tree Classifier

(i) NBTree. This is a tree based classifier under Weka [28]. It contains class for generating a decision tree with naive Bayes classifiers at the leaves. This tree based classifier shows an average accuracy rate of 93.8%.

(ii) Random Forest. Random Forest (RF) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. For more detail refer to [35]. Using RF we obtain 91.1% average accuracy rate. The performance of Random Forest classifier is very promising in our experiment. An average accuracy rate of 98.2% is found and the model is built in 0.09 s which is fastest among all.

Table 2 provides comparison of different classifiers based on two parameters AAR and MBT. It has been found that Simple Logistic classifier which is under the category of functional classifier performs best amongst all. For present experiment 98.9% average accuracy is obtained. For MBT BayesNet and Random Forest perform best compared to all. Table 3 shows the confusion matrix using Simple Logistic classifier where highest accuracy rate of 98.9% was found (Tables 2 and 3, and Figures 10, 11, 12, and 13).

Sample Misclassified Instances. Few of the sample misclassified instances are shown for different classifiers. Figure 12 shows misclassified instance obtained using different classifiers where in (a) using Simple Logistic classifier Oriya

Table 2: Comparison of result for different classifiers: AAR: average accuracy rate, MBT: model building time.

Type	Classifier	AAR (%)	MBT (s)
Bayesian	BayesNet	96.7	**0.09**
Functional	LibLINEAR	97.6	0.38
	MLP	98.4	35.3
	RBFNetwork	95.6	5.9
	Simple Logistic	**98.9**	8.36
Rule based	FURIA	93.8	0.69
	PART	91.7	0.19
Tree based	NBTree	93.8	31.01
	Random Forest	98.2	**0.09**

Table 3: Confusion matrix for Simple Logistic classifier where highest accuracy rate was found.

Classified as	B	D	R	G	O	Te	Ka	Ks	M	U
B	60	0	0	0	0	0	0	0	0	0
D	0	59	0	0	0	0	0	0	1	0
R	0	2	57	0	1	0	0	0	0	0
G	0	0	0	58	0	0	0	0	0	0
O	0	0	0	0	19	0	0	0	1	0
Te	0	0	0	0	0	60	0	0	0	0
Ka	0	0	0	0	0	0	60	0	0	0
Ks	0	0	0	0	0	0	0	22	0	0
M	0	0	0	0	0	0	0	0	29	0
U	0	0	0	0	0	0	0	0	0	30

B: Bangla, D: Devanagari, R: Roman, G: Gujarati, O: Oriya, Te: Telugu, Ka: Kannada, Ks: Kashmiri, M: Malayalam, U: Urdu.

script is misclassified as Malayalam and in (b) using MLP classifier Urdu script is misclassified as Kashmiri. This misclassification was due to presence of visually similar type of graphemes among those scripts, presence of noise, skewness, and unwanted artifacts in the original script.

5.2. Statistical Performance Analysis. In the present work detailed error analysis with respect to different parameters, namely, Kappa statistics, mean absolute error, relative absoluter error, TP rate, FP rate, precision, recall, and F-measure, is computed. Table 4 provides a statistical performance analysis with respect to the said parameters showing weighted average value for all the classes.

Kappa Statistics. It measures the agreement of prediction with the true class.

Mean Absolute Error. It is measured as the average of the difference output or predicted result and target or actual result in all the test cases.

Relative Absolute Error. It is the absolute error made relative to what the error would have been if the prediction simply had been the average of the target values.

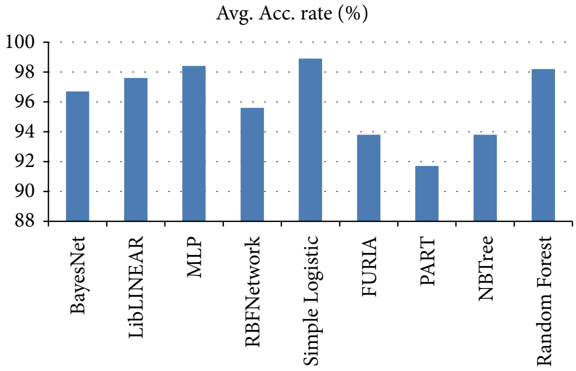


Figure 10: Comparison of average accuracy rate by different classifiers.

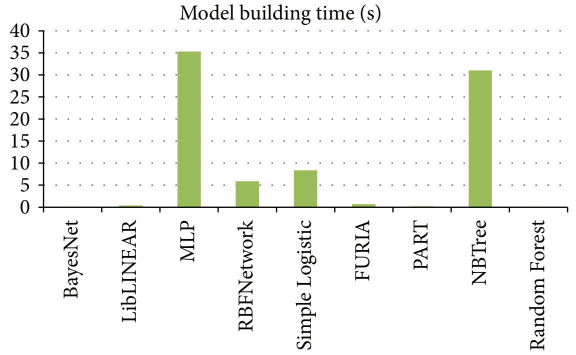


Figure 11: Comparison of model building time by different classifiers.

TP Rate. True positive rate is defined as the proportion of the test samples among all which were classified correctly to a target class to which they should belong.

FP Rate. It is opposite of TP rate. False positive rate is proportion of the test samples which belongs to a particular class but is misclassified to a different class.

Precision. It is defined as proportion of the test samples which truly have been classified to a particular class among all those which were classified to that class. So precision is TP number/(TP number + FP number).

Recall. Recall can be defined as recall = TP number/(TP number + FN number). Here FN number is the false negative number.

F-Measure. It is a combined measure for precision and recall. It is defined as F-measure $= 2 * \text{precision} * \text{recall}/(\text{precision} + \text{recall})$ (Table 4).

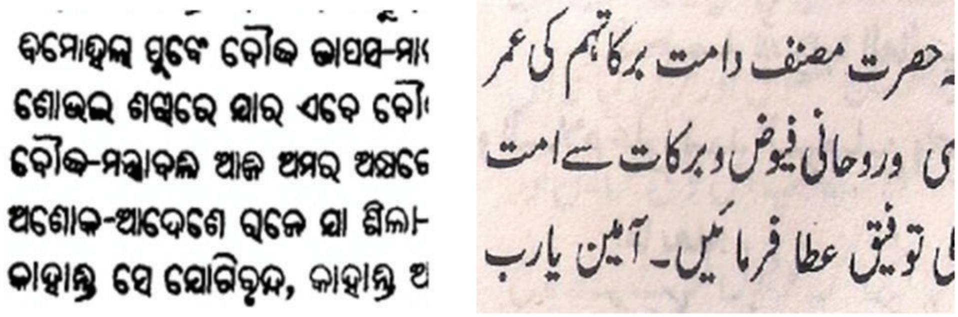
6. Conclusion

In this paper, a script identification technique is proposed for official Indic scripts. Ten popular scripts are considered for the present work. Some computationally easy and robust features are computed and different well-known classifiers are

Table 4: Statistical performance analysis and weighted average of the measuring parameters are shown here.

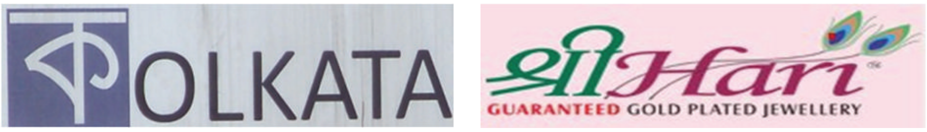
Classifier	KS	MAE	RAE	TP rate	FP rate	Precision	Recall	F-measure
BayesNet	0.9631	0.007	3.9702	0.967	0.004	0.969	0.967	0.967
LibLINEAR	0.9729	0.0048	2.7033	0.976	0.003	0.977	0.976	0.976
MLP	0.9828	0.007	3.9239	0.985	0.002	0.985	0.985	0.985
RBFNetwork	0.9509	0.0101	5.6777	0.956	0.005	0.960	0.956	0.957
Simple Logistic	0.9877	0.0059	3.3448	0.989	0.001	0.989	0.989	0.989
FURIA	0.9277	0.0198	7.2849	0.941	0.013	0.941	0.941	0.940
PART	0.9065	0.0177	9.9616	0.917	0.010	0.917	0.917	0.916
NBTree	0.9312	0.0159	8.978	0.939	0.008	0.940	0.939	0.939
Random Forest	0.9729	0.0248	13.9833	0.976	0.003	0.976	0.976	0.975

Parameters considered: KS: Kappa statistics, MAE: mean absolute error, RAE: relative absolute error and TP rate, FP rate, precision, recall, F-measure.


ବମୋହନ ପୂର୍ବେ ବୌଦ୍ଧ ତାପସ-ମା
ଶୋଭଇ ଶବ୍ଦରେ ଯାର ଏବେ ବୌ
ବୌଦ୍ଧ-ମନ୍ତ୍ରାବଳ ଆଜ ଅମର ଅକ୍ଷରେ
ଅଶୋକ-ଆଦେଶେ ଗ୍ରନ୍ଥେ ଯା ଶିଳା-
କାହାନ୍ତି ସେ ଯୋଗିବୃନ୍ଦ, କାହାନ୍ତି ଅ
حضرت مصنف دامت برکاتہم کی عمر
ی وروحانی فیوض وبرکات سے امت
ی توفیق عطا فرمائیں۔ آمین یارب

(a) (b)

Figure 12: Few of the misclassified instances are shown for different classifiers: (a) Oriya misclassified as Malayalam, (b) Urdu misclassified as Kashmiri.



(a) (b)

Figure 13: Words showing multiple scripts at character level.

used for evaluating the performance. Detail error and statistical performance analyses with different standard parameters are done. The necessity of this kind of script identification system has already been discussed in the introductory section. As mentioned before, till date no such system is available considering all official Indic scripts which encourages us to do the present work. Apart from this, there is a problem of availability of standard database on official Indic scripts. One of our future plans is to develop official Indic script database at different levels like document, block, line word, and so forth for both printed and handwritten domain. The present work is carried out on printed document but challenges are more if handwritten documents are considered. This is due to several reasons such as the following:

(i) different writing style from people of diversified cultures across the globe,

(ii) asymmetric nature of handwritten characters compared to symmetric printed characters,

(iii) presence of skew mainly at line level, sometimes at word level, and also for document level script identification,

(iv) sometimes presence of dissimilar characters within a single word from a single writer,

(v) different spacing between different words, characters, and lines in handwritten document than printed document.

Some upcoming research areas on script identification are character level script identification, video based script identification, mobile based script identification, script identification from scene images, and so forth. In character level script identification script needs to be identified from a word where multiple scripts are present at character level. Figure 13 shows such words where single word is written using multiple scripts.

Video based script identification can be helpful for automatic content based information retrieval and indexing. This can be applicable in the area like movie searching based on actor/actress name, game score based on team/player name, finding of a scene based on location name, and so forth. Challenges and scope are there for developing script identification algorithms for mobile based devices. Script identification from scene images can have several applications like tracking license plate from moving vehicles, development of driver less automating vehicles, building software for blind person for freely walking on the road, building biometric devices, extracting GPS information from Google map, and so forth. So far encouraging work is not found in these areas so attention needs to be given.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The first author of the paper is very much thankful to Hon'ble Vice Chancellor of Aliah University for extending his encouragement and support for carrying out the work besides regular academic and administrative duties.

References

[1] S. M. Obaidullah, S. K. Das, and K. Roy, "A system for handwritten script identification from Indian document," *Journal of Pattern Recognition Research*, vol. 8, no. 1, pp. 1–12, 2013.

[2] D. Ghosh, T. Dube, and A. Shivaprasad, "Script Recognition—a review," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 12, pp. 2142–2161, 2010.

[3] A. L. Spitz, "Determination of the script and language content of document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 235–245, 1997.

[4] L. Lam, J. Ding, and C. Y. Suen, "Differentiating between oriental and European scripts by statistical features," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 12, no. 1, pp. 63–79, 1998.

[5] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic script identification from document images using cluster-based templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 177–181, 1997.

[6] L. Zhou, Y. Lu, and C. L. Tan, "Bangla/English script identification based on analysis of connected component profiles," in *Proceedings of the 7th International Conference on Document Analysis Systems (DAS '06)*, vol. 3872 of *Lecture Notes in Computer Science*, pp. 243–254, 2006.

[7] J. R. Prasad, U. V. Kulkarni, and R. S. Prasad, "Template matching algorithm for Gujrati character recognition," in *Proceedings of the 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET '09)*, pp. 263–268, Nagpur, India, December 2009.

[8] B. Patil and N. V. Subbareddy, "Neural network based system for script identification in Indian documents," *Sadhana*, vol. 27, part i1, pp. 83–97, 2002.

[9] A. M. Elgammal and M. A. Ismail, "Techniques for language identification for hybrid Arabic-English document images," in *Proceedings of the IEEE 6th International Conference on Document Analysis and Recognition*, pp. 1100–1104, 2001.

[10] B. V. Dhandra, P. Nagabhushan, M. Hangarge, R. Hegadi, and V. S. Malemath, "Script identification based on morphological reconstruction in document images," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 2, pp. 950–953, Hong Kong, August 2006.

[11] B. B. Chaudhuri and U. Pal, "An OCR system to read two Indian language scripts: Bangla and Devanagari (Hindi)," in *Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR '97)*, vol. 2, pp. 1011–1015, Ulm, Germany, August 1997.

[12] C. L. Tan, P. Y. Leong, and S. He, "Language Identification in Multilingual Documents," 2003.

[13] S. Chaudhury and R. Sheth, "Trainable script identification strategies for Indian languages," in *Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR '99)*, pp. 657–660, 1999.

[14] M. C. Padma and P. A. Vijaya, "Wavelet packet based texture features for automatic script identification," *International Journal of Image Processing*, vol. 4, no. 1, 2010.

[15] G. D. Joshi, S. Garg, and J. Sivaswamy, "Script identification from Indian documents," in *Proceedings of the 7th International Workshop on Document Analysis Systems VII*, vol. 3872 of *Lecture Notes in Computer Science*, pp. 255–267, Nelson, New Zealand, 2000.

[16] D. Dhanya, A. G. Ramakrishnan, and P. B. Pati, "Script identification in printed bilingual documents," *Sadhana*, vol. 27, part 1, pp. 73–82, 2002.

[17] http://commons.wikimedia.org/wiki/File:States_of_South_Asia.png.

[18] K. Roy, U. Pal, and A. Banerjee, "A system for word-wise handwritten script identification for Indian postal automation," in *Proceedings of the 1st IEEE INDICON India Annual Conference*, pp. 266–271, December 2004.

[19] A. Kaehler and G. R. Bradski, *Learning OpenCV*, O'reilly Media, 2008.

[20] V. Singhal, N. Navin, and D. Ghosh, "Script-based classification of hand-written text documents in a multilingual environment," in *Proceedings of the 13th International Workshop on Research Issues in Data Engineering: Multi-lingual Information Management*, Research Issues in Data Engineering, pp. 47–54, 2003.

[21] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly, "Script and language identification for handwritten document images," *The International Journal on Document Analysis and Recognition*, vol. 2, no. 2-3, pp. 45–52, 1999.

[22] K. Roy, S. Kundu Das, and S. M. Obaidullah, "Script identification from handwritten document," in *Proceedings of the 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG '11)*, pp. 66–69, Karnataka, Hubli, India, December 2011.

[23] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. Kumar Basu, "A novel framework for automatic sorting of postal documents with multi-script address blocks," *Pattern Recognition*, vol. 43, no. 10, pp. 3507–3521, 2010.

[24] V. Singhal, N. Navin, and D. Ghosh, "Script-based classification of hand-written text documents in a multilingual environment," in *Proceedings of the 13th International Workshop on Research Issues in Data Engineering: Multi-Lingual Information Management (RIDE-MLIM '03)*, pp. 47–54, March 2003.

[25] S. B. Moussa, A. Zahour, A. Benabdelhafid, and A. M. Alimi, "Fractal-based system for Arabic/Latin, printed/handwritten script identification," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, IEEE, December 2008.

[26] M. Hangarge, K. C. Santosh, and R. Pardeshi, "Directional discrete cosine transform for handwritten script identification," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR '13)*, pp. 344–348, Washington, DC, USA, August 2013.

[27] R. Rani, R. Dhir, and G. S. Lehal, "Script identification of pre-segmented multi-font characters and digits," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pp. 1150–1154, August 2013.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, pp. 10–18, 2009.

[29] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[30] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

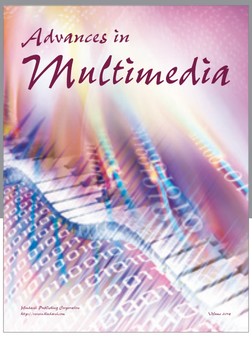
[31] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics (12), Cambridge University Press, Cambridge, UK, 2003.

[32] S. V. Chakravarthy and J. Ghosh, "Scale-based clustering using the radial basis function network," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1250–1261, 1996.

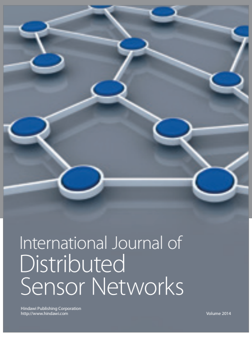
[33] A. J. Howell and H. Buxton, "RBF network methods for face detection and attentional frames," *Neural Processing Letters*, vol. 15, no. 3, pp. 197–211, 2002.

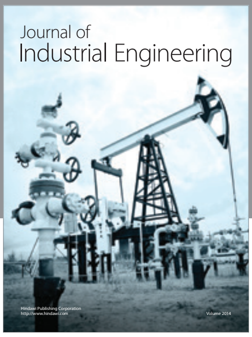
[34] J. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

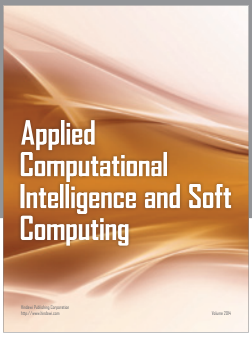
[35] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

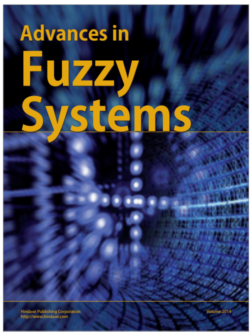

Advances in
Multimedia

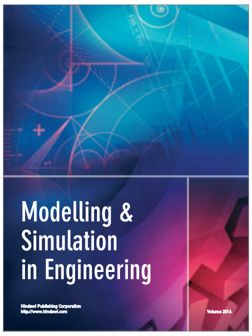
The Scientific
World Journal

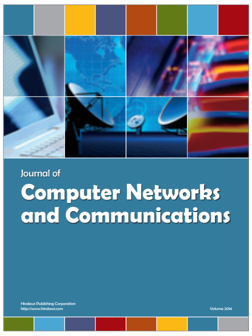

International Journal of
Distributed
Sensor Networks

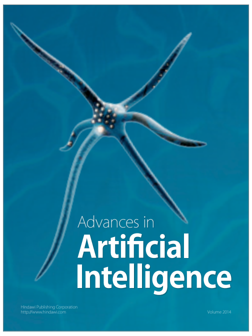

Journal of
Industrial Engineering

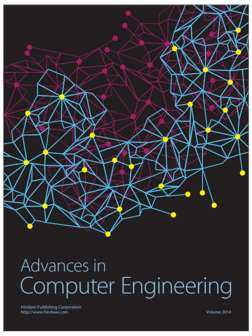

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

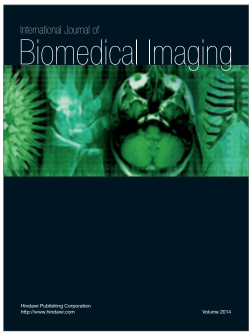

Modelling &
Simulation
in Engineering

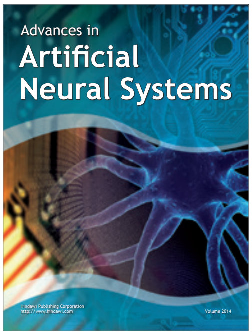

Journal of
Computer Networks
and Communications

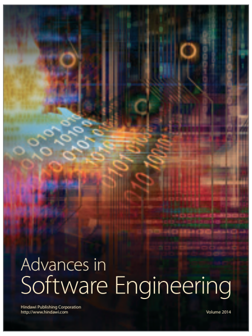

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

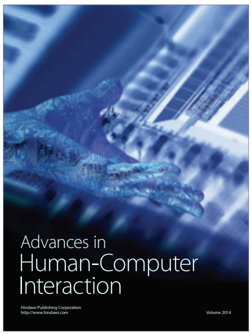
International Journal of
Computer Games
Technology

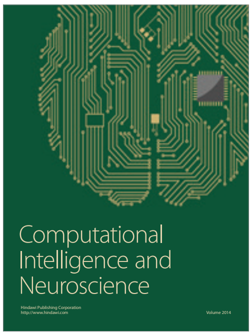

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering