Comparative Analysis of Classifiers Inaccuracies for Bilingual Characters (Gurmukhi and Roman)

Renu Dhir

Department of Computer Science and Engineering, National Institute of Technology, Jalandhar
E-mail : renu_dhir@yahoo.com

ABSTRACT

Recognition of bilingual script in an image of a document page is of primary importance for a system processing bilingual document. In this paper two essential sub stages of recognition phase, feature extraction and classification has been discussed for segmented Gurmukhi and Roman characters. Structural and Statistical features have been proposed for machine printed bilingual characters. Three classifiers such as k-nearest neighbor (KNN), artificial neural networks (ANN) and Support Vector Machines (SVM) schemes have been proposed for analyzing the error rate for recognition. It has been found that error rate is dropped when statistical features are used simultaneously with structural features. It has also been observed that SVM outperform among all the classifiers.

Keywords: *Bilingual Script, Connected Component Neuro- Fuzzy, nearest neighbor, segmentation, simplified fuzzy*

1. INTRODUCTION

India is multi-lingual multi-script country, which has many languages (more than twenty) with their own distinctive scripts. Roman script words are now commonly being used in Gurmukhi script documents. Feature extraction and classification are two very important stages for OCRs. Trier et al. [1] summarized and compared some of the well-known feature extraction methods for off-line character recognition. They discussed feature extraction methods in terms of invariance properties, reconstructability and expected distortions and variability of the characters. Features can be extracted from binary images, binary contours or gray-scale images. The main disadvantage of using gray-scale image is memory requirements and low compression rate.

Lehal and Singh [2] have suggested structural features for feature extraction and used binary decision trees and nearest neighbor classifiers, for recognition of machine printed Gurmukhi characters. Jain, Duin, Mao [3] summarizes and compares the well-known method used in various stages of a pattern recognition system. They have introduced all most all the techniques of classification, clustering, feature extraction, feature selection, error, estimation, and classification combinations. Based upon problem one can make a picture in their mind to select a particular method in pattern recognition. The classification stage is the main decision making stage of an OCR system and uses the extracted features as input to identify the text segment according to preset rules. Performance of the system largely depends upon the type of the classifier used. The most straightforward 1-NN rule can be conveniently used as a benchmark for all the other classifiers since it appears to provide a reasonable classification performance in most applications and does not require any user-specific parameters (except perhaps the distance metric used to find the nearest neighbor, but "*Euclidean distance*"

is commonly used), its classification results are implementation independent. It, however, suffers from problems of accuracy and memory.

Neural network is one of the techniques that have been used for pattern recognition since 1950s and neural networks are preferred for pattern recognition problems because of their parallel processing capabilities as well as learning and decision-making abilities. The fusion of Neural Networks and Fuzzy Systems, termed Neuro-Fuzzy Systems has also been applied for the solution of various pattern recognition applications

RajaSekaran and Pai [4-6] have investigated the capability of SFAM to behave as a Pattern Recognizer/ Classifier of images for both noisy and noise free using moment based features. G. SVM (Support Vector Machine) classification algorithms, proposed by Vapnik [7] used to solve two-class problem, are based on finding a separation between hyper planes defined. It can be used for multi-classification by decomposing the problem into binary classification sub-problems. A. Statnikoy, Aliferis and S. Levy in their paper [8] have described multi-category classification methods for micro array gene expression cancer diagnosis using SVM. For multi-class classification, binary SVMs are combined, in either one-against-one (pair wise OVO) scheme or one-against-rest (OVR) or directed acyclic graph (DAGSVM). Due to the high complexity of training and execution, SVM classifiers have been mostly applied to small category set problems. The SVM classifier with RBF kernel mostly gives the highest accuracy.

2. CHARACTERISTICS OF THE GURMUKHI AND ROMAN SCRIPTS

In our approach, the features that play a major role are the spatial spread of the words formed by the scripts and the direction of orientation of the structural elements of the characters in the word. So, a brief description of the properties of the associated scripts (Roman and Gurmukhi) is in order, for the clear understanding that leads to the proper design of an identifier system. Some of the relevant properties are:

(1) The words of both the scripts, Roman and Gurmukhi, can be divided into 3 distinct zones Fig. 1 and Fig. 2 shows the upper, middle and lower zone for Gurmukhi and roman script.

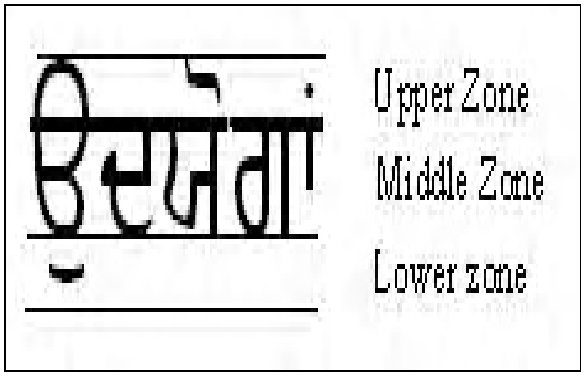

Fig. 1: Three zones of Gurmukhi word

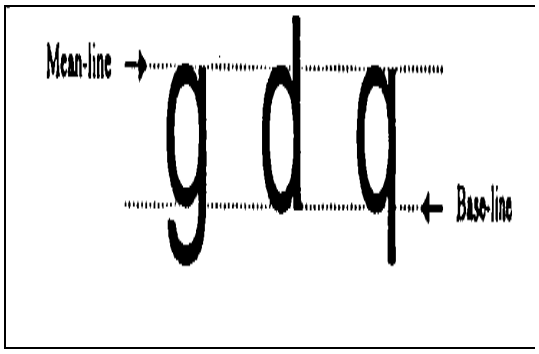

Fig. 2: Three distinct zones of Roman script

(2) The Roman script has 26 each of upper and lower case characters. In addition, there are some special symbols and numerals. The Gurmukhi script has 35 characters, 12 vowels, 18 consonants and 6 special characters.

(3) While the capital letters of the Roman script occupy the middle and the upper zones, most of the lower case characters have a spatial spread that covers only the middle zone or the middle and the upper zones.

(4) Apart from some special symbols (such as ‘,’), only the characters ‘g’, ‘j’, ‘p’, ‘q’ and ‘y’ spread spatially to the lower zone in case of Roman characters. These being few in number, the probability of their occurrence are small.

(5) The aspect ratio (ratio between the width and the height of the bounding box of a character) of the Gurmukhi characters is, different from Roman characters.

Roman characters are distributed along x-axis only. Gurmukhi characters are distributed along x-axis as well as y-axis.

3. PROPOSED RECOGNITION SYSTEM FOR SEGMENTED GURMUKHI AND ROMAN CHARACTERS

The proposed character recognition system for Roman and Gurmukhi characters is composed of the following phases:

1. Segmentation
2. Feature extraction
3. Classification and Recognition

3.1 Segmentation

In our present work, the segmentation process is performed in three successive stages: line segmentation, word segmentation and character segmentation. For line segmentation horizontal and projection profiles are employed while vertical projection profiles are used for word and character segmentation.

Instead of character segmentation we have performed connected component segmentation. The segmentation stage breaks up a word and characters which lie above and below the headline into connected components and the classifier has been trained to recognize these connected components (CC) or sub-symbols and headline is not considered the part of the connected component

Roman characters are distributed along x-axis only. Gurmukhi characters are distributed along x-axis as well as y-axis. We have taken segmented characters of Roman and Gurmukhi characters for Classification and Recognition.

3.2 Feature extraction stage

The feature extraction stage analyzes a text segment and selects a set of features that can be used to uniquely identify the text segment. The derived features are then used as input to the character classifier.

Prior to the recognition stage, some set of features is extracted from the image. These features are a reduced representation of the contents of the image which focus on preserving the characteristics that are most relevant to the task of recognition while eliminating those characteristics that are relevant to, or may confuse the discrimination power of the recognition stage. Extracted features are represented as a vector of values that are then passed to the classifier.

Following types of approaches are used in feature extraction.

- *Structural / syntactic features*: It tries to infer structure from the image data. It describes a pattern in terms of its topology and geometry. For example number of end points, Percentage of headline coverage, Number of branches from the headline, Presence of side bar concave arcs, intersection between characters and straight lines, bend points, tee points and Aspect Ratio etc.
- *Statistical features:* They look only at predefined feature vector, which provide only partial information about the shape. Statistical features are statistical measures of distribution of points on the bitmap, the contour curve, the profiles, or the HV-projections.
- *Moments:* Moments are pure statistical measure of pixel distribution around center of gravity of characters and allow capturing global character shapes information. Moments [9] describe numerical quantities at some distance from a reference point or axis. Moments are applicable to many different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation [13-15]. When applied to images, they describe the image content (or distribution) with respect to its axes. Moment invariants are considered reliable features if their values are insensitive to the presence of image noise.

- *Zoning:* The zoning method is simple and easy to implement. Its principle is to divide the bitmap in K non-overlapping regions or zones generally a 4x4 grid is superimposed on the character image and for each of the 16 zones; the average black pixel count is computed giving feature vector of length 16.

  It is not invariant to rotation and scaling. Scaling can be overcome by size normalizing the bitmaps before feature extraction. In order to achieve good recognition rates, the number of regions may be big, resulting in large feature vectors and therefore long classification times.

- *Projections:* Histogram of horizontal and vertical projections of black pixels is used.

B.1 Proposed Feature set for Gurmukhi and Roman characters:

Structural features are:

(1) *Number of branches from the headline*: The numbers of branches from the headline are noted. Thus for example, ਸ has two branches while ਕ has one branch. Total no of this feature is one (1).

(2) *Presence of side bar:* The presence of side bar in the character image is determined. For example it is present in the A while it is absent in the C. Total no of this feature is one (1).

(3) *Number of endpoints and their location*: A black pixel is considered to be an end point if there is only one black pixel in its 3 x 3 neighborhood. The number of end points and their positions in terms of 9 (3x3) quadrants of the character image are noted. For example the character ਹ has 3 endpoints in quadrants 1, 3 and 4. Total number of these features is nine (9).

(4) *Number of junctions (Tee points) and their location*: A black pixel is considered to be junctions if there are more than two black pixels in its 3 x 3 neighborhood. The number of junctions as well as their positions in terms of 9(3x3) quadrants is considered. For example, the character **ਹ** has 1 junction in quadrant 3 and ਜ has 3 junctions in quadrant 3, 4, 6. For Roman characters **T** has 1 junction in quadrant 2 and **X** has 1 junction in quadrant 5. Total number of this feature is nine (9).

(5) *Horizontal Projection Count:* Horizontal Projection Count represented as $HPC(i) = \sum_i F(i, j)$, where F(i,j) is a pixel value (0 for background and 1 for foreground) of a document image, and i and j denote vertical and horizontal coordinates of the pixel respectively, when the image's top left corner is set to F(0,0). Scanning the image row-wise and finding the sum of foreground pixels in each row calculate it. To take care of variations in character sizes, the horizontal projection count of a character image is represented by percentage instead of an absolute value and in our present work it is stored as a 4 component vector where the four (4) components symbolize the percentage of rows with 1 pixel, 2 pixels, 3 pixels and more than 3 pixels. Total number of this feature is four (4).

(6) *Vertical Projection Count*: Vertical Projection Count represented as $VPC(j) = \sum_j F(i, j)$, where F(i,j) is a pixel value (0 for background and 1 for foreground) of a document image, and i and j denote vertical and horizontal coordinates of the pixel respectively, when the image's top left corner is set to F(0,0). Total numbers of this feature are four (4).

(7) *Right Profile Direction Code*: The right profile is scanned from top to bottom and local directions of the profile at each pixel are noted. Starting from current pixel, the pixel distance of the next pixel in left; downward or right direction is noted. The cumulative count of movement in three directions is represented by the percentage occurrences with respect to the total number of pixel movement and stored as a 3 component vector with the three components representing the distance covered in left, downward and right directions respectively. Total number of this feature are three (3)

(8) *Left Profile Direction Code:* The left profile is scanned from top to bottom and local directions of the profile at each pixel are noted as described

above. Total number of this feature are three (3)

(9) *Top Profile Direction Code:* The top profile is scanned from left to right and local directions of the profile at each pixel are noted as described above. Total number of this feature are three (3)

(10) *Bottom Profile Direction Code*: The bottom profile is scanned from left to right and local directions of the profile at each pixel are noted as described above. Total number of this feature are three (3)

(11) *Aspect Ratio*: Aspect ratio, which is obtained by dividing the character height by its width for Roman and Gurmukhi characters. Total number of this feature is one (1)

(12) *Number of loops*: This feature appears if the character image forms a loop with the headline. Thus it is present in **l** and **b** while it is absent in n and x. Total no of this feature is one (1).

Statistical features:

(1) *Zoning:* A 7x7 grid is superimposed on the character image and for each of the 7x7 zones; the average black pixel count is computed giving a feature vector of length 7x7=49. It is found that 7x7 zoning grid gives best and optimal results as compared to other size of grids in our case for normal fonts12/14. Total numbers of this feature are forty- nine (49).

(2) *Geometrical Invariant moments:* We have used a set of eleven invariant functions including moments given by Hu [10] and Flusser and Suk [11] for experimental purpose, which are translation, scale and rotation. Total number of this feature is eleven (11).

(3) *Zernike Moments (ZM):* Zernike moments of order 12 are calculated. With experiments it has been found that as we increase order of moment from 12, quality increases, but accuracy start decreasing and best results are obtained at 12 order moments. We obtained total of 49 feature vector set, but first two features are ignored as scale and translation invariancy stage does affect two of these Zernike features as explained by A. Khotanzad [12] that first is going to be same for all images and second is equal to zero. Total number of this feature is forty- seven (47).

(4) *Pseudo-Zernike moments (PZM):* We have used 27 (Twenty seven) pseudo Zernike moments for present bilingual system. Best and good quality reconstructed images are obtained at order 6 for all the characters. Total number of this feature is twenty - seven (27).

(5) *Orthogonal Fourier Mellin moments (OFMM):* We have used 35 (Thirty five) Fourier Mellin moments for bilingual system. Best and good quality reconstructed images are obtained at order 7 for all the characters. Total number of this feature is thirty-five (35).

Total feature vector set (structural + statistical) = (169+ 42) = 211 (Two hundred and eleven)

Original unthinned image is retained for the statistical features.

3.2 Classification

The classification stage is the main design making stage of an OCR system and uses the features extracted in the previous stage to identify the text segment according to preset rules.

In the last stage of recognition of Gurmukhi and Roman characters, merging of characters take place. The information about coordinates of bounding box of sub-symbols and context is used to merge some of the sub-symbols in case of Gurmukhi characters.

For isolated characters of Gurmukhi, different font sizes such as 10, 12, and 14 of different font styles such as normal, bold and thin are taken. Different quality levels are also considered such as good medium and worst. A complete data base is made in MS access by taking different fields such as *Character Name, Sources, Quality, Font Style, Font Size, File name* etc. The tagged corpus is stored in a data base, which can easily be used for testing and training purpose by other researchers also. For detailed analysis we have not considered *matras*. We have used three types of classifiers: k-Nearest neighbor, neural networks and SVM. All the experiments with SVM have been conducted using one against one method with voting.

Fig. 3a: Samples of machine printed Gurmukhi characters used for Training

Fig. 3b: Samples of machine printed Gurmukhi characters used for Testing.

Fig. 3c: Samples of machine printed Roman characters used for Training

Fig. 3d: Samples of machine printed Roman characters used for Testing

We have used neural network tool Nunet pro 2.3 for windows for classification of Roman and Gurmukhi characters and also analyzed all the results using MATLAB version 7.2.

For classification data files are separated into two sets: — training and testing. We have used data file consisting of approximately 10000 samples of black and white machine printed Gurmukhi and Roman characters. We have taken 20 different categories of Gurmukhi and Roman for training and testing. The samples of machine printed Gurmukhi and Roman characters used in recognition for training and testing is shown in Fig. 5a, 5b, 5c and 5d. Gurmukhi characters of different font styles such as Anmol/ Amrit, Anmol Kalmi, Assees, Gurmukhi 010 Wide, Samtol, Satlej, Punjabi Sans, Likhari-P, Gurbani Akhar thick etc are used. Approximately 1000 samples of Gurmukhi and Roman each are taken for training and testing from the main data file. TABLE 1and 2 shows the effect of different features in recognition of Gurmukhi and Roman characters with KNN, ANN and SVM as classifier.

It has been observed that there is increase of approximately 1% in the accuracy for Gurmukhi characters as compared with Roman characters.

With experiments it has been observed that low recognition accuracy for Roman characters is due to confusion between upper and lower case letters of Roman characters. It further increases if we ignore this confusion.

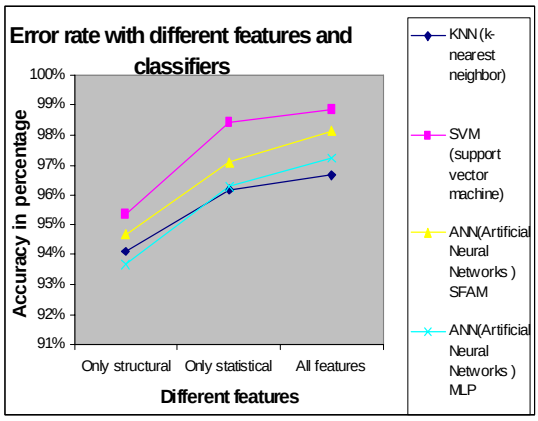

Fig. 4a: Error rate with different features and classifiers for Gurmukhi characters

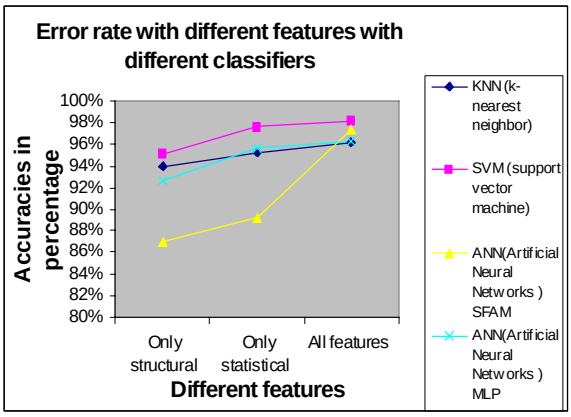

Fig. 4b: Error rate with different features and classifiers for Roman characters

Table 1: Classifiers inaccuracies using different sets of features on Gurmukhi characters

Different types of classifiers		Only structural in %age	Only statistical in %age	All features in %age
KNN (k-nearest neighbor)		94.12	96.14	96.63
SVM (support vector machine)		95.37	98.42	98.86
ANN (Artificial Neural Networks)	SFAM	94.70	97.10	98.14
	MLP	93.67	96.27	97.24

Table 2: Classifiers inaccuracies using different sets of features on Roman characters

Different types of classifiers		Only structural in %age	Only statistical in %age	All features in %age
KNN (k-nearest neighbor)		93.97	95.24	96.13
SVM (support vector machine)		95.17	97.68	98.13
ANN (Artificial Neural Networks)	SFAM	86.92	89.26	97.37
	MLP	92.63	95.66	96.37

4. COMPARISON OF PRESENT WORK WITH EXISTING WORK

It is fair to compare our results with existing work for different OCRs.

Gurmukhi OCR developed by Lehal and Singh [2] is based on structural features only. They have used nearest neighbor classifier by taking the value of k as 1 and accuracy achieved is 94.40% without post processing and 97.32% using post processing techniques. But, in present work system accuracy is 94.12% with structural features and KNN classifier without applying any postprocessor. There is small drop of accuracy of 0.28% (94.40% - 94.12%) of present system as compared with system developed by Lehal and Singh. There are many reasons for the drop of this accuracy. It may be due to data file used for testing, or may be due to some different implementations or design issues.

5. CONCLUSION

In this paper, the bilingual character recognition capability for Gurmukhi and Roman isolated characters has been discussed. Classifiers inaccuracies using different set of features for Gurmukhi and Roman characters are compared. The results obtained are quite encouraging. The model employs structural and statistical feature extractors.

It is observed that a recognition accuracy of 98.86% for Gurmukhi characters and 98.13% for Roman characters is achieved with SVM, which is considered as best classifier for present system. A recognition accuracy of 98.14% is achieved with neural networks for the recognition of Gurmukhi characters and 97.37% for Roman characters by ignoring upper and lower case letters confusion. With nearest neighbor classifier (KNN), recognition accuracy is low as compared to NN for Gurmukhi

and Roman characters. All these results are without applying any post-processing operations. SVMs are trained by QP and the training time is generally proportional to the square of number of samples. SVM learning by QP often results in a large number of SVs which should be stored and computed in classification. Neural classifiers have much less parameters, and the number of parameters is easy to control. Neural classifiers consume less storage and computation than SVMs. Biggest limitation of the support vector approach lies in the choice of the kernel. Second limitation is speed and size, both in training and testing.

This work can be extended for the development of bilingual OCR dealing with degraded, noisy machine printed text and italic text.

REFERENCES

1. O. D. Trier, A. K. Jain and T. Taxt, "Feature Extraction Methods for Character Recognition: – A survey", *Pattern Recognition,* Vol. 29(4), pp. 641-662, 1996.
2. G S Lehal and Chandan Singh, "Feature extraction and classification for OCR of Gurmukhi script", *Vivek*, Vol. 12(2), pp. 2-12, 1999.
3. A. K. Jain, P. W. Duin and J. Mao, "Statistical Pattern Recognition: – A Review", *IEEE Transactions on PAMI*, Vol. 22(1), pp. 1-37, 2000.
4. S. Rajasekaran and G. A. V. Pai, "*Application of Simplified Fuzzy ARTMAP to structural engineering problems*," All India Seminar on Application of NN in Science, Engineering and Management, Bhubaneswar, June 1997.
5. S. Rajasekaran and G. A. Vijayalakshmi Pai, "*Image Recognition using Simplified Fuzzy ARTMAP Augmented with a Moment Based Feature Extractor*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14 (8), pp. 1081-1095, 2000.
6. S. Rajasekaran and G. A. Vijayalakshmi Pai, "*Simplified Fuzzy ARTMAP As Pattern Recognizer*", Journal of Computing in Civil Engineering, Vol. 14 (2), pp. 92-99, 2000.
7. V. Vapnik, "Statistical Learning Theory, New York, NY, USA *Wiley-Interscience*, 1998.
8. A. Statnikov, F. Aliferis and Shawn Levy, "A Comprehensive Evaluation of Multi-Category Classification Methods for Micro Array Gene Expression Cancer Diagnosis" *Oxford journals*, 2004.
9. Cho-Huak Teh and R. T. Chin, "On Image Analysis by the Method of Moments, *IEE Transactions on PAMI, Vol. 10(4),* 496-513, 1988.
10. M. K. Hu, "Visual Pattern Recognition by Moment Invariants", *IRE Transactions on information Theory, Vol. 8,* pp. 179-187, 1962.
11. J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants", *Pattern Recognition,* Vol. 26(1), pp. 167-174, 1993.
12. A. Khotanzad and Y. H. Hong, "Invariant Image Recognition by Zernike Moments, *IEEE Transactions on PAMI, Vol. 12(5),* 489-497, 1990.
13. C. Kan and M. D. Srinath, "On the Accuracy of Zernike Moments for Image Analysis, *Pattern Recognition, Vol. 35,* 143-154, 2002.
14. Y. Sheng and L. Shen, "Orthogonal Fourier-Mellin moments for invariant Pattern Recognition, *J. Optical Society of America, Vol. 11(6),* 17481757, 1994.
15. Chandan Singh, "Improved quality of reconstructed images using floating point arithmetic for moment calculation, *Pattern Recognition, Vol. 39(11),* 2047-2064, 2006.