Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features

Kartar Singh Siddharth [*], Mahesh Jangid, Renu Dhir, Rajneesh Rani

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

[*]kartarsiddharth@gmail.com

***Abstract*— In this manuscript handwritten Gurmukhi character recognition for isolated characters is proposed. We have used some statistical features like zonal density, projection histograms (horizontal, vertical and both diagonal), distance profiles (from left, right, top and bottom sides). In addition, we have used background directional distribution (BDD) features. Our database consists of 200 samples of each of basic 35 characters of Gurmukhi script collected from different writers. These samples are pre-processed and normalized to 32*32 sizes. SVM, K-NN and PNN classifiers are used for classification. The performance comparison of features used in different combination with different classifiers is presented and analysed. The highest accuracy obtained is 95.04% as 5-fold cross validation of whole database using zonal density and background distribution features in combination with SVM classifier used with RBF kernel.**

Keywords-isolated handwritten Gurmukhi character recognition; statistical features; zoning; density; projection histogram; distance profile; background directional distribution; SVM; K-NN; PNN; RBF kernel

I. INTRODUCTION

Optical character recognition is the prominent area of research in the world. OCR is the translation of scanned images of handwritten, typewritten or printed document into machine encoded form. This machine encoded form is editable text and compact in size.

Character recognition can be applied on printed, type-written or handwritten text. Character recognition for handwritten characters is more complex due to varying writing styles of people.

Further the optical character recognition can be classified as offline recognition and online recognition. The offline recognition is associated with static applications in which entire document first scanned and then processed to recognize, while the online recognition is associated with dynamic application as web application where we need recognized result simultaneously or within a fraction of time.

The pattern recognition mechanism is also classified traditionally 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. In feature based approach features of pattern are extracted and based on these features the patterns are classified using appropriate classifier or combination of classifiers. If the features and classifiers are chosen in efficient combination, the overall performance can be improved. However it will increase the complexity and time consumed.

To recognize characters we need to divide the document into classifiable objects. In a simple form, these classifiable objects can be directly inputted in the form of isolated characters to classify. To recognize the sentences or paragraphs, consideration of segmentation at line, word and character level and upto the level to decompose into classifiable objects is required. In context to most Indian scripts, many researchers has proposed the segmentation into three horizontal zones, in which upper and lower zones consist of modifiers and middle zone is framed by basic character set.

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

In pre-processing scanned document is converted to binary image and various other techniques to remove noise, to make it ready and appropriate for feature extraction are applied. These techniques include segmentation to isolated individual characters, skeletonization, contour making, normalization, filtration etc.

Feature extraction is the important step in character recognition, however other steps also need to be optimized because these steps are closely related to each other as outputs of earlier step is inputted to later step. Feature extraction is used to extract the most relevant information which is used to classify the objects. Most relevant is in the sense to minimize the within class pattern variability and to maximize the between class variability.

Features can be broadly classified into two categories: structural features and statistical features. Structural features are involved of structural elements like loop, line, crossing point, curve, end point and stroke etc. Statistical features are computed by some statistical operations on image pattern and these include features like zoning, projection, profiling, histogram and distance etc. Structural and statistical features appear complementary to each other and many other features can be derived from the basics of these features.

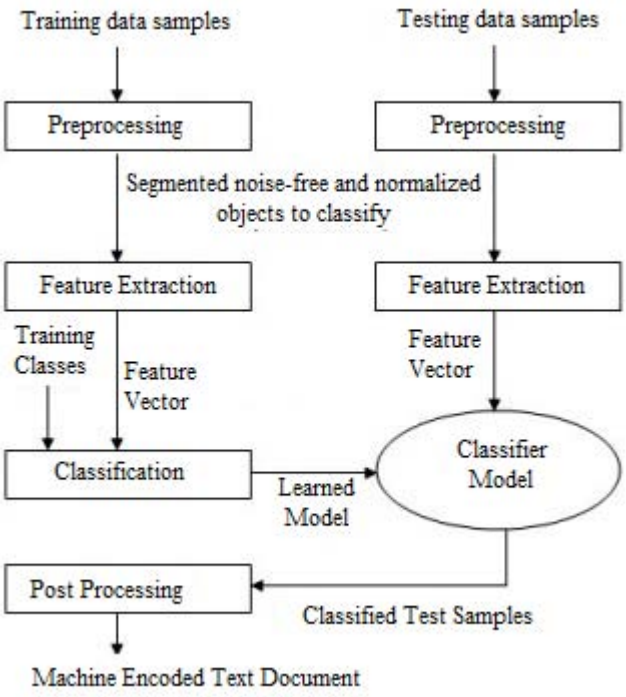


Fig.1. Basic Steps of Character Recognition

A detailed survey 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.

Each pattern having feature vector is classified in predefined classes using classifiers. Classifiers are first trained by a training set of pattern samples to prepare a model which is later used to recognize the test samples (see figure 1). The training data should consist of wide varieties of samples to recognize all possible samples during testing. Some examples of generally practiced classifiers are- Support Vector Machine (SVM), K- Nearest Neighbour (K-NN), Artificial Neural Network (ANN) and Probabilistic Neural Network (PNN) .

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 post processing step we bind up our work to create complete machine encoded document through the process of recognition, assigning Unicode values to characters and placing them in appropriate context to make characters, words, sentences, paragraphs and finally whole document.

II. INTRODUCTION TO GURMUKHI SCRIPT

Gurmukhi script, which is mainly used to write Punjabi language, consists of 35 basic characters. In addition to these 35 characters, there are 10 vowels and modifiers, 6 additional modified consonants, forming 41 consonants including 35 basic characters [4], [5]. (Table.1)

TABLE 1. GURMUKHI ALPHABET

Vowels and corresponding modifiers				
ਅ(none)	ਆ (ਾ)	ਇ (ਿ)	ਈ (ੀ)	ਉ (ੁ)
ਊ (ੂ)	ਏ (ੇ)	ਐ (ੈ)	ਓ (ੋ)	ਔ (ੌ)
Basic Characters (Consonants)				
ੳ	ਅ	ੲ	ਸ	ਹ
ਕ	ਖ	ਗ	ਘ	ਙ
ਚ	ਛ	ਜ	ਝ	ਞ
ਟ	ਠ	ਡ	ਢ	ਣ
ਤ	ਥ	ਦ	ਧ	ਨ
ਪ	ਫ	ਬ	ਭ	ਮ
ਯ	ਰ	ਲ	ਵ	ੜ
Additional Characters (with lower bindi)				
ਸ਼	ਖ਼	ਗ਼	ਜ਼	ਫ਼
ਲ਼				
Other symbols				
ੰ (tippi)	ਂ (bindi)	ੱ (adhak)	੍ (halant)	ਃ(visarg)

The three characters ੳ, ਅ and ੲ are called vowel carriers, because these carry all the vowels with additional modifiers (*matras*). Most of the Gurmukhi characters are like Devnagari script grouped in the sets of 5-5 characters which make 7 sections (*vergas*) of 35 basic characters. The sections ਕ to ਪ are arranged in the row depending on which part of mouth these characters are originated from and these are arranged in columns depending on how these are pronounced [4].

III. RELATED WORK

The Many researchers have worked on Indian script recognition in general and Gurmukhi in particular. A detailed survey on research work on Indian languages is presented in [6]. In this paper, properties of Indian scripts, methods and approaches applied to recognize are discussed.

Vikas Dungre et al. [7] have reviewed 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.

Prachi Mukherji and Priti Rege [8] have used 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.

Giorgos Vamvakas et al. [9], [10] have described the statistical and structural features they have used in their approach of Greek handwritten character recognition. The statistical features they have used are zoning, projections and profiling, and crossings and distances. Further through zoning they derived local features like density and direction features. In direction features they used directional histograms of contour and skeleton

images. In addition to normal profile features they described in- and out- profiles of contour of images. The structural features they have depicted are end point, crossing point, loop, horizontal and vertical projection histograms, radial histogram, radial out-in and in-out histogram.

Sarbajit Pal et al. [11] have described projection based statistical approach for handwritten character recognition. They proposed four sided projections of characters and projections were smoothed by polygon approximation.

Nozomu Araki et al. [12] proposed a statistical approach for character recognition using Bayesian filter. They reported good recognition performance in spite of simplicity of Bayesian algorithm.

Wang Jin et al. [13] evolved a series of recognition systems by using the virtual reconfigurable architecture-based evolvable hardware. To improve the recognition accuracy of the proposed systems, a statistical pattern recognition-inspired methodology was introduced.

Chain code histogram and moment based features were used in [14] while recognizing handwritten Devnagari characters. Chain code was generated by detecting the direction of the next in-line pixel in the scaled contour image. Moment features were extracted from scaled and thinned character image.

Four types of profiling horizontal, vertical and both diagonal are used in [15] to recognize handwritten Gujarati numerals.

Fuzzy directional features are used in [16] in which directional features were derived from the angle between two adjacent curvature points. This approach was used to recognize online handwritten Devnagari characters. 12 directional features were derived in [17] by computing gradient features by Sobel's mask, finding angles using tangent and categorizing the angle in one of the 12 directions.

In perticulat to Gurmukhi script, C. Singh and G. S. Lehal have done major work in the field of Gurmukhi character recognition. They have designed a complete printed Gurmukhi character recognition system [18].

Anuj Sharma et al. [29], [32] have presented the implementation of three approaches: elastic matching technique, small line segments and HMM based technique, to recognize online handwritten Gurmukhi characters and reported 90.08%, 94.59% and 91.95% recognition accuracies respectively. Dharam Veer Sharma et al. [30] first extracted Gurmukhi digits from printed documents and then recognised. They have used many structural features like loops, entry points, curve, line, aspect ratio, and statistical features like zoning, directional distance distribution for recognition and observed 92.6% recognition rate for Gurmukhi digits. For offline handwritten Gurmukhi character recognition two approaches are reported. First one is proposed by Puneet Jhajj et al. [20] and second one by Ubika Jain et al. [21]. A little more detailed survey on Gurmukhi recognition is presented in [6] and [19].

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 of 92.78% at average.

In the following sections we describe Data generation, Preprocessing, Feature Extraction, classification and finally results and discussion.

IV. PRE-PROCESSING

In our proposed methodology of isolated handwritten Gurmukhi character recognition we have considered 35 basic characters of Gurmukhi 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. The samples were collected such that these can be separated line by line through straight horizontal white spaced lines. Also the space between adjacent characters within line was present. The contributors to these data samples were of different educational backgrounds of metric, graduate, post graduate level qualification and different professions as student, teacher, security guard and hostel care-taker. We preprocessed and segmented these samples. Initially, we scanned handwritten these samples in RGB format.

In pre-processing step, we converted these samples into gray scale. Then, we applied following techniques:

- We converted these gray scale images into binary images using threshold value obtained by Otsu's method plus adding 0.1 to it.
- We applied median filtration, dilation, and removed noise having less than 30 pixels.
- We applied some morphological operation to bridge unconnected pixels, to remove isolated pixels, to smooth pixels boundary by majority and to remove spur pixels.

- Now, we segmented these samples first line wise then column wise within line in an iterative approach. The white space present was used to separate these lines and columns.
- We clipped the character images by removing extra white spaced rows and columns residing in four sides of image.
- We resized each character image into 32*32 pixel size.

Now, we stored all sample images such obtained in our database in matrix form for further recognition process.

V. FEATURE EXTRACTION

We have used following listed features for our experiment which are statistical features and background directional distribution features.

A. Zoning Density (ZD) Features

In zoning, the character image is divided into N×M zones. From each zone features are extracted to form the feature vector. The goal of zoning is to obtain the local characteristics instead of global characteristics. We have created 16 (4*4) zones of 8*8 size each out of our 32*32 normalized samples by horizontal and vertical division. 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.2)

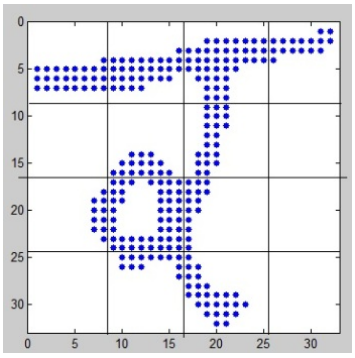
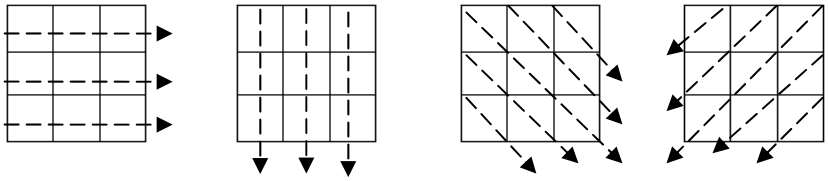


Fig.2. 16 zones of 32*32 normalized handwritten character Gurmukhi character ਕ (‘*k*’).

B. Projection Histogram Features

Projection histograms count the number of pixels in specified direction. In our approach we have used three directions of horizontal, vertical and diagonal traversing. We have created three types of projection histograms-horizontal, vertical, diagonal-1 (left diagonal) and diagonal-2 (right diagonal). These projection histograms for a 3*3 pattern are depicted in figure 3. (Fig. 3)

In our approach projection histograms are computed by counting the number of foreground pixels. In horizontal histogram these pixels are counted by row wise i.e. for each row. In vertical histogram the pixels are counted by column wise. In diagonal-1 histogram the pixels are counted by left diagonal wise. In diagonal-2 histogram the pixels are counted by right diagonal wise. The lengths of these features are 32, 32, 63 and 63 respectively according to lines of traversing.



(a) Horizontal Histogram (b) Vertical Histogram (c) Diagonal-1 Histogram (d) Diagonal-2 Histogram

Fig. 3. Evaluation of 4 types of Projection Histograms on 3*3 patterns

In the figure 4 sample image of 32*32 sized Gurmukhi character ਇ (‘*i*’) and its horizontal, vertical, diagonal-1 and diagonal-2 histograms are shown.

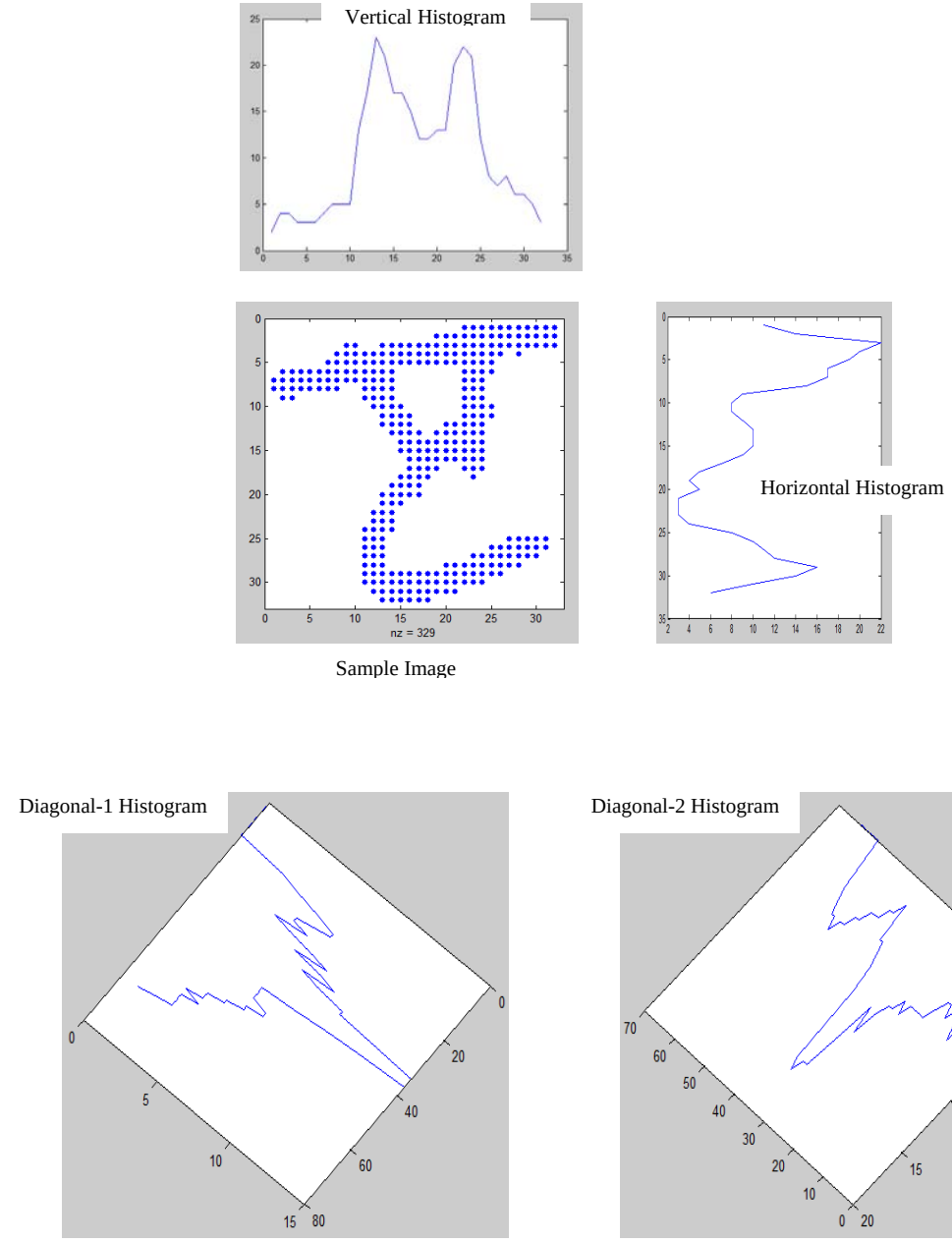


Fig.4. Sample of ਇ (‘i’) and its four histograms: horizontal, vertical, diagonal-1 and diagonal-2

C. Distance Profile Features

Profile counts the number of pixels (distance) from bounding box of character image to outer edge of character. This traced distance can be horizontal, vertical or radial. In our approach we have used profiles of four sides left, right, top and bottom. Left and right profiles are traced by horizontal traversing of distance from left bounding box in forward direction and from right bounding box in backward direction respectively to outer edges of character. Similarly, top and bottom profiles are traced by vertical traversing of distance from top bounding box in downward direction and from bottom bounding box in upward direction respectively to outer edges of character. The size of each profile in our approach is 32. The figure 5 depicts the sample image of character ਘ ‘*gh*’ and its four sided profiles.

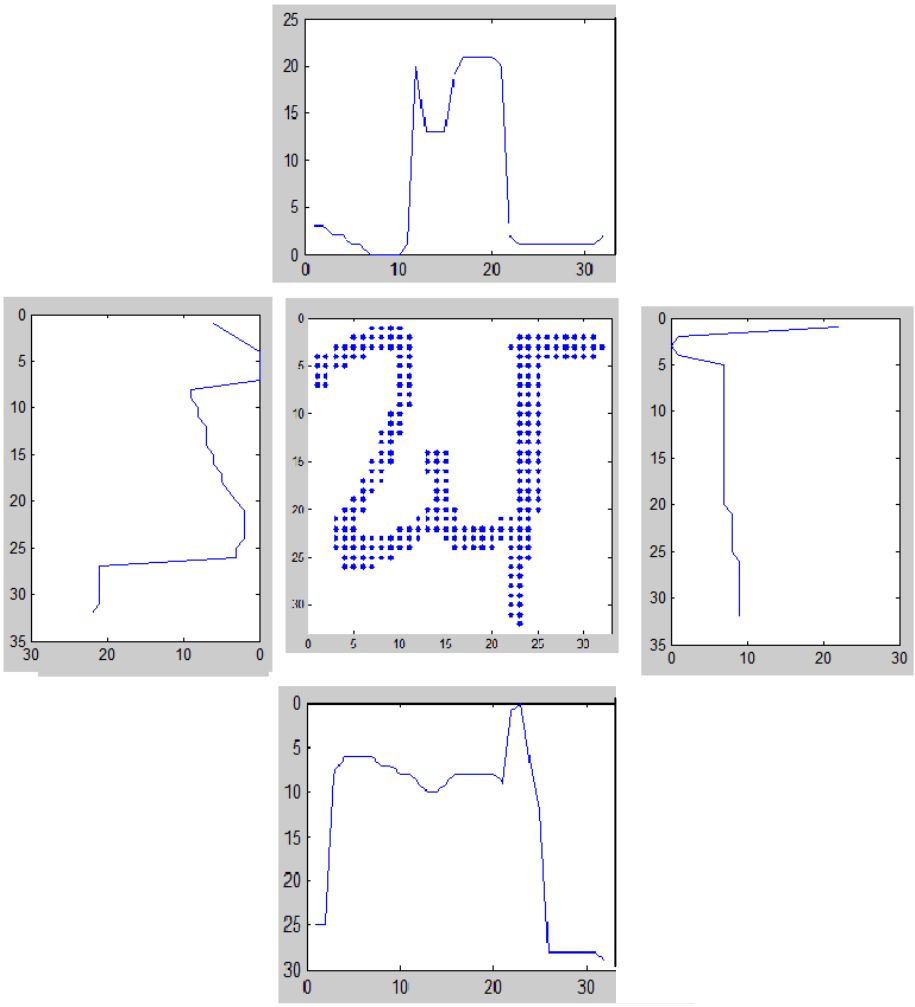
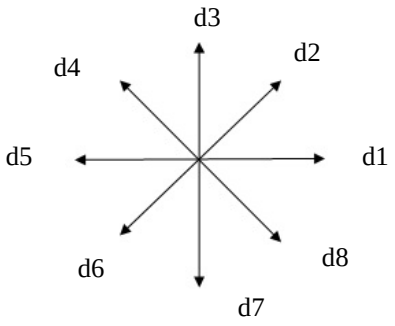


Fig.5. A sample image of character ਘ ‘gh’ and its left, right, top and bottom profiles.

D. Background Directional Distribution (BDD) 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 the masks for each direction shown in figure 6. The pixel at center ‘X’ is foreground pixel under consideration to calculate directional distribution values of background. The weight for each direction is computed by using specific mask in particular direction depicting cumulative fractions of background pixels in particular direction.



(a)

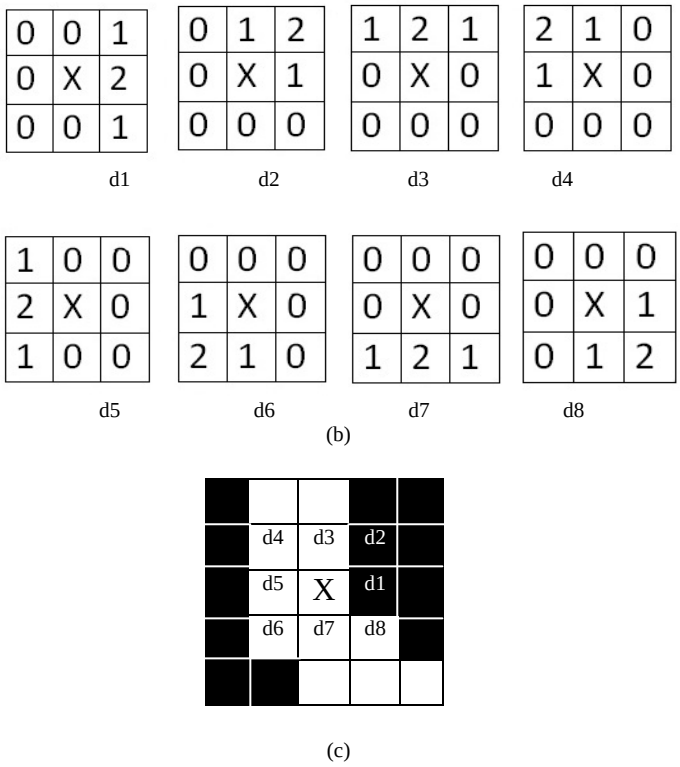


Fig.6. (a) 8 directions used to compute directional distribution, (b) Masks used to compute directional distribution in different directions. (c) An example of sample

To illustrate the computation of BDD features, we have to compute directional distribution value for foreground pixel 'X' in direction d1for the sample given in figure 6(c). We need to superimpose the mask for d1 direction on the sample image coinciding the centered pixel X. The sum of mask values of d1 mask coinciding to background pixels neighbouring to X in figure 6(c) i.e. d1 and d2 (i.e. 1+2) will be feature value in direction d1. 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. Zones are 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.

In addition to these originally extracted features, to form feature vectors we have also used different feature vectors derived by different combinations of above described and extracted features. These sets of feature vectors, we consider in our implementation, are specified in table 2.

TABLE 2. THE SETS OF FEATURE VECTORS

Feature Vector	Included Features	Size
FV1	Zonal Density (ZD)	16
FV2	Profiles	128
FV3	Histograms	190
FV4	BDD	128
FV5	Profiles + ZD	144
FV6	BDD + ZD	144
FV7	Profiles + horizontal and vertical histograms(HVH)	192
FV8	BDD + HVH	192
FV9	BDD + Profiles	256
FV10	BDD + Diagonal (both) histograms	254

It can be observed, FV1 to FV4 each consists of single type of features while FV5 to FV10 consist of different combinations of the features FV1 to FV4.

VI. CLASSIFICATION

We have used three types of classifiers in our implementation- SVM, K-NN and P-NN.

A. Support Vector Machines (SVM) classifier

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 Gurmukhi characters. We obtained such multiclass SVM classifier tool LIBSVM available at [26]. A practical guide for SVM and its implementation is available at [27].

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.

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.

B. K- Nearest Neighbour (K-NN) classifier

K-NN 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 neighbour. Because it considers only neighbour object to a particular level, it uses local approximation of distance function. 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 neighbours to be considered and the class of majority of these neighbours is determined as the class of unknown pattern [22], [23]. In the figure 7 the classification of objects with k=3 (solid line sircle) and k=5 (dotted circle) is shown. It is notable that class of object is altered in both cases. At k=3 the unknown sample will be classified as triangular shape object while at k=5 it will be classified as diamond shape object.

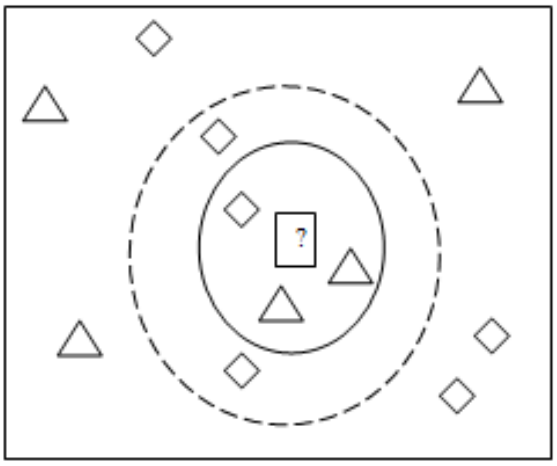


Fig. 7. Classification of objects using K-NN with k=3 and 5

The distance function used to find nearest neighbours can be specified as Euclidean, sum of absolute differences, cosine, correlation and percentage of bits that differ.

C. Probabilistic Neural Network (PNN) classifier

PNN is a multilayered feed-forward neural network classifier in which known probability density function (pdf) of the population is used to classify unknown patterns. PNN is closely related to Parzen window pdf estimator. In our implementation we have used Parzen PNN tool. The estimated pdf reaches to true pdf as the training set size increases, as long as the true pdf is smooth.

If the probability density function (pdf) of each of the populations is known, then an unknown, X, belongs to class "i" if:

$$\boldsymbol{f_i(X) > f_j(X),\ all\ j \neq i,}\ \text{f}_\text{k} \text{ is the pdf for class k.}$$

The pdf estimation for a single population, in general, is given by the average of pdf's for the n samples in the population. The pdf will be estimated by following formula:

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

where,

x = unknown (input),

$x_k = k^{th}$ sample,

W = weighting function,

σ = smoothing parameter.

Commonly we use Gaussian function in the place of weighting function. More details about PNN and Parzen pdf can be found in [24] and [25].

VII. Results and Analysis

A. 5-fold Cross Validation

In our implementation we have used 5-fold cross validation with the three classifiers. First we created randomly generated 5-fold cross-validation index of the length of size of dataset. This index contains equal proportions of the integers 1 through 5. These integers are used to define a partition of whole dataset into 5 disjoint subsets. We used one division for testing and remaining divisions for training. We did so 5 times, each time changing the testing dataset to different division and considering remaining divisions for training. Thus we got 5 sets of feature vectors containing training and testing dataset in the size ratio of 4:1.

The average recognition accuracy of these randomly generated 5 sets of training and testing is referred as cross validation accuracy.

In the table 3 5-fold cross-validation results using SVM, k-NN and PNN classifiers are shown. The corresponding parameters of these classifiers, at which these optimized results are obtained, are also shown. Features used in these feature vectors are shown in table 2. For selection of these parameters to obtain optimized results, first we used small sample of whole dataset and observed the parameters giving highest results. Later we refined this optimization by cross validation of whole dataset.

In SVM classifier, the results vary significantly on small values of C. These results are more sensitive to change with parameter g of RBF kernal comparative to C. At larger values of C results are stable and variation is negligible. Most of the results of SVM listed are observed at larger range of C tested upto 500, while the values of kernel parameter used are listed.

The listed K-NN results are obtained by using 'Euclidian' distance parameter, while the results on 'correlation' parameter were observed to be lower in all cases, and these results are not listed here. The values for k parameter used in K-NN and smoothing parameter σ used in PNN classifier are also listed in the table. The optimized results for K-NN classifier are obtained in the range of 1 to 6 of parameter k.

Figure 8 shows the comparison of the results obtained with ten feature vectors and three classifiers.

Table 5 shows the confusion matrix of Gurmukhi characters recognized using SVM classifier. This confusion matrix is drown from recognition results of one of the 5 sets of training and testing used in 5-fold cross validation. This table gives 95% recognition rate. From confusion matrix it can be observed that characters confuse with similar types of other characters. The most confusions are: ਖ with ਘ, ਪ, ਮ ; ਛ with ਝ, ਢ ; ਜ with ਚ, ਲ ; ਡ with ਚ, ਤ, ਰ ; ਢ with ਚ; ਥ with ਖ, ਬ, ਰ; ਦ with ਕ, ਚ, ਰ etc. these confusions cause the decreased recognition performance.

TABLE 3. OPTIMIZED CROSS VALIDATION RESULTS WITH DIFFERENT CLASSIFIERS

FV# → Classifier ↓	FV1	FV2	FV3	FV4	FV5	FV6	FV7	FV8	FV9	FV10
SVM (γ)	82.4571 (1.8)	85.1714 (0.22)	90.1286 (0.2)	93.2571 (0.22)	91.6286 (0.07)	95.0412 (0.28)	92.1429 (0.05)	93.6857 (0.20)	93.9286 (0.07)	94.4286 (0.15)
k-NN (k)	76.0000 (1, 2)	67.7429 (6)	84.0857 (5)	81.2143 (1, 2)	73.5000 (4)	85.0286 (1, 2)	76.5857 (6)	84.2857 (6)	79.5571 (4)	86.2429 (1, 2)
PNN (σ)	77.2000 (0.15)	70.6000 (0.34)	84.8857 (0.27)	83.3000 (0.23)	76.7143 (0.31)	86.9571 (0.24)	79.4857 (0.36)	85.9857 (0.30)	81.7286 (0.33)	87.8286 (0.29)

B. Comparison with Earlier Approaches

Naveen Garg et al. [31] have recognized offline handwritten Gurmukhi characters using neural network and obtained 83.32% average recognition accuracy.

Anuj Sharma in his Ph.D. thesis [32] has recognized the online handwritten Gurmukhi characters using three approaches. In first approach [29] online Gurmukhi characters are recognized using elastic matching algorithms by matching unknown stroke against the strokes database giving 90.08% recognition accuracy. The second approach, called small line segments, is based on elastic matching and chaincode algorithm and gives 94.59% recognition accuracy. The third approach which is based on hidden Markov model, gives 91.95% accuracy.

For offline handwritten Gurmukhi character recognition two more approaches are reported, one is proposed by Puneet Jhajj et al. [20] and another one by Ubeeka Jain et al. [21]. In both approaches work is done on isolated characters. In Puneet Jhajj et al.'s approach zoning density features were used and the results obtained with two types of classifiers SVM and K-NN were compared. They observed the best result as 73.83% with SVM classifier using RBF kernel. In Ubeeka Jain et al.'s approach neocognitron neural network was used for isolated Gurmukhi character recognition. They reported 92.78% accuracy average to both known and unknown writers.

The table 4 depicts the comparison of our proposed approach with all these approaches.

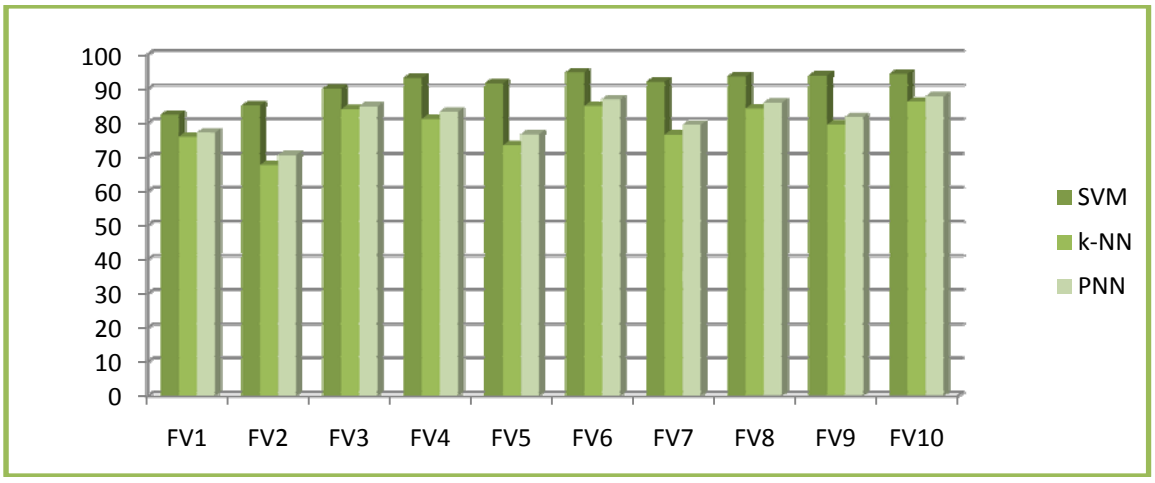


Fig 8. Performance comparison of different feature vectors with three classifiers

TABLE 4. COMPARISON WITH EARLIER APPROACHES

Proposed by	Features used	Classifier	Accuracy
Naveen Garg et al. [31]	Structural Features	Neural Network	83.32%
Anuj Sharma et al. [29]	Strokes recognition and matching	Elastic matching	90.08%
Anuj Sharma et al. [32]	Small line segments	Elastic Matching	94.59%
Anuj Sharma et al. [32]	HMM elements	Hidden Markov model	91.95%
Puneet Jhajj et al. [20]	Zoning density (64)	SVM with RBF kernel	73.83%
Ubeeka Jain et al. [21]	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%

VIII. CONCLUSIONS

At FV6 the highest accuracy obtained is 95.04% using combination of most relevant zonal density and background distribution features in combination with SVM classifier used with RBF kernel. Both the features used in FV6 are zone based, i.e. extraction of local features is more useful than global features used in other feature vectors. Out of these local features Background directional distribution features have major contribution to gain optimal recognition results. It is because these features consist of directional as well as local features. The combination of profile and histogram features with BDD (in FV8, FV9, and FV10) is comparatively less relevant to gain best results. Additionally these combinations of features have a drawback of increased size of feature vector, increasing the computing complexity.

In all listed feature vectors, best results are obtained with SVM classifier, then secondly by PNN classifier and K-NN produced lowest results comparatively.

The work can be extended to increase the results by using or adding some more relevant features. We can use some features specific to the mostly confusing characters, to increase the recognition rate. We can divide the entire character set to apply specific and relevant features differently. More advanced classifiers as MQDF or MIL can be used and multiple classifiers can be combined to get better results.

To extend the work to string level recognition first line, word and character level will be required, then segmentation of characters into top, middle and bottom horizontal zones to separate upper and bottom modifiers will be required. Then these zoned objects can be recognized individually and these results can be placed in suitable context with other character or elements to form machine encoded text document equivalent to input document image.

TABLE 5. CONFUSION MATRIX OF GURMUKHI CHARACTERS CLASSIFIED WITH SVM

Sr. No.	Character	No. of Samples	Recognized	Misclassified with	Accuracy (%)
1	ੳ	35	35	NIL	100.00
2	ਅ	43	42	1(ਜ)	97.67
3	ੲ	35	35	NIL	100.00
4	ਸ	43	41	1(ਗ), 1(ਲ)	91.11
5	ਹ	42	41	1(ਕ)	97.62
6	ਕ	42	40	1(ਬ), 1(ੜ)	95.24
7	ਖ	44	40	1(ਘ), 1(ਪ), 3(ਮ)	88.89
8	ਗ	38	37	1(ਸ)	97.37

9	ਘ	40	37	1(ਅ), 2(ਪ)	92.5
10	ਙ	31	31	NIL	100
11	ਚ	39	38	1(ਥ)	97.44
12	ਛ	45	43	1(ਙ), 1(ਢ)	95.56
13	ਜ	36	33	2(ਚ), 1(ਲ)	91.67
14	ਝ	43	42	1(ਣ)	97.67
15	ਞ	45	42	3(ਵ)	93.33
16	ਟ	36	35	1(ਦ)	97.22
17	ਠ	35	35	NIL	100.00
18	ਡ	28	25	1(ਚ), 1(ਤ), 1(ਰ)	89.29
19	ਢ	42	37	5(ਚ)	88.10
20	ਣ	46	44	1(ਹ), 1(ਙ)	95.65
21	ਤ	32	31	1(ਭ)	96.88
22	ਥ	45	39	1(ਖ), 4(ਬ), 1(ਰ)	86.67
23	ਦ	42	38	1(ਹ), 1(ਕ), 1(ਚ), 1(ਰ)	88.37
24	ਧ	37	36	1(ਸ)	97.30
25	ਨ	41	40	1(ਠ),	97.56
26	ਪ	43	43	NIL	100.00
27	ਫ	40	39	1(ਢ)	97.50
28	ਬ	45	40	5(ਥ),	88.89
29	ਭ	41	39	1(ਓ), 1(ਲ)	95.12
30	ਮ	35	34	1(ਖ)	97.14
31	ਯ	47	44	1(ਗ), 1(ਧ), 1(ਬ)	93.62
32	ਰ	33	31	1(ਚ), 1(ਡ)	93.94
33	ਲ	44	41	1(ਸ), 2(ਥ)	93.18
34	ਵ	46	45	1(ਭ)	97.83
35	ੜ	37	37	NIL	100.00
Total		**1400**	**1330**	**70**	**95.00**

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

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

[6] U. Pal, B.B. Chaudhury, “Indian Script Character Recognition: A Survey”, *Pattern Recognition Society, Elsevier*, 2004.

[7] Vikas J Dungre et al., “A Review of Research on Devnagari Character Recognition”, *International Journal of Computer Applications* (0975-8887), Volume-12, No.2, November 2010.

[8] 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.

[9] Vamvakas, G.; Gatos, B.; Petridis, S.; Stamatopoulos, N.; , "An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition," *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* , vol.2, no., pp.1073-1077, 23-26 Sept. 2007.

[10] Vamvakas, G.; Gatos, B.; Petridis, S.; Stamatopoulos, N.; et al., “Optical Character Recognition for Handwritten Characters” ppt, [Online]. Available: http://www.iit.demokritos.gr/IIT_SS/Presentations/Off-Line%20Handwritten%20OCR.ppt. Accessed in 2010.

[11] Sarbajit Pal, Jhimli Mitra, Soumya Ghose, Paromita Banerjee, "A Projection Based Statistical Approach for Handwritten Character Recognition," in *Proceedings of International Conference on Computational Intelligence and Multimedia Applications*, vol. 2, pp.404-408, 2007.

[12] Araki, N.; Okuzaki, M.; Konishi, Y.; Ishigaki, H.; , "A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter," *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on* , vol., no., pp.194, 18-20 June 2008.

[13] Wang Jin; Tang Bin-bin; Piao Chang-hao; Lei Gai-hui; , "Statistical method-based evolvable character recognition system," *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on* , vol., no., pp.804-808, 5-8 July 2009.

[14] S. Arora, D. Bhattacharjee, M. Nasipuri, M.Kundu, D.K. Basu, "Application of Statistical Features in Handwritten Devnagari Character Recognition", *International Journal of Recent Trends in Engineering* [ISSN 1797-9617], IJRTE Nov 2009.

[15] Apurva A. Desai, "Gujarati Handwritten Numeral Optical character Recognition through Neural Network", *Pattern Recognition* Volume 43 Issue 7, July, 2010.

[16] Lajish, V.L.; Kopparapu, S.K.; , "Fuzzy Directional Features for unconstrained on-line Devanagari handwriting recognition," *Communications (NCC), 2010 National Conference on* , vol., no., pp.1-5, 29-31 Jan. 2010.

[17] D. Singh, S.K. Singh et al., "Handwritten Character Recognition Using Twelve Directional Feature Input and Neural Network", *International Journal of Computer Applications* (0975-8887), Vol.1, No.3, 2010.

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

[19] Kartar Siddharth, Renu Dhir, Rajneesh Rani, "Handwritten Gurumukhi Charater Recognition Using Zoning Density and Background Directional Features", (IJCSIT) *International Journal of Computer Science and Information Technologies*, Vol. 2, Issue 3, pp. 1036-1041, May-June 2011.

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

[21] 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.

[22] Wikipedia website- K-Nearest Neighbour Algorithm, [Online]. Available: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[23] K-NN classifier, [Online]. Available: http://www.fon.hum.uva.nl/praat/manual/kNN_classifiers_1__What_is_a_kNN_classifier_.html

[24] The University of Reading Website [Online]. Avaiable: http://www.personal.reading.ac.uk/~sis01xh/teaching/CY2D2/Pattern2.pdf

[25] The University of Reading Website [Online]. Avaiable: http://www.personal.reading.ac.uk/~sis01xh/teaching/CY2D2/Pattern3.pdf

[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

[27] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification", [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[28] 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

[29] Anuj Sharma, Rajesh Kumar, R. K. Sharma, "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching," *Image and Signal Processing, 2008. CISP '08. Congress on* , vol.2, no., pp.391-396, 27-30 May 2008

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

[31] Naveen Garg, Karun Verma, "Handwritten Gurmukhi Charcter Recognition Using Neural Network", M.Tech. Theis, Thapar University, 2009 [online]. Available: http://dspace.thapar.edu:8080/dspace/bitstream/10266/788/1/thesis+report+final.pdf

[32] Anuj Sharma, R.K. Sharma, Rajesh Kumar, "Online Handwritten Gurmukhi Character Recognition", Ph.D. Thesis, Thapar University, 2009 [Online]. Available: http://dspace.thapar.edu:8080/dspace/bitstream/10266/1057/3/Thesis_AnujSharma_SMCA_9041451.pdf

AUTHORS PROFILE

Kartar Singh Siddharth is currently an M.Tech. (Computer Science& Engineering) student of 2009-11 batch at Dr B R Ambedkar National Institute of Technology, Jalandhar. He completed his B.E. in 2009 from Government Engineering College Bikaner affilated to University of Rajasthan. His areas of interest and research are pattern recognition, image processing and character recognition.

Mahesh Jangid is currently also M.Tech. (Computer Science & Engineering) student of 2009-11 batch at Dr B R Ambedkar National Institute of Technology, Jalandhar. He completed his B.E. in 2007 from RIET, Jaipur affiliated to University of Rajasthan. Earlier to M.Tech. he has 2 years' teaching experience from JECRC, Jaipur. His interested research areas are image processing, pattern recognition and optical character recognition.

Renu Dhir is currently an associate professor in the department of computer science at Dr B R Ambedkar NIT, Jalandhar. She completed her M.Tech. (Computer Science & Engineering) in 1997 from TIET and Ph.D. in 2007 from Punjabi University. Her fields of research are mainly character recognition, pattern recognition and image processing and she has published more than 35 papers in national and international conferences and journals.

Rajneesh Rani is currently an assistant profesor in department of computer science at Dr B R Ambedkar NIT, Jalandhar. She is simalteniously pursuing her Ph.D. from the same institute. She completed her M.Tech. in Computer Science and Engineering in 2003 from Punjabi University, Patiala. She has teaching experience of more than 7 years. She has published many research papers in the field of pattern recognition.