Comparative Recognition of Handwritten Gurmukhi Numerals Using Different Feature Sets and Classifiers

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

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

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

ABSTRACT
Recently there is an emerging trend in the research to recognize handwritten characters and numerals of many Indian languages and scripts. In this manuscript we have practiced the recognition of handwritten Gurmukhi numerals using three feature sets and three classifiers. Among three feature sets, first feature set is comprised of distance profiles having 128 features. Second feature set is comprised of different types of projection histograms having 190 features. Third feature set is comprised of zonal density and Background Directional Distribution (BDD) forming 144 features. The three classifiers used are SVM, PNN and K-NN. The SVM classifier is used with RBF (Radial Basis Function) kernel. We have observed the 5-fold cross validation accuracy in the case of each feature set and classifier. We have obtained the optimized result with each combination of feature set and classifier by adjusting the different parameters. The results are compared and trends of result in each combination of feature set and classifier with varying parameters is also discussed. With PNN and K-NN the highest results are obtained using third feature set as 98.33% and 98.51% respectively while with SVM the highest result is obtained using second feature set as 99.2%. The results with SVM for all feature sets are higher than the results with PNN and K-NN.

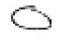
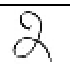
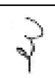
General Terms — Handwritten Recognition; Projection histogram.

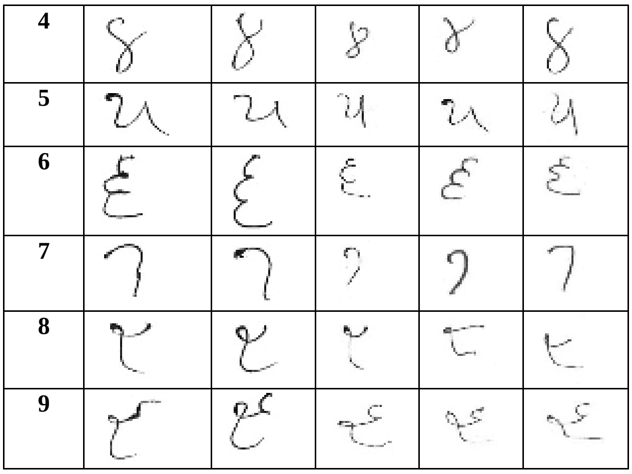
Keywords — Handwritten Gurmukhi numeral recognition, Zonal densityDistance Profiles, Background Directional Distribution (BDD), SVM, PNN, K-NN, RBF kernel.

1. INTRODUCTION
In Indian scripts recently many approaches are available to recognize handwritten characters and numerals. The higher accuracy of numeral recognition is achievable due to lower number of classes equal to 10 different digits comparative to the higher classes equal to different characters in case of characters.
The recognition rate can be increased to much extent by training the system by large number of samples collected from large group of writers.
Many approaches practiced related to numeral recognition using various techniques and algorithms are reviewed here.
G.S. Lehal and N. Bhatt have applied their system to recognize both English and Devnagari numerals [1]. They have used a set of global and local features derived from left and right projection profiles.
Reena Bajaj et al. [2] have recognized Devnagari numerals using three different types of features namely density, moment and descriptive component features. They have used three **different** neural classifiers in their work and finally combined the outputs using a connectionist scheme.
U. Bhattacharya and B.B. Chaudhuri [3] have used the features based on wavelet transforms at different resolution levels and multilayer perceptron for classification purpose to recognize hand-printed numerals.
U. Pal et al. [4] have recognized the offline handwritten numerals of six Indian scripts namely Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil scripts.
Sonatakke and Patil et al. [5] have recognized handwritten Devnagari numerals using general fuzzy hyperline segment neural network. They have implemented a rotation, scaling and translation invariant algorithm. The achieved recognition rate is 99.5%.
Shrivastava and Gharde [7] have recognized handwritten Devnagari numerals using moment invariant and affine moment invariant techniques for feature extraction and SVM classifier. The have obtained 99.48% recognition rate.
M.Jangid and K. Singh et al. [8] have practiced to recognize handwritten Devnagari numerals. They have adopted a feature extraction technique based on recursive subdivision of the character image. They have observed 98.98% recognition rate using SVM classifier.
Apurva Desai [6] have used four types of profiles namely horizontal, vertical, left and right diagonal profiles to recognize Gujarati numerals. He has used neural network for classification and achieved 82% accuracy.

2. DATASET
The dataset of Gurmukhi numerals for our implementation is collected from 15 different persons. Each writer contributed to write 10 samples of each of numeral of 10 different Gurmukhi digits. These samples are taken on white papers written in an isolated manner.
The table 1 shows some of the samples of our collected dataset. These samples are transformed in gray image.

TABLE 1: HANDWRITTEN SAMPLES OF GURMUKHI NUMERALS

Digit	Samples				
0					
1					
2					
3					



3. FEATURE EXTRACTION

We have used following three sets of features extracted to recognize Gurmukhi numerals in the form of feature vectors. These approaches are adopted from our earlier practice [9] used to recognize isolated Gurmukhi handwritten characters.

I. Distance Profile Features [FV1]
II. Projection Histogram Features [FV2]
III. Zonal density and Background Directional Distribution (BDD) Features [FV3]

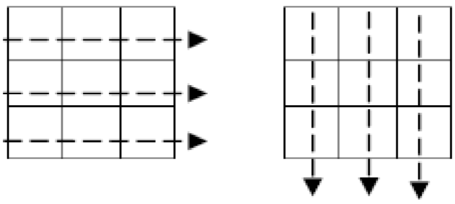
3.1 Distance Profile Features

In our approach we have used distance profiles using distance computation from bounding box to outer edges of character from four sides- two in horizontal direction from left and right sides and other two in vertical direction from top and bottom side.

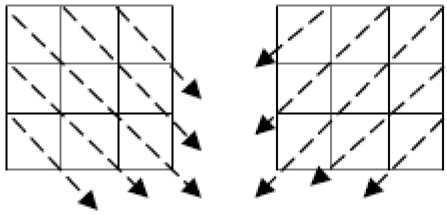
Left and right profiles are traced by horizontal traversing of distance from left bounding box in forward direction and from right bounding box in backward direction respectively to outer edges of character. Similarly, top and bottom profiles are traced by vertical traversing of distance from top bounding 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 similar to number of pixels in each row or column forming total 128 features by all four types of profiles.

3.2 Projection Histogram Features

Projection histograms count the number of foreground pixels in specified direction. In our approach we have used four directions of horizontal, vertical and both diagonal traversing.



(a) Horizontal Histogram (b) Vertical Histogram



(c) Diagonal-1 Histogram (d) Diagonal-2 Histogram

Figure 1 Evaluation of 4 types of Projection Histograms on 3×3 patterns

Thus in our approach we have created four 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 1.

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 pixel 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 forming total 190 features.

3.3 Zonal Density and Background Directional Distribution 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 pixel 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.

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 2(b). 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.

Following algorithm describes the computation of BDD features for each character image of size 32×32 pixels having 4×4 zones and thus each zone having 8×8 pixel size. Eight directional features D1,D2, … D8 are computed for each zone.

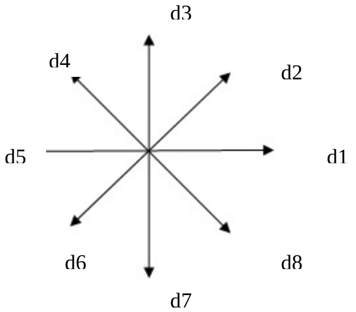
Algorithm for computation of BDD features

// image size=32×32, number of zones =16 // (4×4), pixels in each zone=64(8×8)

(1) START
(2) Initialize D1,D2,... D8 = 0;
(3) Repeat step 4 for each zone Z(r,c)
//where, r=1,2,3,4; c=1,2,3,4;
(4) Repeat step 5 for each pixel P(m,n) of the zone Z(r,c)
//where, m=1,2,...8; n=1,2,...8;
(5) **IF** pixel P(m,n) is foreground
Then, 8 distribution features (d1,d2 …d8) for zone Z(r,c) are computed by summing up the mask values specified for neighbouring background pixels in particular direction (see fig 2(b)) as follows:

D1= D1 + 2 × ~P(m,n+1) + ~P(m-1,n+1) + P(m+1,n+1);
D2 = D2 + 2× ~P(m-1,n+1) + ~P(m-1,n) + ~P(m,n+1);
D3 = D3 + 2× ~P(m-1,n) + ~P(m-1,n-1) + ~P(m-1,n+1);
D4 = D4 + 2× ~P(m-1,n-1) + ~P(m,n-1) + ~P(m-1,n);
D5 = d5 + 2× ~P(m,n-1) + ~P(m-1,n-1) + ~P(m+1,n-1);
D6 = d6 + 2× ~P(m+1,n-1) + ~P(m,n-1) + ~P(m+1,n);
D7 = d7 + 2× ~P(m+1,n) + ~P(m+1,n-1) + ~P(m+1,n+1);
D8 = d8 + 2× ~P(m+1,n+1) + ~P(m+1,n) + ~P(m,n+1);

(6) STOP



(a) 8 directions

0	0	1
0	X	2
0	0	1

0	1	2
0	X	1
0	0	0

1	2	1
0	X	0
0	0	0

2	1	0
1	X	0
0	0	0

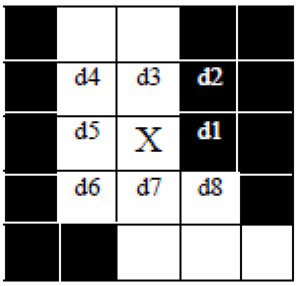
1	0	0
2	X	0
1	0	0

0	0	0
1	X	0
2	1	0

0	0	0
0	X	0
1	2	1

0	0	0
0	X	1
0	1	2

(b) Masks used to compute different directional distributions



(c) An example of 3×3 sample
(d)

Figure 2 Computation of Background Directional Features

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 2(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 2(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.

4. RECOGNITION RESULTS AND ANALYSIS

We have observed the 5-fold cross validation results with all classifiers using different feature sets. In 5-fold cross validation we first divide whole dataset randomly into 5 equal sets. Then we consider one set for testing and remaining four sets for training. Thus we alter the testing set for each of five sets and consider remaining four sets for training the system. We consider the average accuracy of these 5 sets of training and testing combinations as 5-fold cross validation accuracy. Below the results obtained with the different classifiers are discussed.

We first tested with all possible parameters and observed the results obtained and then by refinement we finally discovered the parameters or their combinations giving optimum cross validation accuracy.

4.1 Results with SVM Classifier

We have used SVM classifiers for recognition. Basically SVM classifies objects into binary classes but it can be extended to classify multiple classes. We have obtained such multiclass SVM tool LIBSVM available at [10]. We have used RBF (Radial Basis Function) kernel which is also common choice, in our recognition. RBF has single kernel parameter *gamma* (g or γ). Additionally there is another parameter with SVM classifier called soft margin or penalty parameter (C).

The table 2 depicts the optimized results obtained with different features set at refined parameters. The result variation is more sensitive to value γ comparative to C.

Table 2 Numeral Recognition Results With Svm

Feature Set	Recognition Rate	Parameters
Distance Profiles (128) [FV1]	98.13%	C = 8; γ = 0.12
Projection Histograms (190) [FV2]	99.2%	C > 8; γ = 0.22 - 0.5
Zonal density and BDD (144) [FV3]	99.13%	C > 8; γ = 0.48 - 0.58

While observing the results at other values of parameter C, it is analysed that increasing the value of C irrespective of any change in γ slightly decreases the recognition rate, but after a certain increment normally after 64 at higher values of C the recognition rate becomes stable. In contrast, the recognition rate always changes with the change in γ.

4.2 Results with PNN Classifier

Probabilistic Neural Network is a special type of Neural Network classifiers. It is a multi-layered 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. Smoothing parameter σ is used in PNN classifier.

The highest results obtained with PNN using three feature sets FV1, FV2 and FV3 is shown in table 3.

Table 3 Numeral Recognition Results With Pnn

Feature Set	Recognition Rate	Parameters
FV1	96%	σ = 0.30
FV2	98%	σ = 0.30
FV3	98.33%	σ = 0.15, 0.20

From the table 3 it is clear that with PNN the highest result is obtained with FV3 as 98.33%.

5. RESULTS WITH 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. The highest results observed with the three feature sets (FV1, FV2 and FV3) are listed in table 4 with corresponding parameter K. From the table 4 it can be observed that the highest result with K-NN is obtained using FV3 feature set as 98.51%. The obtained with FV1 and FV2 are 94.58% and 97.72% respectively.

In the case of PNN and K-NN the highest results are obtained with FV3, But in case of SVM the highest result is obtained using FV2. This result is also highest among all the results with all the classifiers. All the results with SVM are highest comparative to other classifiers- PNN and K-NN.

In the figure 3 all the highest results obtained with three classifiers using each of feature set are compared.

The various results with PNN classifier at different σ is shown in figure 4. Similarly the results with K-NN at different values of parameter K are shown in figure 5.

In the table 5 confusion matrix for numeral recognition is given. This confusion matrix is obtained by testing a dataset samples using SVM and giving 99% result.

Table 4 Numeral Recognition Results With K-Nn

Feature Set	Recognition Rate	Parameters
FV1	94.58%	K = 3
FV2	97.72%	K = 1
FV3	98.51%	K = 1

Table 5 Confusion Matrix Of Numeral Recognition

Digit	Classified or misclassified as										Accuracy (%)
	0	1	2	3	4	5	6	7	8	9	
0	27	0	0	0	0	0	0	0	0	0	100
1	0	29	0	0	0	0	0	0	0	0	100
2	0	1	33	0	1	0	0	0	0	0	94.29
3	0	0	0	35	0	0	0	0	0	0	100
4	0	0	0	0	31	0	0	0	0	0	100
5	0	0	0	0	0	32	0	0	0	0	100
6	0	0	0	0	0	0	22	0	0	0	100
7	0	0	0	0	0	0	0	24	0	0	100
8	0	0	0	0	0	0	0	0	34	0	100
9	0	0	0	0	0	0	0	0	1	30	96.77
Average accuracy											99

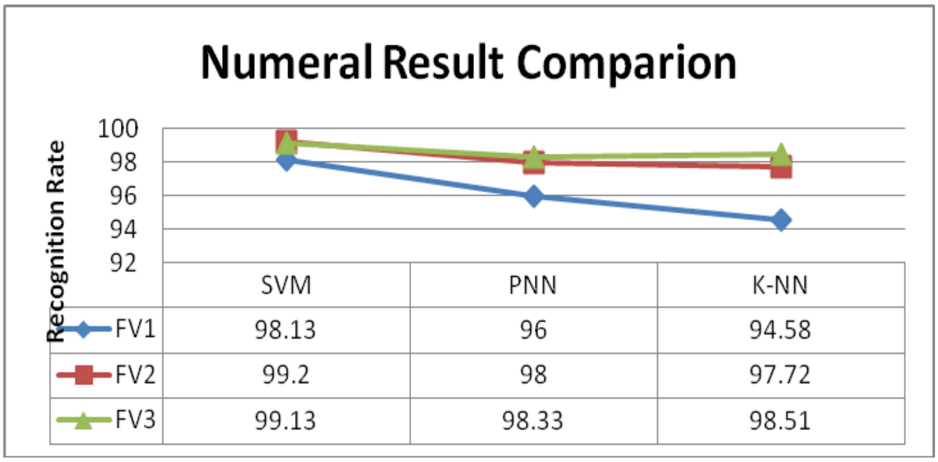


Figure 3 Comparison of Numeral results with the three classifiers and different feature sets

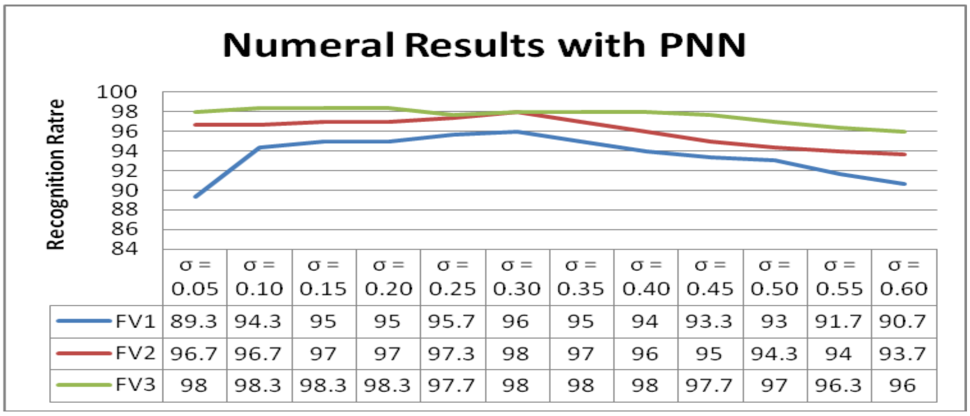


Figure 4 Results of numeral recognition with PNN at different values of σ

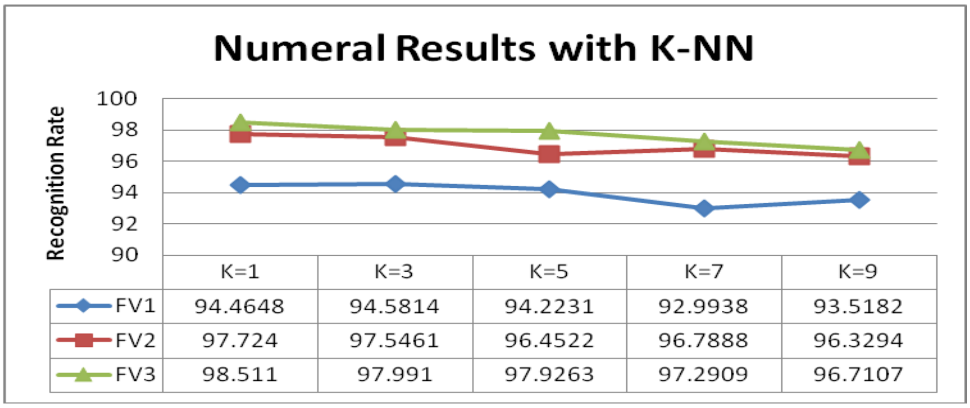


Figure 5 Results of numeral recognition with K-NN with different values of K

By the results observed with SVM, we can conclude that we have obtained the maximum recognition rate as 99.2% at 190 projection histogram features among all three feature sets. Secondly, we obtained 99.13% recognition rate at 144 zonal density and Background Directional Distribution features. Thirdly, we obtained 98.13% recognition rate at 128 distance profile features. It is clear here that feature set having large number of features is giving higher recognition rate.

In terms of efficiency and performance, recognition results using third feature set are best. It is because it has slight reduction in recognition rate (99.2% to 99.13%) but a significant reduction in number of features (190 to 144) comparative to second feature set. Hence it reduces computation and time complexity while processing lesser number of features.

REFERENCES

[1] G.S. Lehal, Nivedan Bhatt, 2000. A Recognition System for Devnagari and English Handwritten Numerals, Proc. ICMI, Springer, pp. 442-449,.

[2] Reena Bajaj, Lipika Dey, Shantanu Chaudhuri, February 2002. Devnagari numeral recognition by combining decision of multiple connectionist classifiers, Sadhana Vol. 27, Part 1, pp. 59–72,.

[3] U. Bhattacharya, B. B. Chaudhuri, 2003. A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals, icdar, vol. 1, pp.16, Seventh International Conference on Document Analysis and Recognition (ICDAR'03) - Volume 1, pp. 29-33,.

[4] U. Pal, T. Wakabayashi, N. Sharma, F. Kimura, 2007. Handwritten Numeral Recognition of Six Popular Indian Scripts, Document Analysis and Recognition,. ICDAR, Vol.2, pp.749-753.

[5] P.M. Patil, T.R. Sontakke, 2007. Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network, Pattern Recognition, Elsevier, Vol. 40, Issue 7, pp. 2110-2117, July.

[6] Apurva A. Desai, 2010. Gujarati Handwritten Numeral Optical Character Reorganization through Neural Network, Pattern Recognition, Vol. 43, Issue 7, pp. 2582-89

[7] Shailedra Kumar Shrivastava, Sanjay S. Gharde, October 2010. Support Vector Machine for Handwritten Devanagari Numeral Recognition, International Journal of Computer Applications, (July 2010), Vol. 7, No. 11, pp. 9-14

[8] Mahesh Jangid, Kartar Singh, Renu Dhir, Rajneesh Rani, 2011, Performance Comparison of Devanagari Handwritten Numerals Recognition, Internation Journal of Computer Applications, (May 2011), Vol. 22, No.1,pp. 1-6.

[9] Kartar Singh Siddharth, Mahesh Jangid, Renu Dhir, Rajneesh Rani, 2011. Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features, IJCSE, (June 2011), Vol. 3, No. 6, pp. 2332-2345.

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