Feature Extraction and Classification for OCR of Gurmukhi Script

G. S. Lehal and Chandan Singh
Department Of Computer Science & Engineering
Punjabi University,
Patiala – 147002, INDIA
E-mail : gslehal@mailcity.com, chandan@pbi.ernet.in

Abstract

In this paper, a feature extraction and hybrid classification scheme, using binary decision tree and nearest neighbour, for machine recognition of Gurmukhi characters is described. The classification process is carried out in three stages. In the first stage, the characters are grouped into three sets depending on their zonal position (upper zone, middle zone and lower zone). In the second stage, the characters in middle zone set are further distributed into smaller sub-sets by a binary decision tree using a set of robust and font independent features. In the third stage, the nearest neighbour classifier is used and the special features distinguishing the characters in each subset are used. One significant point of this scheme, in contrast to the conventional single-stage classifiers where each character image is tested against all prototypes, is that a character image is tested against only certain subsets of classes at each stage. This enhances computational efficiency.

1. Introduction

Optical Character Recognition (OCR) is the process of converting scanned images of machine printed or hand-written text into a computer processable format. The process of optical character recognition of any script can be broadly broken down into five stages:

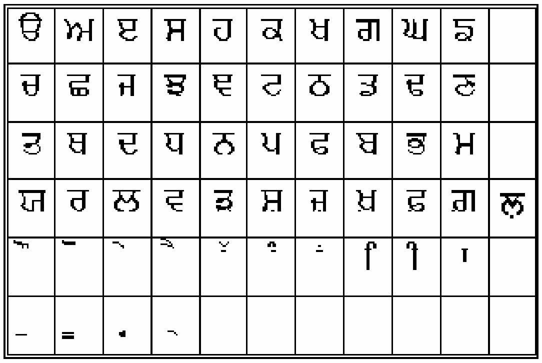
1. Pre-processing
2. Segmentation
3. Feature extraction
4. Classification
5. Post-processing

The preprocessing stage is a collection of operations that apply successive transformations on an image. It takes in a raw image, reduces noise and distortion, removes skewness and performs skeltonizing of the image thereby simplifying the processing of the rest of the stages. The segmentation stage takes in a page image and separates the different logical parts, like text from graphics, lines of a paragraph, and characters of a word. 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 selection of a stable and representative set of features is the heart of pattern recognition system design. Among the different design issues involved in building an OCR system, perhaps the most consequential one is the selection of the type and set of features. The classification stage is the main decision making stage of an OCR system and uses the features extracted in the previous stage to identify the text segment according to preset rules. The post-processing stage, which is the final stage, improves recognition by refining the decisions taken by the previous stage and recognizes words by using context. It is ultimately responsible for outputting the best solution and is often implemented as a set of techniques that rely on character frequencies, lexicons, and other context information.

The practical importance of OCR applications, as well as the interesting nature of the OCR problems have led to great research interest and tangible advances in this field. But these advances are limited to English, Chinese and Arabic languages[1-3]. Although the Indian Sub-continent languages such as Hindi and Bangla are third and fourth most widely spoken in the world, there has been very limited reported research on OCR of the scripts of these languages[2]. Some of the papers dealing with machine recognition of Indian language scripts are[4-7]. Research on the different stages of OCR of Gurmukhi script is being carried out by the authors and their M.Tech. students at Punjabi University, Patiala. A preliminary work was done by Sanjeev Kumar[8] and Khushwant Kaur[9] under the guidance of one of the authors, Lehal, developing a feature based Gurmukhi recognition script system. The count and location of local features such as endpoints, T-points, cross points and loops were used to identify isolated Gurmukhi characters A neural networks based Gurmukhi recognition system has been developed by Goyal et el[10]. Range free skew detection algorithms for de-skewing Gurmukhi machine printed text skewed at any angle, have been developed by Lehal and Madan[11] and Lehal and Dhir[12]. Segmentation techniques for machine printed Gurmukhi text have also been reported in [13-14].

To improve recognition performance, especially for handwritten and cursive scripts such as Arabic and Indian language scripts, a new trend called "Combination of multiple experts" [15] has emerged, which uses diverse feature types and combinations of classifiers arranged in layers. It is based on the idea that classifiers with different methodologies or different features can complement each other. Hence if different classifiers cooperate with each other, group decisions may reduce errors drastically and achieve a higher performance. As a result, increasingly many researchers now use combinations of the above feature types and classification techniques. A method called "Behaviour-Knowledge Space Method" has been developed by Huang and Suen[15], which can aggregate the decisions obtained from individual classifiers and derive the best final results. Almuallim and Yamagochi[16] have used three levels of feature extraction/classification to recognize handwritten words. Using the structural features of a stroke, it is mapped it into one of the five groups. Next a feature vector for the is computed and it is classified by finding its distance to a set of identification vectors. Finally, the strong of identified strokes is converted into characters by syntatic parsing. Another method is described in [3], where the tree structure is used to group the Arabic character set based on the number and location of dots and holes. This is followed by a statistical classifier to identify the characters of the same group. A hierarchical character recognition scheme is proposed by Zhou and Pavlidis[17]. Baird[18] has proposed a general technique for combining the strengths of structural shape analysis with statistical classification. A two stage classification scheme, consisting of binary tree classifier and run-based template matching approach, is employed by Chaudhuri and Pal[7] for compound character recognition of Bangla script. Heutte et al[19]have presented a new feature vector for handwritten character recognition, which combines the strengths of both statistical and structural feature extractors. A multistage scheme for the recognition of handwritten Bangla characters is introduced by Rahman and Rahman [20].

In this paper, we present a feature extraction and a mulit-stage classification scheme for machine recognition of Gurmukhi script. In the first stage of classification, the characters are grouped into three sets depending on their zonal position (upper zone set, middle zone set and lower zone). The concept of zonal positions is discussed in detail in next section. In the second stage, the characters in the middle zone set are further distributed into smaller sub-sets by a binary decision tree using a set of robust and font independent features. The final categorization of the input sample is then easily tackled by using a nearest neighbour classifier, which considers the special features and peculiarities of the characters in each subset.



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	ਸ਼	ਜ਼	ਖ਼	ਫ਼	ਗ਼	ਲ਼
ੌ	ੈ	ੇ	ੋ	੍ਹ	੍ਰ	ੰ	ਿ	ੀ	ਾ	
ੁ	ੂ	ਂ	ੱ							

Fig 1 : Character set of Gurmukhi script

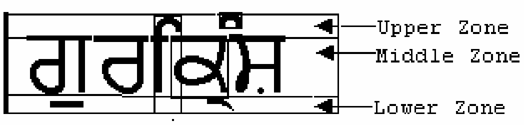


Fig 2 : Three zones of a word in Gurmukhi script with vertically overlapping and intersecting minimum bounding rectangles of characters

2. Characteristics of Gurmukhi Script

The word 'Gurmukhi' literally means from the mouth of the Guru. Gurmukhi script is used primarily for the Punjabi language, which is the world's 14th most widely spoken language. The populace speaking Punjabi is not only confined to North Indian states such as Punjab, Haryana and Delhi but is spread over all parts of the world. There is rich literature in this language in the form of scripture, books, poetry. It is, therefore, important to develop OCR for such a rich and widely used language which may find many practical use in various areas.
The inadequate research on OCR of Gurmukhi script can be attributed in part to the special characteristic of the script. Some of the major properties of the Gurmukhi script are:

- Gurmukhi script alphabet consists of 41 consonants and 12 vowels and 2 half characters which lie at the feet of consonants (Fig 1).

- From Fig.1 it can be noted that most of the characters have a horizontal line at the upper part. The characters of words are connected mostly by this line called head line and so there is no vertical inter-character gap in the letters of a word and formation of merged characters is a norm rather than an aberration in Gurmukhi script The words are, however, separated with blank spaces.
- A word in Gurmukhi script can be partitioned into three horizontal zones (Fig 2). The upper zone denotes the region above the head line, where vowels reside, while the middle zone represents the area below the head line where the consonants and some sub-parts of vowels are present. The middle zone is the busiest zone. The lower zone represents the area below middle zone where some vowels and certain half characters lie in the foot of consonants.
- The bounding boxes of 2 or more characters in a word may intersect or overlap vertically. For example in Fig 2 the bounding boxes of ਿ and ਕ intersect and the bounding boxes of ਕ੍ and ੋ overlap vertically.
- The characters in the lower zone may touch the characters in the middle zone.
- There are a lot of topologically similar characters pairs in Gurmukhi script. They can be categorized as
  i. Character pairs which after thinning or in noisy conditions appear very similar (ਟ and ਦ, ਤ and ੜ, ਬ and ਥ, ਤ and ਭ, ੇ and ੈ).

  ii. Similar looking character pairs which are only differentiated by the property if they are open/closed along the headline (ਸ and ਮ, ਧ and ਪ, ਬ and ਖ).
  iii. Character pairs which are exactly similar and distinguished only by the presence/absence of a dot in the feet of a character (ਸ and ਸ਼, ਖ and ਖ਼, ਜ and ਜ਼, ਫ and ਫ਼, ਗ and ਗ਼).

3. Feature Extraction

Extraction of good features is the main key to correctly recognize an unknown character. A good feature set contains discriminating information, which can distinguish one object from other objects. It must also be as robust as possible in order to prevent generating different feature codes for the objects in the same class. The selected set of features should be a small set whose values efficiently discriminate among patterns of different classes, but are similar for patterns within the same class. Features can be classified into two categories:

1. Local features, which are usually *geometric* (e.g. concave/convex parts, number of endpoints, branches, joints etc).
2. Global features, which are usually *topological* (connectivity, projection profiles, number of holes, etc) or *statistical* (invariant moments etc.).

After a careful analysis of the shapes of the Gurmukhi characters for different fonts and sizes, two set of features were developed.

1. Primary Feature Set
2. Secondary Feature Set

For extracting the primary features (also for secondary features), it is assumed that the character image has been thinned and the zonal information of the character image is known. Thinning is a process, which reduces the thickness of all the strokes and lines of the image to one pixel. Thinning is an efficient method for expressing structural relationships in characters as it reduces space and processing time. Zonal information is concerned with the information about the zone (upper, middle or lower) in which the character lies. As, for example, the character ਕ lies in the middle zone while the character ੋ is present in the upper zone.

3.1 Primary Feature Set

The first feature set called primary feature set is made up of robust, font and size invariant features. The purpose of primary feature set is to precisely divide the set of the characters lying in middle zone into smaller subsets which can be easily managed. The cardinality of these subsets varies from 1 to 8. For classifying the character set into smaller subclasses, a binary classifier tree is used which makes use of one feature of primary feature set at each node. The feature codes of the primary feature set have following common characteristics:

1. Less sensitive to character size and font
2. High separability: the feature codes present a very high separability for different characters. In other words, the feature codes representing different characters have a very low probability to coincide.
3. Tolerence to noise

The features used in *Primary Feature Set* are :

1. **Number of junctions with the headline (P_1)** : It can be noted that each character in Gurmukhi has either 1 or more than 1 junctions with the headline. For example ਫ has one junction while ਪ has 2 junctions. This feature has been used to divide the complete Gurmukhi character set into almost 2 equal sized subsets.
2. **Presence of sidebar (P_2)** : The presence or absence of sidebar is another very robust feature for classifying the characters. For example ਮ, ਯ and ਰ have a sidebar while ਕ, ਙ and ਛ do not have it.
3. **Presence of a loop (P_3) :** The presence of a loop in the character is another important classification feature. One thing to be noted is that we consider a loop only if headline is not part of that loop. For example ਯ does not have a loop since headline is involved while ਰ has a loop.
4. **No Loop formed with headline(P_4)** : This feature is true if the character is open at top along the headline or in other words if there is no loop containing headline as its subpart. Examples of characters with this feature are ਰ and ਖ while it is absent in ਲ and ਯ.

3.2 Secondary Feature Set

The second feature set, called secondary feature set, is a combination of local and global features, which are aimed to capture the geometrical and topological features of the characters and efficiently distinguish and identify the character from a small subset of characters. The secondary feature set is used for classification of all the characters of the Gurmukhi script lying in any one of the three zones.

This feature set consists of :

1. **Number of endpoints and their location (S_1)** : A black pixel is considered to be an end point if there is only one black pixel in its 3 x 3 neighbourhood in the resolution of the character image. In order to determine the position of an endpoint in one of the 9 quadrants, the character image is divided into a 3x3 matrix as shown in Fig. 3. Using this matrix, the position of the endpoints in terms of their positions in quadrants and their numbers are noted. For example the thinned character image of ਸ of Fig 3 has 2 endpoints in quadrants 7 and 9. The endpoints present on the headline are ignored.
2. **Number of junctions and their location (S_2)**: A black pixel is considered to be a junction if there are more than two black pixels in its 3 x 3 neighbourhood in the resolution of the character image. The number of junctions as well as their positions in terms of 9(3x3) quadrants are recorded. For example, the thinned character image of ਸ of Fig 3 has 2 junctions in quadrants 4 and 6. Junctions lying within a pre-defined radial distance are merged into a single junction and the junctions associated with the headline are ignored.
3. **Horizontal Projection Count (S_3)**: Horizontal Projection Count is represented as $HPC(i) = \sum_j F(i, j)$, where F(i,j) is a pixel value (0 for background and 1 for foreground) of a character image, and i and j denote row and column positions of a pixel, with the image's top left corner set to F(0,0). It is calculated by scanning the image row-wise and finding the sum of foreground pixels in each row (Fig. 4). 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 components symbolize the percentage of rows with 1 pixel, 2 pixels, 3 pixels and more than 3 pixels. The components of this vector for the character image given in Fig 4 will be [20, 50, 10, 20], as there are 2 rows with 1 pixel; 5 rows with 2 pixels; 1 row with 3 pixel and 2 rows with more than 3 pixels.

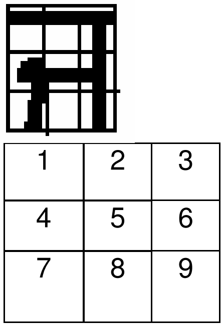


Fig 3 : Division of character image into 9 quadrants

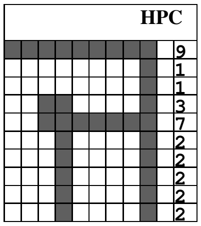


Fig 4 : Horizontal Projection Count of a character image

Left and Right Projection profiles (S_4 through S_8) : The next 5 features are based on projection profiles (Fig 5). Left projection of a character is derived by scanning each line of the character from top to bottom and from left to right, and by storing the first black pixel of the character in each row. Similarly the right projection profile is found by scanning the character from top to bottom and from right to left.

4. **Right Profile depth (S_4)**: The maximum depth of the right profile is stored as percentage with respect to total width of the box enclosing the character image. For example the right profile depth of character ਠ is 50 and right profile depth of character ਰ is 0.
5. **Left Profile Upper Depth (S_5)**: The profile is computed from the left and the maximum depth of the upper half of the profile is stored as percentage with respect to total width of the box enclosing the character image. For example the upper left profile depth of character ਠ is 50 and upper left profile depth of character ਰ is 100.
6. **Left Profile Lower Depth (S_6)**: The maximum depth of the lower half of the left profile is stored as percentage with respect to total width of the box enclosing the character image.
7. **Left and Right Profile Direction Code (S_7, S_8)**: A variation of chain encoding or Freeman code [21] is used on left and right profiles. The 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 directions 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. The direction code of the profile of fig 5(b) is [37, 18, 45] since the movements in left, downward and right direction are 4, 2 and 5 pixels respectively.
8. **Aspect Ratio (S_9) :** Aspect ratio which is obtained by dividing the sub-symbol height by its width, was found to be very useful for classifying the sub-symbols lying in lower-zone.

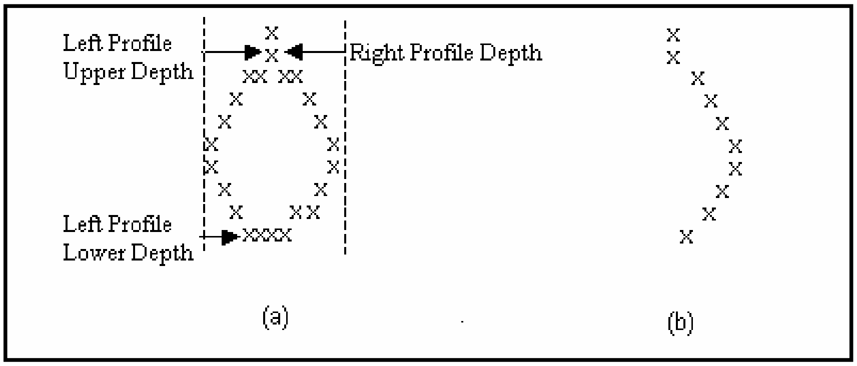


Fig 5 : Features based on Projection Profile (a) Depth of Left and Right Projection Profiles of a symbol (b) Right Projection profile of symbol in (a)

The complete feature set used for classification using nearest neighbour classifier is tabulated in table 1 .

Set #	Cardinality	Character Set	Features for classification
1	2	ਚ ਰ	S_1 S_2 S_3
2	3	ਹ ਜ ।	S_1 S_2 S_3
3	8	ਕ ਙ ਛ ਠ ਡ ਢ ਫ ਭ	S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8
4	7	ਟ ਣ ਤ ਦ ਨ ਵ ੜ	S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8
5	1	ਖ	-
6	2	ਥ ਬ	S_5 S_8
7	4	ਅ ਘ ਪ ਮ	S_1 S_2 S_3 S_5
8	3	ਸ ਧ ਯ	S_1 S_2 S_3 S_5
9	1	ੳ	-
10	4	ੲ ਝ ਞ ਲ	S_1 S_2 S_3 S_4 S_7 S_8
11	7	੍ਹ ੍ਰ ੁ ੂ ੍ ੑ ਼	S_1 S_7 S_8

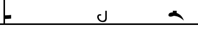
12	3		S_9

Table 1 : Secondary feature set for classification of character sets

4. Proposed Classification Scheme

The classification stage uses the features extracted in the feature extraction stage to identify the text segment according to preset rules. Classification is usually accomplished by comparing the feature vectors corresponding to the input character with the representative(s) of each character class, using a distance metric. Traditionally nearest neighbour classifier and binary classifier trees have been the two most commonly used classifiers. The nearest neighbour classifier is an effective technique[22] for classification problems in which the pattern classes exhibit a reasonably limited degree of variability. For a specific and clear machine printed text, the pattern of each class tends to cluster tightly about a typical or representative pattern for that class. Under these conditions, a nearest neighbour classifier can be a very effective approach to the classification problems. The nearest neighbour method compares the input feature vector with a library of reference vectors and the pattern is identified to be of the class of the library feature vector to which it has the closest distance. It, however, suffers from the twin problems of speed and memory. The size of the library has to continually increase as the classifier is required to recognize more and more fonts. If the classifier requires a library of 1,00,000 vectors to achieve acceptable accuracy on the training set, then 1,00,000 distances must be computed at run time to classify each input vector. Huge amount of memory is also required to store those 1,00,000 library feature vectors.

Binary decision tree classifiers were introduced some twenty five years ago and have now become an established technique in the repertoire of pattern recognition researchers [23]. Though it is particularly suited for binary type features, the method has been extended to handle features which can range over a continuous interval. A tree classifier determines the classification of a point in feature space (the input feature vector), by successively narrowing the region in which it is expected to lie. Starting from the root node, the classifier tests a particular feature or a set of features associated with that node and decides whether to branch to the next left node or the next right node. The process is continued until the classifier traces a path to a particular terminal node and returns the classification associated with that terminal node.

The binary tree classifier has the advantage of speed, since the maximum number of comparisons needed for classification of a character is equal to the height of the classifier tree, which is not more than 10 in most of the cases. Thus at the most only 10 comparisons are needed for the classification of a character using a binary classifier tree. The disadvantage of the binary classifier trees is that they are sensistive to noise and fonts and the intracable decisions based on selected fields of the feature vector are apt to be led down the wrong path if the image is noisy[23]. Once a wrong decision is made at one of the nodes, because of noise or font variation, there is no coming back and a wrong path will be followed from that point onwards and ultimately an incorrect decision will be made about the classification of that character.

Keeping in view the relative advantages and disadvantages of both the classifiers, a hybrid classification scheme which combines the relative advantages of binary tree and nearest neighbour classifiers has been used. This is expected to result in a very effectivelassification scheme.

The above proposed classification scheme for Gurmukhi characters proceeds in 3 stages. These stages are:

1. Using zonal information, we classify the symbol into one of the 3 sets, lying either in upper zone (11 elements) or in middle zone (41 elements) or in lower zone (12 elements).
2. If the symbol is in the middle zone, then we assign it to one of the sets 1-10 of table 1 using primary features and binary classifier tree. At the end of this stage the symbol has been classified into one of 12 sets including the sets for characters in upper and lower zones.
3. Lastly, the symbol classified to one of the 12 sets of table 1 is recognized using nearest neighbour classifier and the feature set of secondary features assigned for that particular set.

4.1 Design of the Binary Tree Classifier

The design of a decision tree has three components:

1. a tree skeleton or hierarchical ordering of the class labels
2. the choice of features at each non-leaf node
3. the decision rule at each non-leaf node

We have designed a strictly binary decision tree with 10 leaf and 9 non-leaf nodes. The leaf nodes correspond to the classification of the character in one of the 10 sub-classes. The height of the tree is 4. Only one feature is tested at each non-terminal node for traversing the tree. The decision rules are binary i.e. the presence/absence of the feature. The features at the non-terminal nodes are chosen according to their robustness and tolerance to noise and remain invariant under font and image size. The most stable feature is used at root node and it divides the character set into two almost equal sized subsets. The complete binary tree classier is shown in Fig. 6.

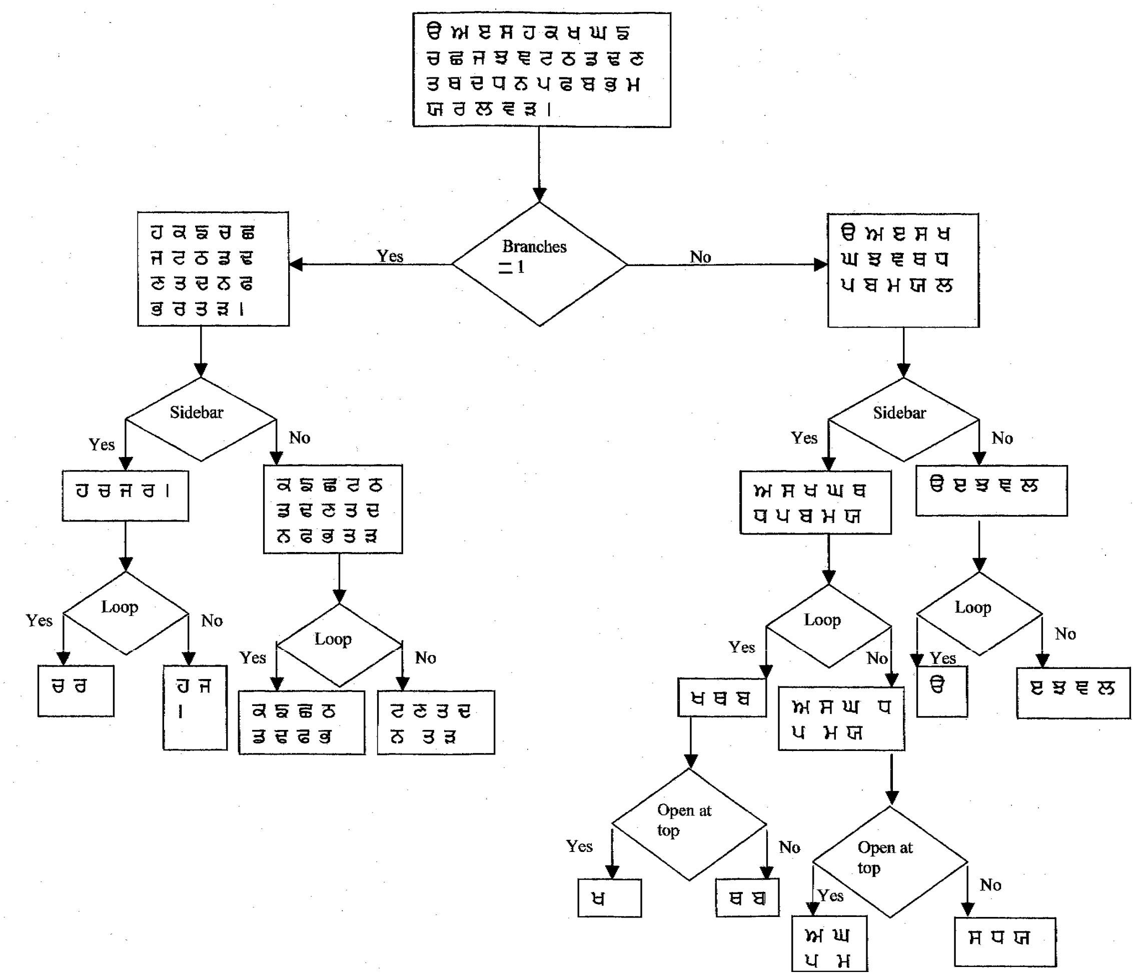


Fig 6 : Binary Classifier Tree for Gurmukhi character set

4.2 Nearest Neighbour Classifier

We have used the Euclidean distance for finding the nearest neighbour. Euclidean distance is the straight line distance between two points in an n-dimensional space. The Euclidean distance between an input feature vector X and a library feature vector C is given by

$$D = \sqrt{\sum_{i=1}^{N}(C_i - X_i)(C_i - X_i)} \qquad (1)$$

where C_i is the ith library feature and X_i is the ith input feature and N is the number of features used for classification. The class of the library feature vector producing the smallest Euclidean distance when compared with the library input feature vector is assigned to the input character. For computational efficiency, the square of the distance is considered.

It has been demonstrated by Cash and Hatamian[24], that the performance of the Euclidean distance can be improved by carrying out a statistical analysis of the training set features. Those features which are found to be more reliable than others are given more importance when making classifications. For the Euclidean distance

$$\sqrt{\sum_{i=1}^{N} w_i(C_i - X_i)(C_i - X_i)}$$

measure, weighting factors are determined which cause the more reliable features to make a larger contribution to the distance between two feature vectors. The weighted Euclidean distance between two feature vectors is given by

D = (2)

where w_i is the weighting factor for the *i*th feature and is chosen appropriately.

The number of features, N, used in equation 2 is a variable number and its value depends on the character set of X. For example, N is 3, if X has been classified as a member of set 1 by the binary classifier tree and it is 2 if X is classified as member of set 2(table 1). N has been kept as a variable, so that only the relevant features corresponding to the character set are computed and unnecessary computations are avoided. The nearest neighbour classifier in our scheme operates at subclass level and finds the similarity measure with only the prototypes of a sub-class of characters instead of prototypes of all characters. The equation 2 is thus slightly modified as follows

$$\sqrt{\sum_{i=1}^{N} w_{ij}(C_i - X_i)(C_i - X_i)}$$

D_j = (3)

where D_j is the Euclidean distance between an input feature vector X and a library feature vector C of the *j*th set of table 1 and $w_{i,j}$ is the weighting factor for *i*th feature of the *j*th set.

$$\frac{1}{\sigma ij}$$

The weights were chosen keeping in mind the fact that the feature that has a smaller variance is more reliable and should contribute more to the decision process. Thus $w_{i,j}$ is calculated as

$w_{i,j}$ = (4)

where $\sigma_{i,j}$is computed as follows:

We calculate the standard deviation for the *i*th feature over all the character prototypes of each character of set *j*.and find the mean of these standard deviations and assign it to $\sigma_{i,j}$.

As an example, we consider an unknown character X for classification. We also assume that the 4 primary features (**P_1, P_2, P_3 and P_4**) have the values {False, True, False, False} respectively. The binary classifier tree will classify X into set 8 (table 1 and Fig. 6). From table 1 it can be observed that there are 3 characters in set 8 and 4 secondary features will be used by the nearest neighbour classifier to identify X from these 3 characters. D_j will be calculated from equation 3, by assigning N=4 and j=8, over all the character templates of set 8 and the class of the template for which D is minimum will be assigned to X.

5. Experiments and Results

All the algorithms have been written in C++ and run under WINDOWS-98 Operating System on Pentium Celeron 333 Mhz system Four major fonts of Gurmukhi script have been used for training: Punjabi, Amrit-Lipi, GurmukhiLys and PN-TTAmar and four point sizes are used : 12, 16, 20 and 26. Thus 16 prototypes have been stored for each character in the training set. About 3000 Gurmukhi characters from laser print outs were scanned using an HP Scanjet P5 scanner at 300 dpi. The source documents were printed and clean. A recognition rate of 91.6% was achieved and the average processing time was 4 millisecond for each character.

6. Conclusion

In this paper, we have presented feature extraction and classification schemes for optical character recognition of Gurmukhi script. In recent years, there has been a renewed attempt to reformat the classification approaches to the recognition of difficult character sets. It has been found that a multiple classification character recognition scheme has the potential of outperforming individual stand-alone classifiers because of its ability to handle extreme variance in the training and testing samples. The present study reported in this paper follows this recent trend in building a multiple classifier character recognition configuration.

The current scheme is designed to overcome the problems encountered with decision tree and nearest neighbour classifiers and a new hybrid classifier which combines the relative strengths of these classifiers has been developed. A very small set of easy-to-compute features has been used for classification of characters. One significant point in this scheme, in contrast to the conventional single-stage classifiers where each character image is tested against all prototypes, is that a character image is tested against only certain subsets of classes at each stage to eliminate unnecessary computations. At present limited experiments have been done on the limited fonts and sizes. Work is going on to improve the accuracy and speed and extensive experiments are being performed. However, the method for classification as presented here, seems to be very effective.

7. References

[1] J. Mantas, "An overview of character recognition methodologies", *Pattern Recognition,* Vol. 19, pp 425-430 (1986).

[2] V. K. Govindan and A. P. Shivaprasad, "Character recognition – A survey ",*Pattern Recognition,* Vol. 23, pp 671-683 (1990).

[3] B. Al-Badr and S.A. Mahmoud, "Survey and bibliography of Arabic optical text recognition", *Signal Processing,* Vol. 41, pp. 49-77(1995).

[4] K. Sethi and B. Chatterjee, "Machine recognition of constrained hand printed Devanagari", *Pattern Recognition,* Vol. 9, pp. 69-75(1977).

[5] R. Chandrasekaran, M. Chandrasekaran and G. Siromony, "Recognition of Tamil, Malayalam and Devanagri characters", *J. Inst. Electron. Telecom. Engg. (India),* Vol. 30, pp. 150-154 (1984).

[6] R. M. K. Sinha, "Rule based contextual post processing for Devnagari text recognition", *Pattern Recognition,* Vol. 20, pp. 475-485(1985).

[7] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", *Pattern Recognition,* Vol. 31, pp 531-549 (1998).

[8] S. Kumar, "A technique for recognition of printed text in Gurmukhi script", M.Tech. thesis, Punjabi University, (1997).

[9] K. Kaur, "An approach towards the recognition of machine printed Gurmukhi script", M.Tech. thesis, Punjabi University, (1999).

[10] A K Goyal, G S Lehal and J Behal, "Machine Printed Gurmukhi Script Character Recognition Using Neural Networks", Accepted for publication in Proceedings 5th International Conference on Cognitive Systems, Delhi, India, (1999)

[11] G S Lehal and S. Madan, "A New Approach to Skew detection and Correction of Machine Printed Gurmukhi Script", Proceedings 2nd International Conference on Knowledge Based Computer Systems, Mumbai, India, 215-224 (1998)

[12] G S Lehal and R. Dhir, "A Range Free Skew Detection Technique for Digitized Gurmukhi Script Documents" , In Proceedings 5th International Conference of Document Analysis and Recognition, IEEE Computer Society Press, California, pp. 147-152, (1999)

[13] G S Lehal and P. Singh, "A Technique for Segmentation of Machine Printed Gurmukhi Script", Proceedings 4th International Conference on Cognitive Systems, Delhi, India, 283-287 (1998)

[14] A. K. Goyal, G S Lehal and S S Deol, "Segmentation of Machine Printed Gurmukhi Script" , Proceedings 9th International Graphonomics Society Conference, Singapore, pp. 293-297 (1999)

[15] Y. S. Huang and C.Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals", IEEE Trans. Pattern Analysis Mach. Intelligence, Vol 17, No.1, 1995, pp. 90-93.

[16] H. Almuallim and S. Yamagochi, "A method of recognition of Arabic cursive handwriting", Pattern Recognition, Vol 9, 1987, pp. 715-722 .

[17] J. Zhou and T. Pavlidis, "Discrimination of characters by a multi-stage recognition process", Pattern Recognition, Vol. 27, 1994, pp 1539-1549.

[18] H. S. Baird, "Feature identification for hybrid structural/statistical pattern classification", Computer Vision, Graphics, and Image Processing, 42, 1988, pp. 318-333.

[19] L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier and C. Olivier, "A structural/statistical feature based vector for handwritten character recognition", Pattern Recognition Letters, Vol. 19, 1998, pp. 629-641.

[20] A.F.R. Rahman and R. Rahman, "Recognition of handwritten Bengali characters : a novel multistage approach" In Proc. of 9th Biennial Conference of International Graphonomics Society, Singapore, 1999, pp. 299-304 .

[21] H. Freeman, "Boundary encoding and processing", Picture Processing and Psycholopictorics, B. S. Lipkin and A. Rosenfeld, Eds.. New York Academic Press. 1970, pp. 210-221.

[22] F. Kimura and M. Shridhar, "Handwritten numerical recognition based on multiple algorithms", Pattern Recognition, Vol. 24, 1991, pp. 969-983.

[23] S. Shlien, "Nonparametric classification using matched binary decision trees", Pattern Recognition Letters, Vol. 13, 1992, pp. 83-87.

[24] G. L. Cash and M. Hatamian, "Optical Character Recognition by the method of moments", Computer Vision, Graphics, and Image Processing, 39, 1987, pp. 291-310.

About the Author – G. S. Lehal, born in 1963, is currently working as lecturer in Department of Computer Science & Engineering, Punjabi University, Patiala. He received his Master's degree in Mathematics from Punjab University in 1988 and M.E. degree in Computer Science from Thapar Institute of Engineering and Technology in 1995. He served at Thapar Corporate Research & Development Centre, Patiala, during 1988-1995 as Systems Analyst and Software Engineer. He has been working as lecturer in Department of Computer Science & Engineering, Punjabi University from 1995. His current research interests include optical character recognition, visualization, image processing and data compression.

About the Author – Born on December 6, 1954, Dr. Chandan Singh, received B.Sc. and M.Sc. (Mathematics) degrees in first class first from Kumaon University, Nainital, in 1975 and 1977, respectively. He received Ph.D. degree from Indian Institute of Technology, Kanpur in the year 1982. Dr. Singh was in the R& D centre of M/S Jyoti Ltd., Baroda from Feb 1982 to Oct. 1987. He then served M/S Thapar Corporate R & D Centre, Patiala, till July 1994. He joined Department of Computer Science and Engineering of Punjabi University, Patiala as Reader in July 1994, and became Professor in May 1995. Currently he is serving as Professor and Head of the Department and Dean, Faculty of Engineering and Technology.

Dr. Singh has done extensive research work in the area of finite element analysis of engineering systems. He has published more than 30 research papers in various international journals and developed many software packages for engineering and scientific applications. He has developed software packages for OMR for evaluating objective tests through scanners. He has guided many M.Tech students and has been guiding Ph.D. students in Computer Science. His current research interests include fractal image compression, optical character recognition and computer graphics.