Script Identification from Multilingual Indian Documents using Structural Features

Rajesh Gopakumar, N.V.SubbaReddy, Krishnamoorthi Makkithaya, U.Dinesh Acharya

Abstract— Script Identification from a given document image is an important process for many computer applications such as automatic archiving of multilingual documents, searching online archives of document images and for the selection of script specific OCR in a multilingual environment. In this paper a Zone-based Structural feature extraction algorithm towards the recognition of South-Indian scripts along with English and Hindi is proposed. The document images are segmented into lines and the line image is divided into different zones and the structural features are extracted. A total of 37 features were extracted in the first level and then reduced to an optimal number of features using wrapper and filter selection approaches. The K-nearest neighbor and the support vector machine classifiers are used for classification and recognition purpose. Very good classification accuracy is achieved on the optimal feature set.

Index Terms— Indian Scripts, Multilingual document, Script identification, Zone-based structural features

1 Introduction

It is very important to automatically identify the scripts before feeding each text line of the document to the respective OCR system in a multi-lingual document. Quite a few results have already been reported in the literature, identifying the scripts in a multi-lingual and multi-script document dealing with Roman and other Oriental scripts such as Chinese, Korean, Japanese, Arabic and Hindi. The Classifiers used include statistical analysis, linear discriminate analysis, cluster analysis and template matching based on the features like texture, upward Concavities, optical densities and characteristic shapes or symbols [5].

In this paper the aim is to find approaches for separating all south Indian scripts. The proposed system addresses the script identification pertaining to south Indian states, Kerala, Tamil Nadu, Andhra Pradesh and Karnataka.

In the context of Indian language document analysis, major literature is due to Pal and Choudhari. The automatic separation of text lines from multi-script documents by extracting the features from profiles, water reservoir concepts, contour tracing [1], [2]. Santanu Choudhury, Gaurav Harit, Shekar Madnani and R. B. Shet has proposed a method for identification of Indian languages by combining Gabor filter based technique and direction distance histogram classifier considering Hindi, English, Malayalam, Bengali, Telugu and Urdu [4]. Chanda and Pal have proposed an automatic technique for word wise identification of Devnagari, English and Urdu scripts from a single document [6]. Gopal Datt Joshi, Saurabh Garg and Jayanthi Sivaswamy have proposed script Identification from Indian Documents [7]. Word level script identification in bilingual documents through discriminating features has been developed by B V Dhandra, Mallikarjun Hangarge, Ravindra Hegadi and V.S.Malemath [8]. Neural network based system for script identification (Kannada, Hindi and English) of Indian documents is proposed by Basavaraj Patil and N. V. SubbaReddy [9]. Lijun Zhou Yue Lu and Chew Lim Tan have developed a method for Bangla and English script identification based on the analysis of connected component profiles [10]. Vijaya and Padma has developed methods for English, Hindi and Kannada script identification using discriminating features and top and bottom profile based features [11].

This paper deals with line-wise script identification for Kannada, Telugu, Tamil, Malayalam, Hindi and English scripts pertaining to documents from Southern States of India. Script identification is done based on the structural features extracted from the line image. The K-nearest neighbor (KNN) and Support Vector Machine (SVM) classifiers are used for classification and recognition purpose.

The rest of the paper is organized as follows. In Section II, brief description on Indian scripts, preprocessing, feature extraction technique, proposed methodology, and classification is explained. Section III describes the experimental results and comparative study. Conclusion and future work is given in Section IV.

2 Proposed Methodology

Structural features describe a pattern in terms of its topology and geometry by giving its global and local properties. Some of the main structural features include features like number and intersections between the character and straight lines, holes and concave arcs, number and position of end points and junctions. These features are generally hand crafted by the researchers for the kind of pattern to be classified. Indian characters contain rich structural information, which remains unchanged over font and size variation. In this paper some of these features for classification of different scripts are used.

2.1 Indian Language Scripts

India has 18 official languages which include Assamese, Bangla, English, Gujarati, Hindi, Konkani, Kannada, Kashmiri,

- *Rajesh Gopakumar is with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal-576104, Karnataka, INDIA.*
- *Dr. N. V. SubbaReddy is with Modi Institute of Science and Technology, Lakshmangarh-332311, Rajasthan, INDIA.*
- *Dr. Krishnamoorthi M. is with the Department of MCA, Manipal Institute of Technology, Manipal-576104, Karnataka, INDIA.*
- *Dr. U. Dinesh Acharya is with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal-576104, Karnataka, INDIA.*

Malayalam, Marathi, Nepali, Oriya, Punjabi, Rajasthani, Sanskrit, Tamil, Telugu and Urdu. This paper deals with four south Indian scripts namely Kannada, Telugu, Tamil and Malayalam along with English and Hindi. The image blocks of these scripts are shown in Fig. 1.

The proposed method extracts structural features of a given text document image for South India along with English and Hindi, and use them to classify the underlying script.

2.2 Preprocessing

The proposed scheme converts the gray tone image to two-tone image using a histogram based thresholding approach. For line segmentation we use a horizontal projection profile based technique. The original image and the binarized segmented line images are shown in Fig. 2. For the proposed feature extraction technique, further preprocessing was undertaken. The proposed technique sought to locate individual strokes or line segments in the character image to serve as features to the classification stage. To facilitate this, two types of preprocessing were investigated using the direction feature technique: 1) Thinning [17] and 2) Boundary extraction [18].

2.3 Feature Extraction

The feature extraction scheme derives 37 features from each line image focusing in extracting finer details of script at word level and character level. In this paper, features are extracted at line level. A feature subset selection is performed on the initial feature set for two reasons, (1) to select relevant and contributing features for classification and (2) feature set reduction.

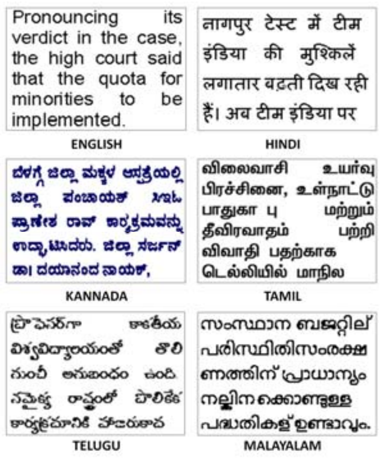
Initially features are extracted from the whole segmented line image. Further the line image is divided horizontally into 3 zones of equal size, and the feature extraction is done on individual zone. Feature extraction was applied to individual zones, for the reason that it gives more information about fine details of different scripts.

To extract different line segments in a particular zone, the entire text skeleton in that zone should be traversed. For this purpose, certain pixels in the text skeleton were defined as starters, intersections and minor starters.

Starters are those pixels with one neighbor in the character skeleton. Before character traversal starts, all the starters in the particular zone is found and is populated in a list.

The definition for intersections is somewhat more complicated. The necessary but insufficient criterion for a pixel to be an intersection is that it should have more than one neighbor. A new property called true neighbors is defined for each pixel. Based on the number of true neighbors for a particular pixel, it is classified as an intersection or not. For this, neighboring pixels are classified into two categories, direct pixels and diagonal pixels. Direct pixels are all those pixels in the neighborhood of the pixel under consideration in the horizontal and vertical directions. Diagonal pixels are the remaining pixels in the neighborhood which are in a diagonal direction to the pixel under consideration. Now for finding number of true neighbors for the pixel under consideration, it has to be classified further based on the number of neighbors it have in the text skeleton.

Pixels under consideration are classified as those with


ENGLISH
HINDI
KANNADA
TAMIL
TELUGU
MALAYALAM

Pronouncing its verdict in the case, the high court said that the quota for minorities to be implemented.

नागपुर टेस्ट में टीम इंडिया की मुश्किलें लगातार बढ़ती दिख रही हैं। अब टीम इंडिया पर

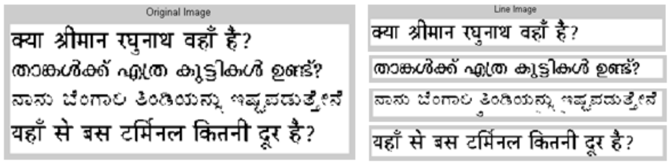
ಬೆಳಗ್ಗೆ ಜಿಲ್ಲಾ ಮಕ್ಕಳ ಆಸ್ಪತ್ರೆಯಲ್ಲಿ ಜಿಲ್ಲಾ ಪಂಚಾಯತ್ ಸಿಇಒ ಪ್ರಾಣೇಶ ರಾವ್ ಕಾರ್ಯಕ್ರಮವನ್ನು ಉದ್ಘಾಟಿಸಿದರು. ಜಿಲ್ಲಾ ಸರ್ಜನ್ ಡಾ। ದಯಾನಂದ ನಾಯಕ್,

விலைவாசி உயர்வு பிரச்சினை, உள்நாட்டு பாதுகா பு மற்றும் தீவிரவாதம் பற்றி விவாதி பதற்காக டெல்லியில் மாநில

ప్రొఫెసర్‌గా జాతీయ విశ్వవిద్యాలయంతో తొలి నుంచీ అనుబంధం ఉంది. సమైక్య రాష్ట్రంలో పొలిటిక్ కార్యక్రమానికి హాజరయ్యావ

സംസ്ഥാന ബജറ്റില് പരിസ്ഥിതിസംരക്ഷ ണത്തിന് പ്രാധാന്യം നല്കിനക്കൊണ്ടുള്ള പദ്ധതികള് ഉണ്ടാവും.

Fig.1. Indian Scripts


Original Image
Line Image

क्या श्रीमान रघुनाथ वहाँ है?
താങ്കൾക്ക് എത്ര കുട്ടികൾ ഉണ്ട്?
ನಾನು ಬೆಂಗಾಲಿ ತಿಂಡಿಯನ್ನು ಇಷ್ಟಪಡುತ್ತೇನೆ
यहाँ से बस टर्मिनल कितनी दूर है?

Fig.2. Original Image and Line Segmented images

- 3 neighbors: If any one of the direct pixels is adjacent to anyone of the diagonal pixels, then the pixel under consideration cannot be an intersection, else if none of the neighboring pixels are adjacent to each other then it is an intersection.
- 4 neighbors: If each and every direct pixel has an adjacent diagonal pixel or vice-versa, then the pixel under consideration cannot be considered as an intersection.
- 5 or more neighbors: If the pixel under consideration has five or more neighbors, then it is always considered as an intersection.

Once all the intersections are identified in the image, then they are populated in a list.

Minor starters are found along the course of traversal along the text skeleton. They are created when pixel under consideration have more than two neighbors. There are two conditions that can occur

- Intersections: When the current pixel is an intersection. The current line segment will end there and all the unvisited neighbors are populated in the minor starters list.
- Non-intersections: Situations can occur where the pixel under consideration has more than two neighbors but still it is not an intersection. In such cases, the current direction of traversal is found by using the position of the previous pixel. If any of the unvisited pixels in the neighborhood is in this direction, then it is considered as the next pixel and all other pixels are populated in the minor starter list. If none of the pixels is not in the

current direction of traversal, then the current segment is ended there and all the pixels in the neighborhood are populated in the minor starters list.

Each zone is individually subjected to the process of extracting line segments. For this purpose, first the starters and intersections in the zone are found and then populated in a list. Minor starters are found along the course of traversal. Algorithm starts by considering the starters list. Once all the starters are processed, the minor starters obtained so are processed. After that, the algorithm starts with the minor starters. All the line segments obtained during this process are stored, with the positions of pixels in each line segment. Once all the pixels in the image are visited, the algorithm stops.

Consider the skeleton in Fig.3 from which individual line segments have to be identified. Pixel in the top left corner is numbered as (1,1) and standard matrix convention is assumed so forth. Starters list in this image would be [(1,1),(1,5),(5,1),(5,5)]. Intersections would contain only the pixel (3,3). Now algorithm starts by processing the first starter i.e. (1,1). The next pixel would be (2,2) and the one after that would be (3,3). Pixel (3,3) is a intersection, so the algorithm stops the current segment, and declares all the neighbors as minor starters. So the minor starters list would contain [(4,2),(2,4),(4,4)]. The line segment formed now would contain [(1,1),(2,2),(3,3)]. Now the next starter (1,5) is considered. The next pixel (2,4) is a minor starter. So the algorithm stops the current segment here and declares all unvisited neighbors of (2,4) , if it has any, as minor starters. Also (2,4) is removed from the minor starters list, since it has been visited. Now the next starter (5,1) is considered and the next line segment so formed would be [(5,1),(4,2)]. Pixel (4,2) would be removed from the starters list. In a similar fashion, the next line segment would be [(5,5),(4,4)]. So in end, there would be 4 line segments in total.

After line segments have been extracted from the image, they have to be classified into any one of the following line types

1. Horizontal line
2. Vertical line
3. Right diagonal line
4. Left diagonal line

For this, a direction vector is extracted from each line segment which will help in determining each line type. For this, a convention is required to define the position of a neighboring pixel with respect to the center pixel of the 3x3 matrix under consideration.

1	0	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

Fig.3. Skeleton

In the direction vector matrix shown in Fig. 4, 'C' represents the center pixel. The neighboring pixels are numbered in a clockwise manner starting from pixel below the central pixel. To extract direction vector from a line segment, the algorithm travels through the entire pixels in the line segments in the order they forms the line segment. For instance, consider the line segment [(1,1),(2,2),(3,3),(4,4)]. The first pixel (1,1) is considered to be the current central pixel. With respect to this central pixel, (2,2) is in the position 8. The first entry of direction vector would be 8. Next, pixel (2,2) is considered as central pixel. Now (3,3) is also in the position 8 with respect to this pixel. So the second entry of direction vector would also be 8. Going in this fashion, the direction vector for the given line segment would be (8,8,8). Though the above set of rules identifies all line segments, a drawback is that segments in the shape of 'V' or its rotated variations will be detected as a single line segment. To prevent such errors, a new set of rules is applied once direction vector has been extracted to find new line segments inside it. The direction vector is subjected to following rules to find new line segments.

- The previous direction was 6 or 2 AND the next direction is 8 or 4 OR
- The previous direction is 8 or 4 AND the next direction is 6 or 2 OR
- The direction of a line segment has been changed in more than three types of direction OR

If a new line segment is detected, then the direction vector is broken down into two different vectors at that point. Now the following rules are defined for classifying each direction vector.

1. If maximum occurring direction type is 2 or 6, then line type is right diagonal
2. If maximum occurring direction type is 4 or 8, then line type is left diagonal
3. If maximum occurring direction type is 1 or 5, then line type is vertical
4. If maximum occurring direction type is 3 or 7, then line type is horizontal

If two line types occur same number of times, then the direction type detected first among those two is considered to be the line type of the segment.

The feature extraction scheme steps are given in the algorithm.

Algorithm: feature extraction
Begin

Step 1: Compute the following feature values for the line image (9 features)

1. Number of Horizontal lines
2. Number of Vertical lines
3. Number of Right Diagonals
4. Number of Left Diagonals
5. Normalized Length of Horizontal lines
6. Normalized Length of Vertical lines
7. Normalized Length of Right diagonals
8. Normalized Length of Left diagonals
9. Normalized Area of the line image

4	5	6
3	*C*	7
2	1	8

Fig.4. Direction vector matrix

Step 2: Divide the line image into 3 zones horizontally and compute the values in step 1 for each Zone (9 x 3 = 27 features)
Step 3: Compute Euler number for the line image (1 feature)
End

The feature extraction algorithm is applied to line images segmented from the document image. Algorithm is applied first to the whole line image and extracts the 10 features (9 features mentioned in step 1 and Euler number in step 4). Euler number is defined as the difference of number of objects and number of holes in the image. Secondly, the line image is divided into 3 zones horizontally and the features mentioned in step 1 are extracted. From 3 zones, 9 features each will give a total of 27 features. The feature vector of 37 features extracted from each line image is used for the script classification.

The number of any particular line type is normalized using the following method,

Normalized value = 1 - ((number of lines/100) x 2) and Normalized length of any particular line type is found using the following method,

Normalized length = (Total Pixels in that line type)/ (Total zone pixels).

The feature extraction procedure is shown in Fig. 5.

2.4 Script Classification

The script classification is done using k-Nearest Neighbor (kNN) classifier and Support Vector Machine (SVM) classifier.

2.4.1 k-Nearest Neighbor Classifier

For large-scale pattern matching, a long-employed approach is the kNN classifier. The training phase of the algorithm consists only of storing the feature vectors of the training samples. In the actual classification phase, the same features as before are computed for the test samples. Distances from the new vector to all the stored vectors are computed. Then, classification and recognition is achieved on the basis of similarity measurement.

2.4.2 Support Vector Machine Classifier

The support vector machine is a new classifier that is extensively used in many pattern recognition applications. Regarding the pattern classification problem, the SVM demonstrates a very good generalization performance in empirical applications. SVM is binary classifier that separates linearly any two classes by finding a hyper plane of maximum margin between the two classes. The margin means the minimal distance from the separating hyper plane to the closest data points. The SVM learning machine searches for an optimal separating hyper plane, where the margin is maximal. The outcome of the SVM is based only on the data points that are at the margin and are called support vectors.

There are two approaches to extend the SVM for multiclass classification. The first one is one against one (ONO) and the other is one against all (ONA). We have used the ONA approach where N SVM classifiers are performed to separate one of N mutually exclusive classes from all other classes. A kernel is utilized to map the input data to a higher dimensional feature space, so that the problem becomes linearly separable. The kernel plays a very important role. We have used the Radial Basis function kernel.

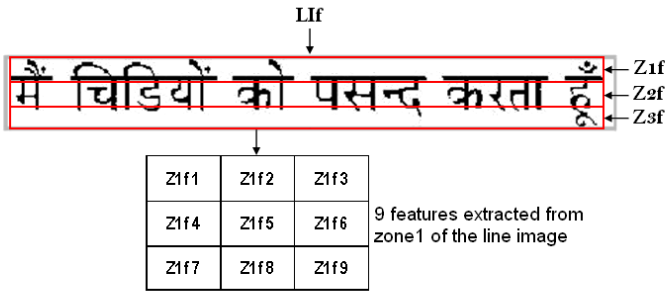

LIf
मैं चिड़ियों को पसन्द करता हूँ
Z1f
Z2f
Z3f
Z1f1
Z1f2
Z1f3
Z1f4
Z1f5
Z1f6
Z1f7
Z1f8
Z1f9
9 features extracted from zone1 of the line image

Fig.5. Feature Extraction Procedure

3 Experiments and Results

3.1 Dataset

Presently, in India, standard databases of multilingual documents for Indian scripts are unavailable. Hence, data for training and testing the classification scheme was collected from different sources. The dataset includes 2400 text lines, 400 lines of each script from different document images. The document images are prepared by downloading the web pages of e-news papers (Times of India, Navbharat Times, Udayavani, Eenadu, Daily Thanthi and Mathrubhumi respectively for English, Hindi, Kannada, Telugu, Tamil and Malayalam). Fig. 6 shows a sample page with multilingual scripts.

3.2 Feature Subset Selection

All the 37 features extracted may not be contributing to the process of script identification. We have done an analysis to find the prominent features from the whole set of 37 features.

According to Kohavi [16], to design a feature selection algorithm one has to define three following elements: *search algorithm* (technique that looks through the feature subsets), *evaluation function* (used to evaluate examined subsets of features) and, *classifier* (a learning algorithm that uses the final subset of features). These elements can be integrated in two ways. They are called *filter* and *wrapper* approaches respectively [16].

In the filter approach, features are selected as a preprocessing step before classifier is used. Features are selected (i.e. filtered) based on properties of data itself independent of the learning algorithm used in the classifier. In the wrapper approach, the search algorithm conducts a search for a good subset of features using the classifier itself as the evaluation function. Evaluation is usually done by estimating classification accuracy. The wrapper model is superior because it takes into account the biases of learning algorithm in order to get a

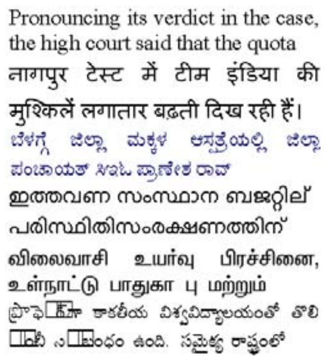

Pronouncing its verdict in the case,
the high court said that the quota
नागपुर टेस्ट में टीम इंडिया की
मुश्किलें लगातार बढ़ती दिख रही हैं।
ಬೆಳಗ್ಗೆ ಜಿಲ್ಲಾ ಮಕ್ಕಳ ಆಸ್ಪತ್ರೆಯಲ್ಲಿ ಜಿಲ್ಲಾ
ಪಂಚಾಯತ್ ಸಿಇಒ ಪ್ರಾಣೇಶ ರಾವ್
ഇത്തവണ സംസ്ഥാന ബജറ്റില്
പരിസ്ഥിതിസംരക്ഷണത്തിന്
விலைவாசி உயர்வு பிரச்சினை,
உள்நாட்டு பாதுகா பு மற்றும்
ప్రొఫె[illegible] కాకతీయ విశ్వవిద్యాలయంతో తొలి
[illegible] ఎ[illegible]ంధం ఉంది. సమైక్య రాష్ట్రంలో

Fig.6. A document image

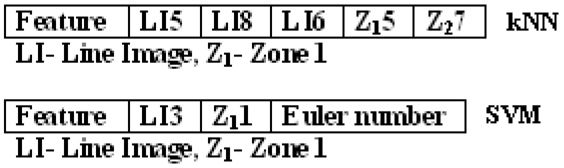
feature subset. The major limitation of the wrapper method is the computational cost resulting from using the learning algorithm to check each feature subset. This leads to choosing a simple and relatively fast learning algorithm. Other reasons for choosing classifiers are connected with an assumption that the classifier should not have an internal ability to select features while performing the learning process, i.e. it should neither remove redundant features itself nor modify them. A k-Nearest Neighbor algorithm which is a non-parametric instance-based learning algorithm with Euclidian distance measure and Support Vector Machine algorithm with Gaussian kernel for classification is used here.

The next issue concern is constructing the search algorithm. Since the exhaustive search is of exponential complexity, it is more efficient to perform the heuristic search. Commonly employed algorithms are backward elimination and forward feature selection. Former starts with all attributes and successively removes the one that its elimination improves performance. The second starts with an empty set of features and successively adds the one with the best performance. In proposed scheme we used the forward feature selection approach. In the proposed methodology, both Wrapper subset evaluation and Filtering approaches were used along with kNN and SVM classifiers.

3.3 Performance Analysis

The features for classification are selected using two subset selection methods: 1) Wrapper and 2) Filter approaches discussed earlier in section III. Fig.7 shows the features selected by wrapper approach for kNN and SVM learning. Fig. 8 shows the features selected by filter approach.

For classification purpose a 10-fold cross validation scheme is used with k-NN and SVM classifiers. As mentioned earlier, 2400 text lines, 400 of each script is used to test the system. A classification accuracy of 100% is achieved using the proposed system with optimal features selected by wrapper subset selection approach using both kNN and SVM classifiers. A classification accuracy of 100% and 98.3% is achieved for filter subset selection approach using kNN and SVM respectively. Table 1 and 2 gives the confusion matrices (in percentage) for script identification with wrapper subset selection approach using k-Nearest Neighbor and SVM classifiers, respectively. Table 3 and 4 show the confusion matrices for script identification with filter subset selection approach using k-Nearest Neighbor and SVM classifiers respectively.



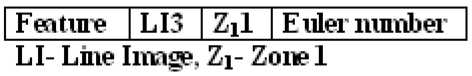
Feature	LI5	LI8	LI6	$Z_1 5$	$Z_2 7$	kNN

LI- Line Image, Z_1- Zone 1

Feature	LI3	$Z_1 1$	Euler number	SVM

LI- Line Image, Z_1- Zone 1

Fig.7. Features extracted using Wrapper approach



Feature	LI3	$Z_1 1$	Euler number

LI- Line Image, Z_1- Zone 1

Fig.8. Features extracted using Filter approach

TABLE 1
CONFUSION MATRIX FOR SCRIPT IDENTIFICATION WITH WRAPPER APPROACH USING KNN CLASSIFIER

	En	Hi	Ka	Ta	Te	Ma
En	100					
Hi		100				
Ka			100			
Ta				100		
Te					100	
Ma						100

En-English, Hi-Hindi, Ka-Kannada, Ta-Tamil, Te-Telugu, Ma-Malayalam

TABLE 2
CONFUSION MATRIX FOR SCRIPT IDENTIFICATION WITH WRAPPER APPROACH USING SVM CLASSIFIER

	En	Hi	Ka	Ta	Te	Ma
En	100					
Hi		100				
Ka			100			
Ta				100		
Te					100	
Ma						100

TABLE 3
CONFUSION MATRIX FOR SCRIPT IDENTIFICATION WITH FILTER APPROACH USING KNN CLASSIFIER

	En	Hi	Ka	Ta	Te	Ma
En	100					
Hi		100				
Ka			100			
Ta				100		
Te					100	
Ma						100

TABLE 4
CONFUSION MATRIX FOR SCRIPT IDENTIFICATION WITH FILTER APPROACH USING SVM CLASSIFIER

	En	Hi	Ka	Ta	Te	Ma
En	94.5	1.5	0.5	0.5	1	2
Hi	2.75	95	2.25	0	0	0
Ka			100			
Ta				100		
Te					100	
Ma						100

3.4 Comparative Study

The results are compared with some of the published work on line-wise script identification of Indian scripts. Table 5 provides the comparison of proposed work with other contemporary works.

TABLE 5
COMPARISON OF PROPOSED WORK WITH CONTEMPORARY WORKS

Methodology	Dataset	Overall accuracy
Rule based classifier using Top and bottom profile features[11]	500 (approx.)	96.6%
PNN classifier using Direction based pixel distribution features [9]	300 document images	98.89%
Separation scheme using script characteristics and shape based features [1]	500 (approx.)	99.3%
Proposed work (using wrapper subset selection)	2400 text lines	100%
Proposed work (using filter subset selection)	2400 text lines	99.13%

4 CONCLUSION AND FUTURE WORK

In this paper a zone based structural feature extraction algorithm is proposed for the recognition of 4 south Indian scripts along with English and Hindi. The k-nearest neighbor and support vector machine classifiers are used for subsequent classification and recognition. A good overall recognition rate of 100% is obtained for wrapper subset selection approach and 99.13% is obtained for filter approach respectively.

The proposed system can be extended to word level and character level script identification for all Indian scripts as Indian scripts are of more complex structures.

REFERENCES

[1] U. Pal, B. B. Choudhuri, "Script line separation from Indian multi-Script documents," *Proc. of fifth Intl. Conf. on Document Analysis and Recognition (IEEE computer society press)*, pp. 406-409, 1999.

[2] U. Pal, S. Sinha and B. B. Chaudhuri, "Multi-Script line identification from Indian documents," *Proc. of seventh Intl. conf. on document analysis and Recognition (ICDAR 2003)*, vol. 2, pp. 880-884, 2003.

[3] M.C. Padma and P. Nagabhushan, "Identification and separation of text words of Kannada, Hindi and English languages through discriminating features," *Proc. of second national conference on document analysis and recognition, Karnataka, India*, pp. 252-260 2003.

[4] Santanu Choudhury, Gaurav Harit, Shekar Madnani, R.B. Shet, "Identification of Scripts of Indian Languages by Combining Trainable Classifiers," *ICVGIP, Bangalore, India*, Dec.20-22, 2000.

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

[6] S. Chanda, U. Pal, "English, Devanagari and Urdu Text Identification," *Proc. Intl. Conf. on Document Analysis and Recognition*, pp. 538-545, 2005.

[7] Gopal Datt Joshi, Saurabh Garg and Jayanthi Sivaswamy, "Script Identification from Indian Documents," *LNCS 3872*, DAS, pp. 255-267, 2006.

[8] B.V. Dhandra, Mallikarjun Hangarge, Ravindra Hegadi and V.S.Malemath, "Word Level Script Identification in Bilingual Documents through Discriminating Features," *IEEE - ICSCN 2007, Chennai, India,* pp.630-635, Feb. 2007.

[9] S Basavaraj Patil and N.V. SubbaReddy, "Neural network based system for script identification in Indian documents," *Sadhana*, vol. 27, part1, pp. 83-97, February 2002.

[10] Lijun Zhou, Yue Lu and Chew Lim Tan, "Bangla/English Script Identification Based on Analysis of Connected Component Profiles," *Proc. of seventh DAS*, pp. 243-254, 2006.

[11] P. A. Vijaya, M. C. Padma, "Text line identification from a multilingual document," Proc. *of Intl. Conf. on digital image processing (ICDIP 2009) Bangkok*, pp. 302-305, March 2009.

[12] Ahmed M Elgammal Mohamed A. Ismail, "Techniques for language identification for hybrid Arabic-English document images," *Proc. of the Sixth Intl. Conf. on Document Analysis and Recognition, Seattle,* pp. 1100-1104, September 2001.

[13] Rangachar Kasturi, Lawrence O'Gorman Venu Govind Raju, "Document image analysis- a primer," *Sadhana*, vol. 27, part1, pp. 3-22, February 2002.

[14] D Dhanya, A G Ramakrishnan and Peeta basa pati, "Script Identification in printed bilingual documents," *Sadhana*, vol. 27, part1, pp. 73-82 February 2002.

[15] Jacek Jelonek, Krzysztof Krawiec and Jerzy Stefanowski, "Comparative study of feature subset selection techniques for machine learning tasks", *Proc. of VIIth Intelligent Information Systems* IIS'98, *Malbork, Poland,* pp. 68 – 77, June 1998.

[16] John G, Kohavi R, Pfleger K, "Irrelevant features and subset selection problem", *Proc. of Eleventh International Machine Learning Conference , New Brunswick, NJ, Morgan Kaufmann,* pp. 121 – 129, 1994.

[17] T Y Zhang and C Y Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns", *Communications of the ACM*, Vol. 27, pp. 236-239, 1984.

[18] Parker, J. R., *Practical Computer Vision using C*, John Wiley and Sons, New York, NY, 1994.

[19] Rajesh Gopakumar, N V SubbaReddy, Krishnamoorthi Makkithaya and U Dinesh Acharya, "Zone-based Structural Feature extraction for Script Identification from Indian Documents", *Proc. of Fifth International Conference on Industrial and Information Systems-2010, NITK Surathkal,India,* 2010.(Accepted)

Mr. Rajesh Gopakumar received his B.E. degree in Computer Science & Engineering from Gulbarga University in 1999 and M.Tech degree in Systems Analysis and Computer Applications from NITK Surathkal (Deemed) in 2005. He is currently working as Senior Lecturer in Department of Computer Science and Engineering at Manipal Institute of Technology and pursuing PhD in Computer Science and Engineering from Manipal University. He is a life member of Indian Society for Technical Education (ISTE).

Dr. N.V. SubbaReddy received his B.E. degree in Electrical and Electronics from University of Mysore in 1983. He received his M.E. degree in 1989 and PhD in Computer Science and Engineering in 1997 from Indian Institute of Science, Bangalore and University of Mysore respectively. He is a Professor in Computer Science and Engineering. He is currently the Vice-Chancellor of Modi Institute of Technology (Deemed University), Rajasthan. He has guided several PhD's in the area of pattern recognition, neural networks, fuzzy logic and genetic algorithm. Currently he is guiding PhD's in document image processing, character recognition, script identification, and machine intelligence and computer cognition.

Dr. Krishnamoorthi Makkithaya received his B.E. degree in Electronics and Communication from University of Mysore in 1988. He received his M.Tech degree in DEAC in 1994 from Mangalore University and PhD in Computer Science and Engineering in 2009 from Manipal University. He is currently working as a Professor in Department of MCA at Manipal Institute of Technology, Manipal. Currently he is guiding PhD's in Network Security, Data Mining and Pattern Recognition. He is a life member of Indian Society for Technical Education (ISTE).

Dr. U. Dinesh Acharya received his B.E. degree in Electrical and Electronics from University of Mysore in 1983. He received his M.Tech degree in Computer Science and Engineering in 1996 from Mangalore University and PhD in Computer Science & Engineering in 2008 from Manipal University. He is currently working as a Professor in Computer Science and Engineering at Manipal Institute of Technology, Manipal. Currently he is guiding PhD's in Pattern Recognition, Knowledge Based Systems and Database Systems. He is a life member of Indian Society for Technical Education (ISTE).