ISSN:0975-9646
Kartar Singh Siddharth et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1036-1041

Handwritten Gurumukhi Character Recognition Using Zoning Density and Background Directional Distribution Features

Kartar Singh Siddharth[#1], Renu Dhir[#2], Rajneesh Rani[#3]

[#]*Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of Technology, Jalandhar- 144011, Punjab (India)*

***Abstract*— This paper presents an approach to recognize isolated handwritten Gurumukhi characters. Extracted features used are 16 zonal densities and the 8 background directional distribution features for each of 16 zones. All sample images of Gurumukhi characters used are normalized to 32*32 pixel sizes. Zoning density is computed by dividing number of foreground pixels in each zone by number of total pixels in the zone. Background directional distribution values are calculated for each foreground pixel by directional distribution of its neighbouring background pixels. The value for each directional distribution is computed by summing up the values specified in a mask for corresponding neighbour background pixels. A specific mask is used in particular direction. These feature values for each direction are summed up for all pixels in each zone. Thus adding 16 zoning density features and 128 background directional distribution features (8 features in each of 16 zones) we used feature vector containing 144 features in our experiment. SVM classifier with RBF kernel is used for classification. We have taken 200 samples of each of 35 basic Gurumukhi characters in our experiment, which are collected from 20 different writers each contributing to write 10 samples of each of 35 characters. Thus we have used total 7000 character samples. By training the classifier with whole dataset we obtained 95.04% 5-fold cross validation accuracy. By using dataset taken by 16 writers (size 5600) to train the system and testing the 1400 samples by another 4 unknown writers to classifier, we obtained 89.14% accuracy. By splitting the handwritten samples of each writer to train and test the system in the ratio of 4:1, we received 99.93% test accuracy for known writers. Thus we got 94.53% average test accuracy for known and unknown writers and 95.04% 5-fold cross validation accuracy.**

***Keywords*— Isolated Handwritten Gurumukhi Character Recognition, Zoning Density, Background Directional Distribution, SVM Classifier.**

I. INTRODUCTION

Optical Character Recognition (OCR) is a mechanism in which document in scanned or image format is converted to editable text format. The inputted document image consists of printed, type-written or hand-written text of any language script. Character recognition for printed text is simpler than handwritten text because handwritten text consists of more variations of different styles of different people even different styles of same writer at different times. The text to recognize consists of characters, words, sentences, paragraphs, thus in increasing cumulative order, forming complete image document.

The input document image can be taken as offline, i.e. processing it after completely writing down the document and then scanning it. The input for character recognition is also taken as online in which characters are recognized as soon as these are written down or within a fraction of time. In this method complete document is not first scanned and then recognized cumulatively. Online mode of recognition is generally associated with some web or other application.

The pattern recognition approach is classified traditionally as template based and feature based. In template based approach an unknown pattern is superimposed on the ideal pattern and pattern is classified based on degree of correlation between those. In feature based approach features of pattern are extracted and based on these features these patterns are classified using appropriate classifier or combination of classifiers. Combining different types of features or classifiers increase the complexity of the system but increase the accuracy of the result.

The recognition is practiced at many levels. The simplest way is to recognize isolated characters of a script. While recognizing a script at word level, sentence level or any higher level requires segmentation mechanism to isolate classifiable objects. After recognition recognized characters are grouped accordingly in words, sentences or paragraphs as they were presented in input document.

The basic mechanism of offline character recognition consists of following phases: *Image Pre-processing, Segmentation, Feature Extraction, Classification and Post Processing*. (Fig.1)

In image pre-processing we apply processes like noise removal, skeletonization, edge detection, filtration, binarization, normalization, removing or reducing other irregularities and making the document image easier to process in order to increase the overall efficiency of the recognition system.

By segmentation phase we segment the single characters or units which we are intended to recognise at module level. We

can also segment a single character in many zones to recognize these zones independently and then to merge these recognized results to form particular character. This particular character may be inclusive of vowel modifiers, half characters and many other variations to a character.

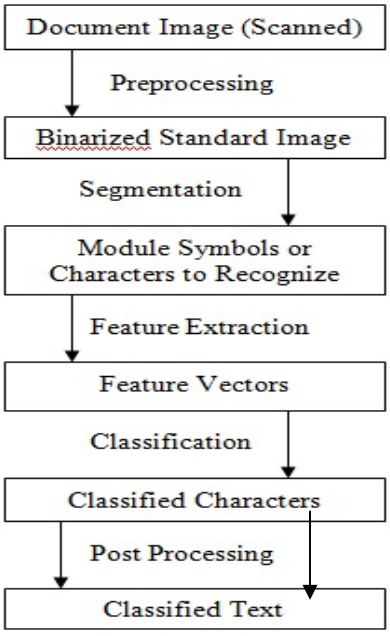

Fig.1. Basic Steps of Character Recognition

Feature extraction phase is prominent phase that highly affects the recognition efficiency. So feature extraction should be done carefully and suitably. A detailed surveys on feature extraction methods and the facts that decide applicability and suitability of a particular feature extraction method is presented in [1]. It suggests different types of feature extraction methods suitable for different types of input image as gray scale, binary (solid or contour) or vector. Later, a survey of methods and strategies for feature extraction in handwritten script identification was presented in [2]. This survey paper suggests decomposing the feature extraction phase in two sub-phases- feature construction and feature selection. In feature construction raw features and even irrelevant features are considered not to lose any information. Adding all these features increases the dimensionality of patterns. In feature selection step only relevant features are identified and selected to create feature vectors.

In our implementation we have used 8-directional background distribution and zoning density features forming a sum of 144 features. This approach is described later in this paper.

Each pattern having feature vector is classified in predefined classes using appropriate classifiers. Today mostly used and practiced classifiers are- Support Vector Machine (SVM), K- Nearest Neighbour (K-NN), Artificial Neural Network (ANN).

Cheriet and Kharma et al. [3] have presented a prominent and useful guide for tools for image pre-processing, Feature extraction, selection and creation, pattern classification methods including statistical methods, Artificial neural networks (ANN), Support Vector Machines (SVM), Structural pattern recognition and combining these multiple classifiers.

In our recognition approach we have used SVM classifier with its Radial Basis Function (RBF) kernel.

In post processing step we bind up our work to create complete digitized document after recognition process.

II. Gurumukhi Script and Characteristics

The Gurumukhi alphabet was devised during the 16th century by Guru Nanak, the first Sikh guru, and popularized by Guru Angad. The word *Gurumukhi* means the sayings came out from the mouth of (Sikh) Gurus [4]. Gurumukhi script, also spelled as Gurmukhi, is used to write primarily Punjabi and secondarily Sindhi language. Guru Angad not only modified, rearranged the letters but also documented them properly into a script [5].

Gurumukhi script consists of 35 basic character, among which there are 3 vowel holder. In addition to these 35 characters, there are 12 vowel modifiers, 6 additional modified consonants, forming 41 consonants with 35 basic characters [6] [7]. (Fig.2)

Fig.2. Gurumukhi Alphabet (Courtesy- [4])

Additionally, Gurumukhi alphabet contains some symbols, which are used as modifier of characters. These symbols are *tippi* (ੰ), *bindi* (ਂ), *adhak* (ੱ), *halant* (੍) and *visarg* (ਃ).

Characteristics of Gurumukhi script

Gurumukhi script contains following listed characteristics [4] [8]:

- Gurumukhi alphabet is syllabic in which all consonants have an inherent vowel. Gurumukhi is written from left to right like most other Indian and Roman scripts.
- Unlike Roman characters, Gurumukhi is written below the line and it has no concept of lower and upper case.
- A text of Gurumukhi script can be partitioned into three horizontal zones namely upper zone, middle zone and lower zone. (Fig.3)
- Normally consonants appear in middle zone and diacritics (vowel modifiers, half characters and other modifiers and symbols) appear in upper zone, lower zone, before or after the consonant to change syllable sound of that particular consonant.
- The Gurumukhi script, unlike the Greek and Roman alphabets, is arranged in a logical fashion: vowels first, then consonants (Gutturals, Palatals, Cerebrals, Dentals, and Labials) and semi-vowels.
- Upper zone and middle zone is separated by a line called header line or sirorekha as present in other Indian script like Devnagari.

Fig.3. Three horizontal zones of Gurumukhi text

III. Earlier Related Work

For Indian languages most of research work is performed on firstly on Devnagari script and secondly on Bangla script. U. Pal and B.B. Chaudhury presented a survey on Indian Script Character Recognition [9]. This paper introduces the properties of Indian scripts and work and methodologies approached to different Indian script. They have presented the study of the work for character recognition on many Indian language scripts including Devnagari, Bangla, Tamil, Oriya, Gurumukhi, Gujarati and Kannada.

U. Pal, Wakabayashi and Kimura also presented comparative study of Devnagari handwritten character recognition using different features and classifiers [10]. They used four sets of features based on curvature and gradient information obtained from binary as well as gray scale images and compared results using 12 different classifiers as concluded the best results 94.94% and 95.19% for features extracted from binary and gray image respectively obtained with Mirror Image Learning (MIL) classifier. They also concluded curvature features to use for better results than gradient features for most of classifiers.

A later review of research on Devnagari character recognition is also presented by Vikas Dungre et al. [11]. They have reviewed the techniques available for character recognition. They have introduced image pre-processing techniques for thresholding, skew detection and correction, size normalization and thinning which are used in character recognition. They have also reviewed the feature extraction using Global transformation and series expansion like Fourier transform, Gabor transform, wavelets, moments ; statistical features like zoning, projections, crossings and distances ; and some geometrical and topological features commonly practiced. They also reviewed the classification using template matching, statistical techniques, neural network, SVM and combination of classifiers for better accuracy is practiced for recognition.

Sandhya Arora et al. [12] used intersection, shadow features, chain code histogram and straight line fitting features and weighted majority voting technique for combining the classification decision obtained from different Multi Layer Perceptron (MLP) based classifier. They obtained 92.80% accuracy results for handwritten Devnagari recognition.

For online Devnagari handwritten character recognition Fuzzy directional features are proposed in [13] and result is obtained with upto 96.89% accuracy.

Prachi Mukherji and Priti Rege [14] used shape features and fuzzy logic to recognize offline Devnagari character recognition. They segmented the thinned character into strokes using structural features like endpoint, cross-point, junction points, and thinning. They classified the segmented shapes or strokes as left curve, right curve, horizontal stroke, vertical stroke, slanted lines etc. They used tree and fuzzy classifiers and obtained average 86.4% accuracy.

Handwritten character recognition using directional features and neural network as classifier is proposed in [15]. In this approach gradient features using Sobel's mask are extracted and then these features are converted into 12 directional features. Neural network with back propagation is used for classification and 97% accurate result is reported.

In particular to Gurumukhi script, Earliest and major contributors founded are C. Singh and G. S. Lehal. G. S. Lehal and C. Singh proposed Gurumukhi script recognition system [16]. Later they developed a complete machine printed Gurumukhi OCR system [17]. Some of their other research works related to Gurmukhi Script recognition are [18], [19] in which they proposed feature extraction, classification and post processing approaches for Gurumukhi scripts. Sharma and Lehal proposed an iterative algorithm to segment isolated handwritten words in Gurumukhi script [20]. G.S. Lehal used four classifiers in serial and parallel mode and combined the results of classifiers operated in parallel mode. Combining multiple classifiers their individual weaknesses can be compensated and their strength are preserved. A comparative study of Gurumukhi and Devnagari Script is presented in [21]. In this paper closeness of Devnagari and Gurumukhi script, consonants, conjunct consonants, vowels, numerals and punctuation is discussed.

D. Sharma and G.S. Lehal et al. recognized Gurumukhi and Roman numeral using 12 structural features comprising of presence of loop, no. of loops, position of loop, entry point of

loop(5 features), curve in right upper portion, , presence of right straight line and aspect ratio and statistical features like zoning and directional distance distribution of foreground pixels [22]. They compared the results of different features and observed 92.6% highest accuracy using 256 Directional Distance Distribution (DDD) features.

Anuj Sharma et al. [23] proposed an approach for online handwritten Gurumukhi character recognition using Elastic matching. In First stage they recognised the strokes and in second stage they evaluated the character based on the strokes. They achieved the 90.08% accuracy.

For isolated handwritten Gurumukhi character recognition two work are found- first, Puneet Jhajj et al. [24] and second, Ubeeka Jain et al. [25].

Puneet Jhajj et al. used a 48*48 pixels normalized image and created 64 (8*8) zones and used zoning densities of these zones as features. They used SVM and K-NN classifiers and compared the results and observed 72.83% highest accuracy with SVM kernel with RBF kernel. Ubeeka Jain et al. created horizontal and vertical profiles, stored height and width of each character and used neocognitron artificial neural network for feature extraction and classification. They obtained accuracy 92.78% at average.

IV. Proposed Methodology

In our proposed methodology of isolated handwritten Gurumukhi character recognition we have considered 35 basic characters of Gurumukhi alphabet for our experiment. These characters are assumed to bear header line on top. 20 writers of different profiles, age and genders have written these samples in isolated manner on A-4 size white papers. 10 samples of each character by each writer are taken, thus forming a total of 7000 size of our database. We preprocessed and segmented these samples. We have used zoning density and background directional distribution to extract total of 144 features. We used SVM classifier with RBF Kernel for classification. We obtained 95.19% accuracy by 5-fold cross-validation of whole database. For training and testing purpose we divided the dataset into the ratio of 4:1 respectively. For unknown test data accuracy is 89.14% while for known test data it is observed as 99.92% giving the average test accuracy of 94.58%. In following sections we will study our proposed method in details.

A. Pre-processing

Originally we scanned handwritten sample in RGB format. Then we converted these samples into gray scale. We converted this gray image into binary image using a suitable threshold value. We also applied other preprocessing techniques like median filtration, dilation, some morphological operations to join unconnected pixels, to remove isolated pixels, to set neighbor pixel values in majority and to remove the spur pixels. Thus we obtained a clean and noise free binary single image of all samples written by single writer. By segmentation of this image we obtained each sample in isolated and clipped form, resized it into 32*32 pixels, and stored all samples in our database in operable form for recognition.

B. Feature Extraction

We have used two types of features for our experiment- *zoning density* (ZD) and *Background Directional Distribution* (BDD).

1) Zoning Density (ZD) Features

We have created 16 (4*4) zones of our 32*32 sized sample. By dividing the number of foreground pixels in each zone by total number of pixels in each zone i.e. 64 we obtained the density of each zone. Thus we obtained 16 zoning density features. (Fig.4)

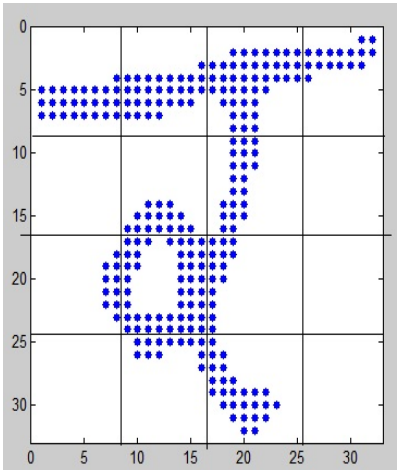

Fig.4. 16 zones of 32*32 normalized handwritten character Gurumukhi character ‘k’.

2) Background Directional Distribution (BDD) Features

For these features we have considered the directional distribution of neighboring background pixels to foreground pixels. We computed 8 directional distribution features. To calculate directional distribution values of background pixels for each foreground pixel, we have used following masks for each directional values (Fig.5). The pixel at center ‘X’ is foreground pixel under consideration to calculate directional distribution values of background.

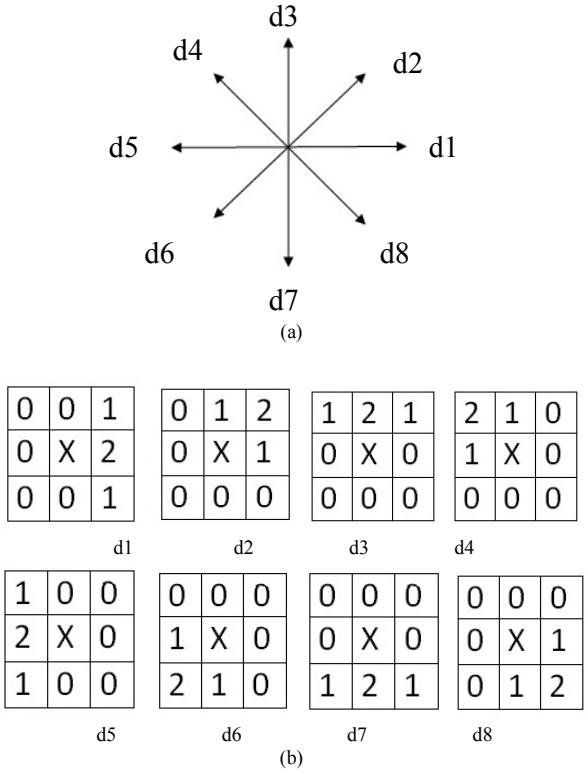

Fig.5. (a) 8 directions used to compute directional distribution, (b) Masks used to compute directional distribution in different directions.

To compute directional distribution value for foreground pixel 'X' in direction d1, for example, the corresponding mask values of neighboring background pixels will be added. Similarly we obtained all directional distribution values for each foreground pixel. Then, we summed up all similar directional distribution values for all pixels in each zone, described earlier in zoning density features description. Thus we finally computed 8 directional distribution feature values for each zone comprising total 128 (8*16) values for a sample image.

We combined both types of features extracted from zoning density and directional distribution forming totally 144 (16+128) features. SVM classifier uses these 144 features to for classification.

C. Classification

Support Vector Machines (SVM) classifier

For classification we chose Support Vector Machines (SVM) classifier. At present SVM is popular classification tool used for pattern recognition and other classification purposes. Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. The standard SVM classifier takes the set of input data and predicts to classify them in one of the only two distinct classes. SVM classifier is trained by a given set of training data and a model is prepared to classify test data based upon this model. For multiclass classification problem, we decompose multiclass problem into multiple binary class problems, and we design suitable combined multiple binary SVM classifiers. Our problem also needs to classify the characters into 35 different classes of Gurumukhi characters. We obtained such multiclass SVM classifier tool LIBSVM [26].

According to how all the samples can be classified in different classes with appropriate margin, different types of kernel in SVM classifier are used. Commonly used kernels are: *Linear kernel*, *Polynomial kernel*, Gaussian *Radial Basis Function* (RBF) and *Sigmoid* (*hyperbolic tangent*).

The effectiveness of SVM depends on kernel used, kernel parameters and soft margin or penalty parameter C.

The common choice is RBF kernel, which has a single parameter *gamma* (g or γ). We also have selected RBF kernel for our experiment. Best combination of C and γ for optimal result is obtained by grid search by exponentially growing sequence of C and γ and each combination is cross validated and parameters in combination giving highest cross validation accuracy are selected as optimal parameters.

In *V*-fold cross validation we first divide the training set into V equal subsets. Then one subset is used to test by classifier trained by other remaining V-1 subsets. By cross validation each sample of train data is predicted and it gives the percentage of correctly recognized dataset.

V. Results and Discussion

We have used 7000 samples of isolated handwritten Gurumukhi characters in our experiment. These samples are written by 20 different writers, each contributing to write 10 samples of each character out of 35 characters.

A. Optimal parameters selection and cross validation

Initially a random sample out of total dataset was taken to train the SVM classifier and we optimized the parameters C and g (or γ). Further by training the system by whole dataset the optimization of these parameters was refined. We finally selected the optimal parameters combination giving highest cross validation accuracy. Thus selected optimal parameters are- C = 11.3 and g = 0.28. With these parameters 5-fold cross validation accuracy obtained is 95.04% while 10-fold cross validation accuracy obtained is 95.29%.

B. Testing Performance

We have tested our system in two ways. First, test samples are taken from the new writers, i.e. the writers whose samples are not taken in train data. In this way, we have taken the dataset written by 16 writers (of 5600 size) to train the system and dataset of remaining 4 writers to test the system. Thus for testing the samples of unknown writers we obtained 89. 14%. Second, the samples written by each writer are divided into data to train and test the system in the ratio of 4:1. In this case the samples to test are distinct but their writers are known to

the system. In other words, the all writers whose handwritten samples are to test also participate to write their samples in training dataset. Thus, for known writers we obtained the 99.93% accuracy. The average accuracy for both known and unknown writers obtained is 94.54%.

C. Comparison with earlier approaches

The earlier approaches for isolated handwritten Gurumukhi character recognition are proposed by Puneet Jhajj et al. and Ubeeka Jain et al. [24], [25].

Puneet Jhajj et al. used zoning density features and compared the results obtained with two types of classifiers- SVM and K-NN. They observed the best result as 73.83% with SVM classifier using RBF kernel. Ubeeka Jain et al. used neocognitron neural network for isolated Gurumukhi character recognition. They reported 92.78% accuracy average to both known and unknown writers.

The following table (Table I) depicts the comparison of recognition results of our proposed methods with Puneet Jhajj et al.'s and Ubeeka Jain et al.'s methods.

TABLE-I
COMPARISON OF RECOGNITION METHODS FOR ISOLATED HANDWRITTEN GURUMUKHI CHARACTERS

Method	Feature extraction	Classification	Accuracy
Puneet Jhajj et al. [24]	zoning density	SVM with RBF kernel	73.83%
Ubeeka Jain et al. [25]	profiles, width, height, aspect ratio, neocognitron	Neocognitron Neural Network	92.78
Our proposed approach	zoning density and background directional distribution features	SVM with RBF kernel	95.04%

The approach can be extended in future to recognize characters in words or sentences by applying appropriate segmentation techniques. These experiments can be also analyzed and compared with more prominent classifiers other than SVM or at least with other kernels of SVM. While recognizing characters in words or sentences we will need to recognize these isolated characters after removing header line. Such type of basic character recognition will comprise the result for middle zone that can be combined with the result of top and bottom zones, and result can be combined to recognize compound character with top and bottom modifiers.

REFERENCES

[1] O. D. Trier, A. K. Jain and T. Text, "Feature Extraction Methods For Character Recognition- A Survey", Pattern Recognition, Vol. 29, No. 4, pp. 641-662, 1996.

[2] Snehal Dalal and Latesh Malik, "A Survey of Methods and Strategies for Feature Extraction in Handwritten Script Identification", First International Conference on Emerging Trends in Engineering and Technology, IEEE Computer Society, 2008.

[3] M. Cheriet, N. Kharma, Cheng-Lin Liu, Ching Y. Suen, "Character Recognition Systems: A Guide for Students and Practitioners", Wiley Publications, 2007.

[4] Gurmukhi (Punjabi) Script and Pronounciation. [Online]. Available (Accessed in April): http://www.omniglot.com/writing/gurmuki.htm

[5] Gurmukhi Script. [Online]. Available (Accessed in April 2011): http://www.sikh-history.com/sikhhist/events/gurmukhi.html

[6] Gurmukhi Alphabet Introduction. [Online]. Available (Accessed in April 2011): http://www.billie.grosse.is-a-geek.com/alphabet.html

[7] Gurmukhi Script Wikipedia. [Online]. Available (Accessed in April 2011): http://en.wikipedia.org/wiki/Gurmukh%C4%AB_script

[8] Online Punjabi Teaching. [Online]. Available (Accessed in April 2011): http://www.learnpunjabi.org/intro1.asp

[9] U. Pal, B.B. Chaudhury, "Indian Script Character Recognition: A Survey", Pattern Recognition Society, Elsevier, 2004.

[10] U. Pal, Wakabayashi, Kimura, "Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers", 10th International Conference on Document Analysis and Recognition, 2009.

[11] Vikas J Dungre et al., "A Review of Research on Devnagari Character Recognition", International Journal of Computer Application (0975-8887), Volume-12, No.2, November 2010.

[12] Sandhya Arora et al., "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA December 8-10, 2008.

[13] Lajish V.L. et al., "Fuzzy Directional Features for Unconstrained On-line Handwritten Character Recognition", IEEE, 2010.

[14] Prachi Mukherji, Priti Rege, "Shape Feature and Fuzzy Logic Based Offline Devnagari Handwritten Optical Character Recognition", Journal of Pattern Recognition Research 4 (2009) 52-68, 2009.

[15] Sanjay, Dayashankar, Maitreyee, "Handwritten Characterr Recognition Using Twelve Directional Feature Input and Neural Network", International Journal of Computer Application (0975-8887) Volume-1 No.3, 2010.

[16] G. S. Lehal, C. Singh, "A Gurmukhi Script Recognition System", Proceedings of 15th International Conference on Pattern Recognition, 2000, Vol. 2, pp. 557-560.

[17] G. S. Lehal, C. Singh, "A Complete Machine printed Gurmukhi OCR", Vivek, 2006.

[18] G.S. Lehal, C. Singh, "Feature Extraction and Classification for OCR of Gurmukhi Script", Vivek Vol. 12, 1999, pp. 2-12.

[19] G. S. Lehal, C. Singh, "A Post Processor for Gurmukhi OCR", Sadhana Vol. 27, Part 1, 2002, pp. 99-111.

[20] D. Sharma, G. S. Lehal, "An Iterative Algorithm for segmentation of Isolated Handwritten Words in Gurmukhi Script", The 18th International Conference on Pattern Recognition, IEEE Computer Society, 2006.

[21] V. Goyal, G.S. Lehal, "Comparative Study of Hindi and Punjabi Language Scripts", Nepalese Linguistics, Vol. 23, 2008, pp. 67-82.

[22] D. Sharma, G. S. Lehal, Preety Kathuria, "Digit Extraction and Recognition from Machine Printed Gurmukhi Documents", MORC Spain, 2009.

[23] Anuj Sharma, Rajesh Kumar, R.K. Sharma, "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching", Conference on Image and Signal Processing, IEEE Computer Society, 2008.

[24] Puneet Jhajj, D. Sharma, "Recognition of Isolated Handwritten Characters in Gurmukhi Script", International Journal of Computer Applications (0975-8887), Vol. 4, No. 8, 2010.

[25] Ubeeka Jain, D. Sharma, "Recognition of Isolated Handwritten Characters of Gurumukhi Script using Neocognitron", International Journal of Computer Applications (0975-8887), Vol. 4, No. 8, 2010.

[26] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm