

Gabor Features Based Script Identification of Lines within a Bilingual/Trilingual Document

Rajneesh Rani¹, Renu Dhir¹ and Gurpreet Singh Lehal²

¹ Dr B R Ambedkar National Institute of Technology Jalandhar,
Jalandhar - 144011, India

² Department of Computer Science and Engineering, Punjabi University
Patiala - 144011, India

ranir@nitj.ac.in, dhirr@nitj.ac.in, gslehal@gmail.com

Abstract

The OCR technology for Indian documents is in emerging stage and most of these Indian OCR systems can read the documents written in only a single script. As many commercial and official documents of different states of India are tri-lingual in nature, therefore identification of script and/ or language is one of the elementary tasks for multi-script document recognition. A script recognizer simplifies the task of multi-lingual OCR by improving the accuracy and reducing the computational complexity. This script recognition may be at line, word or character level depending on interlacing of different scripts at different levels. This paper presents the effectiveness of Gabor Filter banks with kNN, SVM and PNN classifiers to identify the scripts at line level from such trilingual documents. The experiments show that Gabor features with SVM classifier achieve a recognition rate of 99.85% for trilingual documents.

Keywords: Script Recognition at Line Level, Gabor Filter, Support Vector Machine, k Nearest Neighbour, Probabilistic Neural Network

1. Introduction

Document image analysis has been an active research area from a few decades, and that facilitates the establishment of paperless offices across the world. The process of converting textual symbols present on printed and/ or handwritten paper to a machine understandable format is known as optical character recognition (OCR) which is the core of the field of document image analysis. The OCR technology for Indian documents is in emerging stage and most of these Indian OCR systems can read the documents written in only a single script. As per the trilingual formula of Indian constitution [3], every state Government has to produce an official document containing a national language (Hindi), official language (English) and state language (or regional language). For example, an official document of Punjab state contains Hindi, English and Punjabi scripts. The processing of such complex multi-script documents is a challenging problem for OCR researchers. The monolingual OCR systems will not process such multi-script documents without human involvement for delineating different script zones of multi-lingual pages before activating the script specific OCR engine. The need for such manual involvement can result in greater expense and crucially delays the overall image-to-text conversion. Thus, an automatic forwarding is required for the incoming document images to handover this to the particular OCR engine depending on the knowledge of the intrinsic scripts. In view of this, identification of script

and/ or language is one of the elementary tasks for multi-script document processing. A script recognizer, therefore, simplifies the task of OCR by enhancing the accuracy of recognition and reducing the computational complexity.

The disparity or interlacing of different scripts in a single document may be at paragraph, line, and word or character level [1, 2]. The application of a script identification system depends on the minimum size of the text from which the features proposed can be detected reliably. Script identification at paragraph/ block level for Indian languages has been reported in [4, 5]. Word-wise script identification systems for Indian scripts has been discussed in [8-15]. However, very few research works deal with script identification at line level.

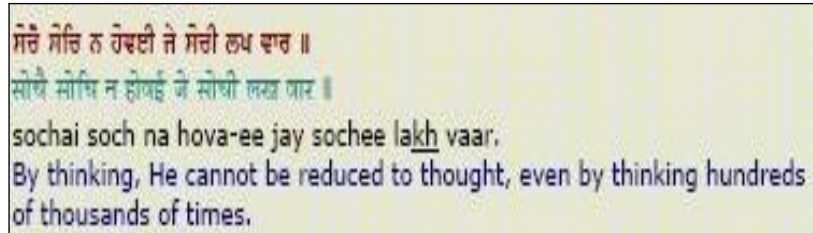


Figure 1. Sample of Punjabi text written in Gurmukhi Script interspersed with English and Hindi Lines

Pal *et al.* [6] have done work for the identification of the script from a given triplet consisting of Devanagari, Roman and a state Indian language, where each line of the text contains only a single script. Here, they have dealt with almost all the Indian scripts. Besides the headline, they have used some script dependent structural properties, such as the distribution of ascenders and descenders, the position of the vertical line in a text block and the number of horizontal runs. Different accuracy rates are available for different groups. But the overall accuracy is about 98.5%. An automatic linewise script identification scheme has been proposed by Pal *et al.* [7]. Combination of different features like shape based features, statistical features and some features obtained from a concept of water overflow from the reservoir has been employed to identify the five scripts namely Roman, Chinese, Arabic, Devanagari and Bangla. The overall accuracy of the system is 97.33%. A more generalized scheme for script identification in printed multiscript documents that can classify as many as twelve Indian scripts viz., Devanagari, Bengali, Latin, Gujrati, Kannada, Kashmiri, Malayalam, Oriya, Gurumukhi, Tamil, Telugu, and Urdu is available in [3]. According to this scheme, for the separation task, at first the scripts are grouped into a few cases according to script characteristics. Next, features based on water reservoir principle, contour tracing etc. are employed to identify them. A method to identify English Text line from Arabic text lines has been proposed by Elgammal *et al.* [16]. Features used for distinguishing Arabic from Latin are the number of peaks and the moments in the horizontal projection profile and distribution of run lengths over the location length space. Script classification using these features is done in a two-layer feed-forward network. B.Kumar *et al.* [17] have used topological, structural features with rule based classifier for line based multi-script identification. The use of Gabor features has not been discussed for Line-wise script identification to the best of our knowledge. In the present work, we have reported the use of Gabor features with kNN(k-Nearest Neighbour), SVM(Support Vector Machine) and PNN (Probabilistic Neural Network) for script identification of text lines. The experiments have been done on bi-scripts Punjabi and English text lines, Hindi and English text lines. Based on

encouraging results of these experiments, we decided to extend the experiments to tri-script containing Hindi, Punjabi and English text lines.

The remainder of paper is organized as follows: Section 2 describes Gabor filters and feature extraction using these Gabor filters. Section 3 briefly explains the kNN (k-Nearest Neighbour), SVM (Support Vector Machine) and PNN (Probabilistic Neural Network) classifiers. Section 4 provides performance analysis and experimental results. Finally, in Section 5, we summarize the conclusions of our studies.

2. Feature Extraction

Features of an image/pattern are the symbolic properties which are used to differentiate it from other image/pattern. Gabor filters, which are capable of providing multi-resolution analysis, have been used to find directional energy features [18].

2.1. 2 Dimensional Gabor Function

A Gabor function $G(x,y)$ is a linear function as given in Equation 1, defined by multiplication of harmonic function with a Gaussian function [15]

$$G(x, y) = g(x, y)s(x, y) \quad (1)$$

where $s(x,y)$ is a complex sinusoid harmonic function, known as carrier and $g(x,y)$ is a Gaussian shaped function, known as envelope. Thus the 2D Gabor filter with orientation θ and centered at frequency f can be written as in Equation 2

$$G_{x,y,\theta,f,\sigma_x,\sigma_y} = \exp^{-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)} (\cos 2\pi f x' + j \sin 2\pi f x') \quad (2)$$

where σ_x and σ_y act as the spatial spread and are the standard deviations of the Gaussian envelope along x and y direction and x' and y' are defined as:

$$x' = x \cos(\theta) + y \sin(\theta) \quad y' = y \cos(\theta) - x \sin(\theta)$$

Any combination of frequency f and orientation θ involves two filters, one corresponding to sine function and other corresponding to cosine function in Equation 2. The cosine filter also known as the real part of the filter function, is an even symmetric filter as in Equation 3 and acts as a low pass filter,

$$G_{Even}(x, y, \theta, f, \sigma_x, \sigma_y) = \exp^{-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)} \times \cos 2\pi f x' \quad (3)$$

The sine part being odd-symmetric acts like a high pass filter as in following Equation 4.

$$G_{Odd}(x, y, \theta, f, \sigma_x, \sigma_y) = \exp^{-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)} \times \sin 2\pi f x' \quad (4)$$

For a given image $I(x,y)$, the convolution of Gabor function at frequency f and angle θ with the given image yields Gabor filtered output image $O(x,y)$ as given in Equation 5

$$O(x, y) = I(x, y) * G(\theta, f) \quad (5)$$

2.2 Gabor filter Design and feature extraction

In our proposed system, we have used multi-bank Gabor filter. Five different values for Spatial frequency ($f = 0.0625, 0.125, 0.25, 0.5, 1$) and seven different values for filter orientation ($\theta = 0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 180^\circ$) have been selected to give 35 even symmetric and 35 odd symmetric filters. From the output of convolution of input image with each filter, mean and standard deviation are computed, which act as Gabor features. Thus for each text line image, we get a feature vector F of 140 values given by Equation 6:

$$F = \{\mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3, \dots, \mu_{70}, \sigma_{70}\} \quad (6)$$

Algorithm for Feature Extraction

```

Read the given input text line image I(x,y)
count=1;
Create the array Gabor_Features[ ] of size 140
  for f = 0.0625 to 1
    for  $\theta = 0^\circ$  to  $180^\circ$ 
      Gabout_Even=Convolve I(x,y) with Even Symmetric Filter
      Gabout_Odd=Convolve I(x,y) with Odd symmetric Filter
      Gabor_Features[count]= mean (Gabout_Even)
      count=count+1;
      Gabor_Features[count]= standard_deviation(Gabout_Even)
      count=count+1;
      Gabor_Features[count]= mean (Gabout_Odd)
      count=count+1;
      Gabor_Features[count]= standard_deviation(Gabout_Odd)
      count=count+1;
       $\theta = \theta + 30^\circ$ 
    end
    f=f*2
  end
end
    
```

3. Classification

The main task of classification is to use the feature vectors provided by feature extraction algorithm to assign the object/pattern to a category [15]. To observe the script recognition behavior of proposed algorithm, a comprehensive study has been made through experimental tests that are conducted on bi-script and tri-script databases using kNN, SVM and PNN classifiers.

3.1 kNN Classifier

kNN classifier uses the instance based learning by relating unknown pattern to the known according to some distance or some other similarity function. It classifies the object by majority vote of its neighbor. Because it considers only neighbor object to a particular level, it uses local approximation of distance function. In our implementation we have used Euclidean distance as the distance parameter, while using other distance functions we obtained degraded results.

Let x_i be an input training sample with m features $\{x_{i1}, x_{i2}, x_{i3}, \dots, x_{im}\}$, n be the total number of input samples $\{i = 1, 2, \dots, n\}$ and m the total number of features $\{j = 1, 2, \dots, m\}$. The Euclidean distance between sample x_i and test sample y is defined in Equation 7:

$$d(x_i, y) = \sum_{j=1}^m \sqrt{(x_{ij} - y_j)^2} \quad (7)$$

It means lazy or instance learning is used in k-NN while in other classifiers as SVM eager learning is used. k specifies the number of nearest neighbors to be considered and the class of majority of these neighbors is determined as the class of unknown pattern. More robust models can be achieved by locating k , where $k > 1$, neighbors and letting the majority vote decide the outcome of the class labeling. A higher value of k results in a smoother, less locally sensitive, function. Generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct.

3.2 SVM Classifier

Binary (two-class) classification using support vector machines (SVMs) is a very well developed technique to find the optimal hyperplane to maximize the distance or margin between two classes .

Given a training set of instance-label pairs $(x_i, y_i), i = 1, 2, \dots, l$ where $x_i \in R^n$, i.e. having n features for a particular training sample and $y_i \in \pm 1$, i.e. class label either 1 or -1 for corresponding training instance x_i . If the training data are linearly separable, we can select two hyperplanes in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". The distance between these two hyperplanes is $\frac{2}{\|w\|}$, so $\|w\|$ should be minimum [19].

In testing phase, for a given input pattern x , the decision function of an SVM binary classifier is defined in Equation 8

$$f(x) = \text{sign}(\sum_{i=1}^n y_i \alpha_i K(x, x_i) + b) \quad (8)$$

where:

$$\left(\text{sign}(u) = \begin{cases} 1 & \text{for } u > 0 \\ -1 & \text{for } u < 0 \end{cases} \right)$$

b is the bias, α_i is the langrage multiplier and $K(x, x_i)$ is the kernel function [20,21].

There are several number of kernels used in support vector machines. Some of the popularly used kernel functions are shown in Table 1:

Table 1. Different Kernel Functions of SVM Classifier

Linear Kernel	$K(x, x_i) = x^T x_i$
Polynomial Kernel	$K(x, x_i) = (x^T x_i + 1)^d$
Gaussian (RBF) Kernel	$K(x, x_i) = \exp(-\gamma * \ x - x_i\ ^2)$

3.3 PNN Classifier

The PNN is a direct implementation of the work on Bayes classifiers. The PNN classifier estimates the probability density function for each class based on the training samples using the Parzen estimators [22]. In other words, PNN is interpreted as a function, which estimates the probability density of the underlying examples distribution. The PNN consists of an input layer followed by two computational layers and one output layer. PNN receives the n -dimensional feature vectors $x(x_1, x_2, \dots, x_n)$ as input and in first layer these inputs are applied to the input neurons $x_i(1 \leq i \leq n)$ and passed to the neurons in the first hidden layer (i.e. second layer). This second layer consists of the probability density functions (*pdf*); mostly Gaussian functions are used as pdf which are formed using the given set of data points. m_k Gaussians are computed for each class where $1 \leq k \leq c$. The third layer performs an average operation of the outputs from the second layer for each class. The fourth layer performs a vote, selecting the largest value. The associated class label is then determined. If the probability density function (*pdf*) of each population is known, then an unknown x , belongs to class i if: $f_i(x) > f_j(x)$, for all $j \neq i$, here $f_k(x)$ is the pdf of unknown x for class k . *pdf* for a population of n samples is computed by computing the average of *pdfs* of all samples as given in Equation 9:

$$p(x) = \frac{1}{n\sigma} \sum_{k=1}^n W\left(\frac{x-x_k}{\sigma}\right) \quad (9)$$

Where, x_k is k^{th} sample, x is unknown input, W is weighting function also called window function or kernel and σ is smoothing parameter in the form of standard deviation. The value of smoothing parameter σ has been varied in different experiments to get appropriate value.

4. Experimental Results and Discussions

Most of the multi-lingual documents in India are bi-script in nature, so bi-script Gurumukhi/English and Hindi/English cases are handled first. Based on the encouraging results that we got in these experiments, we decided to extend the experiments to tri-script case that is Gurmukhi/Hindi/English as well.

4.1 Data Set

Due to lack of availability of standard databases, about hundred pages are scanned from English, Punjabi and Hindi documents and are segmented into lines by using horizontal projection profile method. Three datasets of different script lines have been prepared using 2512 lines of Gurumukhi Script, 2548 English and 2514 of Hindi as shown in Table 2.

Table 2. Datasets for Experiments

Dataset	Scripts	Size of Dataset
Dataset A	Gurmukhi and English	2512+2548=5060
Dataset B	Hindi and English	2514+2548=5062
Dataset C	Gurmukhi, Hindi and English	2512+2514+2548=7574

4.2 Results and Discussions

We have used 10-fold cross validation scheme for recognition result computation. First we created randomly generated 10-fold cross validation indices of the length of each of the

dataset. These indices contain equal proportions of the integers 1 through 10. These integers are used to define a partition of whole datasets into 10 disjoint subsets. We have used one subset for testing and remaining nine subsets for training for each dataset. We have done this 10 times for each classifier SVM, kNN and PNN with different parameters, each time changing the testing dataset to different subset and considering remaining subsets for training.

The recognition rates for all the test subsets are averaged to calculate average recognition accuracy. The results of average accuracy μ and standard deviation σ for different subsets for bi-script and tri-script recognition using each classifier are shown in Table 3, Table 4 and Table 5. The maximum recognition accuracy and corresponding standard deviation is represented with boldface in these tables.

4.2.1 Bi-Script Recognition:

The results of SVM experiments are represented in table 3 using three kernel functions as described in table 1. The close inspection of table 3 reveals that with all kernel functions the average recognition accuracy is above 99%. For dataset A maximum μ is 99.92% with linear kernel function and dataset B maximum μ is 99.96% with polynomial kernel function.

Table 3. Results for Script Identification using different kernel functions of SVM Classifier

Datasets	Scripts		<i>Linear</i>	<i>Polynomial</i>	<i>RBF</i>
Dataset A	Gurmukhi and English	μ	99.92	99.88	99.27
		σ	0.167	0.167	0.406
Dataset B	Hindi and English	μ	99.94	99.96	99.01
		σ	0.095	0.083	0.510
Dataset C	Gurmukhi, Hindi and English	μ	99.85	99.79	98.02
		σ	0.131	0.142	0.577

In kNN experiments, different values of $k=1, 3, 5$ and 7 are taken in different experiments using Euclidean distance. It can be observed from Table 4 that for all values of k results are above 99%. However the maximum recognition accuracies with least standard deviation for both datasets A and datasets B are 99.57% and 99.66% respectively with $k=1$.

Table 4. Results for Script Identification using different values of k with kNN Classifier

Datasets	Scripts		$k=1$	$k=3$	$k=5$	$k=7$
Dataset A	Gurmukhi and English	μ	99.57	99.55	99.57	99.47
		σ	0.292	0.407	0.382	0.373
Dataset B	Hindi and English	μ	99.66	99.57	99.37	99.33
		σ	0.295	0.382	0.404	0.449
Dataset C	Gurmukhi, Hindi and English	μ	98.59	98.19	98.18	97.95
		σ	0.324	0.413	0.469	0.466

The results of performing the experiments with PNN classifier are depicted in Table 5 with different values of σ . While observing the table carefully, it has been examined that results for both datasets A and B are above 99% only when σ lies between 0.15 and 0.25. Out of this range, accuracy decreases and also the standard deviation increases.

Table 5. Results for Script Identification using different values of σ with PNN Classifier

Datasets	Scripts		$\sigma=0.1$	$\sigma=0.15$	$\sigma=0.2$	$\sigma=0.25$	$\sigma=0.5$
Dataset A	Gurmukhi and English	μ	94.17	99.59	99.53	99.19	97.57
		σ	0.890	0.301	0.339	0.488	0.639
Dataset B	Hindi and English	μ	97.98	99.60	99.21	99.85	97.14
		σ	0.548	0.406	0.475	0.500	0.978
Dataset C	Gurmukhi, Hindi and English	μ	92.43	94.55	90.39	87.45	79.13
		σ	0.945	0.961	1.370	1.399	1.919

4.2.2 Tri-script recognition

For tri-script recognition, maximum average accuracy obtained with SVM classifier is 99.85% with linear kernel function, kNN classifier is 98.59% with $k = 1$ and PNN classifier is 94.55% for $\sigma = 0.15$ as shown in Tables 3, 4 and 5. Further the average accuracy and standard deviation for tri-script recognition using all classifiers with different parameters are represented in Figure 2 and Figure 3.

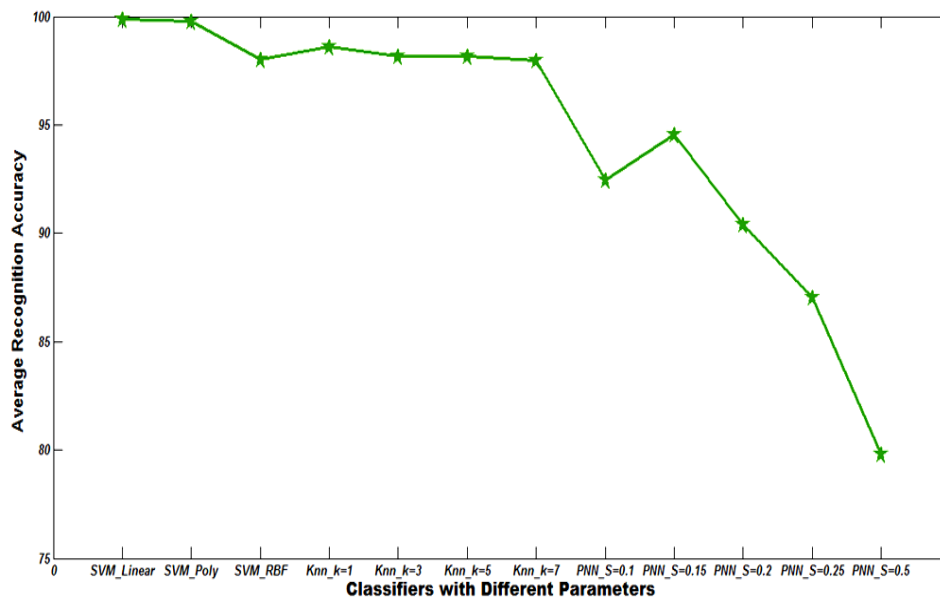


Figure 2. Plot of Averaged Recognition Accuracy obtained using different classifiers for tri-script recognition of Lines

The results reported in these figures clearly show that SVM with linear and polynomial kernel function, giving 99.85% and 99.79% accuracy, are leading as compared to other two classifiers as accuracy obtained here is higher with least standard deviation. The performance of kNN classifier for all values of k lies between 97.95% and 98.59% which is better than PNN. Using PNN, the worst performance of 79.13% is for $\sigma = 0.5$ and best performance is 94.55% for $\sigma = 0.15$.

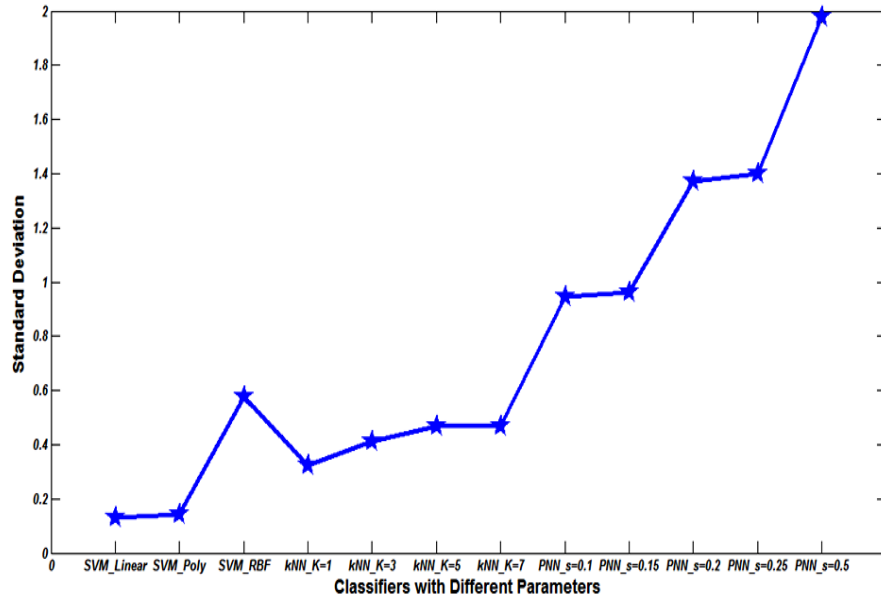


Figure 3. Plot of standard deviation obtained using different classifiers for tri-script recognition of Lines

Table 6. Confusion matrix of tri-script recognition using linear kernel function of SVM Classifier

Scripts	Gurmukhi	English	Hindi
Gurmukhi	2507	0	5
English	0	2548	0
Hindi	5	1	2508

Table 7. Confusion matrix of tri-script recognition using polynomial kernel function of SVM Classifier

Scripts	Gurmukhi	English	Hindi
Gurmukhi	2506	2	4
English	2	2546	0
Hindi	7	1	2506

The error distributions (Confusion matrices’) among different scripts using linear and polynomial kernel functions of SVM classifier are shown in Table 6 and Table 7. It can be noticed that considerable confusion has occurred in case of Gurumukhi and Hindi. The reason is straightforward, that is Hindi and Gurmukhi languages have many similar features.

5. Conclusion

A novel feature extraction algorithm using Gabor filter banks has been presented in this paper for script identification of individual lines in a document image. The effectiveness of the proposed algorithm has been demonstrated by performing the comparative analysis between three classifiers kNN, SVM and PNN. The experimental results have clearly shown the potential of the proposed approach with linear and polynomial kernel functions of SVM

classifier for line wise script identification. In future more rigorous investigations are needed to study the potential of proposed approach by considering other Indian and non-Indian scripts.

References

- [1] D. Ghosh, T. Dube and A. P. Shivaprasad, "Script Recognition A Review" IEEE, Transactions on Patter Analysis and Machine Intelligence, vol. 32, no. 12, (2010) December, pp. 2142-2161.
- [2] R. Rani and R. Dhir, "A Survey: Recognition of Scripts in Bi-Lingual/Multi-Lingual Indian Documents", National PIMT Journal of Research vol. 2, no. 1, (2009) March- August, pp. 55-60.
- [3] U. Pal, S. Sinha and B. B. Chaudhri, "Multi-Script Line Identification from Indian Documents", Proceedings of International Conference on Document Analysis and Recognition, (2003) August, pp. 880-884.
- [4] G. D. Joshi, S. Garg and J. Sivaswamy, "Script Identification from Indian Documents", Proceedings of IAPR International Workshop Document analysis Systems, (2006) February, pp. 255-267.
- [5] S. Chaudhury and R. Sheth, "Trainable Script Identification Strategies for Indian Languages", Proceedings of International Conference on Document Analysis and Recognition, (1999) September, pp. 657-660.
- [6] U. Pal and B. B. Chaudhri, "Script Line Separation from Indian Multi-Script Documents", Proceedings of International Conference on Document Analysis and Recognition, (1999) September, pp. 406-409.
- [7] U. Pal and B. B. Chaudhuri, "Identification of Different Script Lines from Multi-Script Documents", Image and Vision Computing, vol. 20, no. 13, (2002) December, pp. 945-954.
- [8] S. Sinha, U. Pal and B. B. Chaudhuri, "Word-Wise Script Identification from Indian Documents", Proceedings of IAPR International Workshop on Document Analysis Systems, (2004) September, pp. 310-321.
- [9] S. Chanda, S. Sinha and U. Pal, "Word-Wise English Devanagari and Oriya Script Identification", Speech and Language Systems for Human Communication, (2004), pp. 244-248.
- [10] S. Chanda, R. K. Roy and U. Pal, "English, Devanagari and Urdu Text Identification", Proceedings of International Conference on Cognition and Recognition, (2005) December, pp. 538-545.
- [11] M. C. Padma and P. Nagabhushan, "Identification and Separation of Text Words of Kannada, Hindi and English Languages through Discriminating Features", Proceedings of National Conference on Document Analysis and Recognition, (2003) July, pp. 252-260.
- [12] P. B. Pati and A. G. Ramakrishnan, "Word level multi-script Identification", Pattern Recognition Letters, (2008), pp. 1218-1229.
- [13] B. V. Dhandra and M. Hangarge, "On Separation of English Numerals from Multilingual Document Images", Journal of multimedia , vol. 2, no. 6, (2007) November, pp. 26-33.
- [14] R. Rani, R. Dhir and G. S. Lehal, "Comparative Analysis of Gabor and Discriminating Feature Extraction Techniques for Script Identification", Proceedings of ICISIL, Communications in Computer and Information Science 139, Springer, (2011) March.
- [15] R. Rani, R. Dhir and G. S. Lehal, "Modified Gabor Feature Extraction Method for Word Level Script Identification- Experimentation with Gurumukhi and English Scripts", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 6, no. 5, (2013), pp. 25-38.
- [16] A. Elgammal and M. A. Ismail, "Techniques for Language Identification for Hybrid Arabic-English Document Images", Proceedings of International Conference on Document Analysis and Recognition, (2001) September, pp. 1100-1104.
- [17] B. Kumar, A. Bera and T. Patnaik, "Line Based Robust Script Identification for Indian Languages", International Journal of Information and Electronics Engineering, vol. 2, no. 2 (2012), pp. 189-192.
- [18] D. Dhanya, A. G. Ramakrishnan and P. B. Pati, "Script identification in printed bilingual documents", Sadhana, vol. 27, no. 1, (2002) February, pp. 73-82.
- [19] Support Vector Machines (SVM), StatSoft website, <http://www.statsoft.com/textbook/support-vector-machines/>.
- [20] Support Vector Machine, Wikipedia website, <http://en.wikipedia.org/wiki/Support-vector-machine>.
- [21] C. -W. Hsu, C. -C. Chang and C. -J. Lin, "A Practical Guide to Support Vector Classification", <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>, (2010) April.
- [22] A. H. Khandoker, C. K. Karmakar and M. Palaniswami, "Automated recognition of patients with obstructive sleep apnea using wavelet-based features of electrocardiogram recordings", Computers in Biology and Medicine Elsevier, vol. 39, (2009), pp. 88-96.

Authors



Rajneesh Rani

She has received the B.Tech and M.Tech degrees, both in Computer Science and Engineering, from Punjab Technical University, Jalandhar, India in 2001 and Punjabi University Patiala, India in 2003 respectively. From 2003 to 2005, she was a lecturer in Guru Nanak Dev Engineering College, Ludhiana. She is currently working as an assistant professor in NIT Jalandhar since 2007. Her teaching and research include Image Processing, Pattern Recognition, Machine Learning, Computer Programming and Document Analysis and Recognition.



Dr. Renu Dhir

Dr. Renu Dhir received the B.Tech degree in Electrical Engineering from Guru Nanak Dev Engineering College, Ludhiana, India in 1983, M.Tech in Computer Science and Engineering from TIET Patiala, India in 1997 and Ph.D in Computer Science and Engineering from Punjabi University Patiala, India in 2007. She is currently associate professor in the Department of Computer Science and Engineering, NIT Jalandhar. Her teaching and research include Image Processing, Pattern Recognition, Machine Learning and Network Security.



Dr. Gurpreet Singh Lehal

Gurpreet Singh Lehal received B.Sc and M.Sc (Maths Hons.) from Punjab University, Chandigarh, India. He has done M.E (Computer Science) from TIET Patiala, India and Ph.D from Punjabi University Patiala. He is currently working as a Professor in the Department of Computer Science and Engineering, Punjabi University, Patiala. He is also acting as a Director of Advanced Center for Technical Development of Punjabi Language Literature and Culture. For many years, he has done research in image processing, pattern recognition and Natural Language Processing and has developed many technologies and language software. He is a member of IEEE.

