Feature Extraction and Classification Techniques in O.C.R. Systems for Handwritten Gurmukhi Script – A Survey

Pritpal Singh*, Sumit Budhiraja**

*(Department of Electronics and Communication Engineering,
UIET, Panjab University, Chandigarh, INDIA
** (Department of Electronics and Communication Engineering,
UIET, Panjab University, Chandigarh, INDIA

ABSTRACT

Optical character recognition (OCR) is very popular research field since 1950's. A great work has been done for various scripts particularly in case of English. But in case of Indian scripts the research is limited. This paper presents an overview of the various O.C.R. systems for gurmukhi which are developed for handwritten isolated gurmukhi text. In case of printed gurmukhi text a lot of research has been done but in case of handwritten gurmukhi text very less work has been done. So handwritten gurmukhi character recognition needs more attention of researchers.

Keywords: **Gurmukhi Script, handwritten isolated text, handwritten character recognition, OCR.**

1. INTRODUCTION

Optical Recognition has been one of the most challenging research area in field of image processing in the recent years. Several research works have been done to evolve newer techniques and methods that would reduce the processing time while providing higher recognition accuracy [1]-[8]. Optical Character Recognition (OCR) is the process of converting scanned images of machine printed or handwritten text into a computer process able format. The process of optical character recognition [2] has following five stages:

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

The pre-processing stage takes a raw image then following operations are applied on it:

1. **Thresholding:** Raw image either colour or grey is converted into binary image.
2. **Noise reduction:** Various techniques like morphological operations are used to connect unconnected pixels, to remove isolated pixels, to smooth pixels boundary.
3. **Normalization:** The character segmented image is normalized to 32*32 or 64*64 matrix.

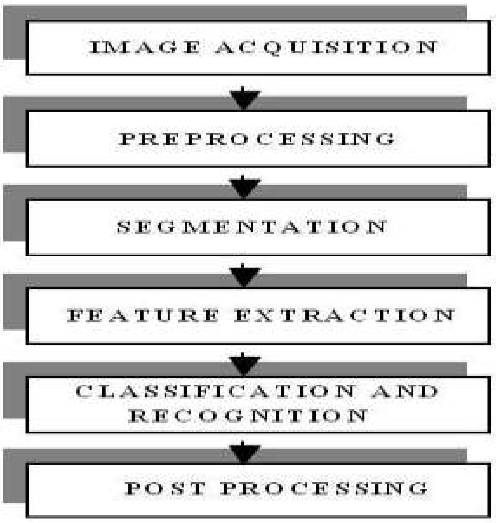


Fig. 1 Block Diagram of OCR System

The segmentation stage takes in an image and separates the different parts of an image, like text from graphics, lines of a paragraph, and characters of a word. The feature extraction stage is used to extract the most relevant information from the text image which helps us to recognize the characters in the text. The selection of a stable and representative set of features is the heart of pattern recognition system design. The classification stage uses the features extracted in the previous stage to identify the text segment according to preset rules. The post-processing stage is used to further improve recognition. Use of a dictionary for correcting the minor mistakes of the OCR systems is the simplest example of post processing.

1.1. Features Of Gurumukhi Script

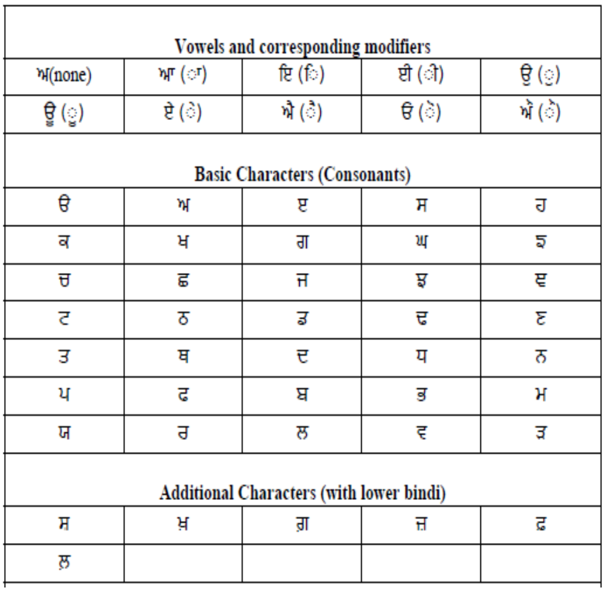
Following are the main features of gumukhi script [3] [4]:

i. Gurmukhi script consists of 35 characters, 10 vowels and modifiers, 6 additional modified consonants.

ii. Writing style is from left to right.

iii. No concept of upper or lowercase characters.

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				

1.2. THE PROBLEM OF GURUMUKHI CHARACTER RECOGNITION

Gurmukhi script OCR has some problems which are discussed below:

i. Variability of writing style, both between different writers and between separate examples from the same writer overtime.

ii. Problem is similarity of some characters.

iii. Low quality of text images.

iv. Unavoidable presence of background noise and various kinds of distortions (such as poorly written, degraded, or overlapping characters).

2. FEATURE EXTRACTION METHODS:

2.1. Zoning:

The frame containing the character is divided into several overlapping or non-overlapping zones and the densities of object pixels in each zone are calculated. Density is calculated by finding the number of object pixels in each zone and dividing it by total number of pixels [4], [5].

2.2. Projection Histogram Features:

Projection histograms count the number of pixels in specified direction [4]. There are three types of projection histograms

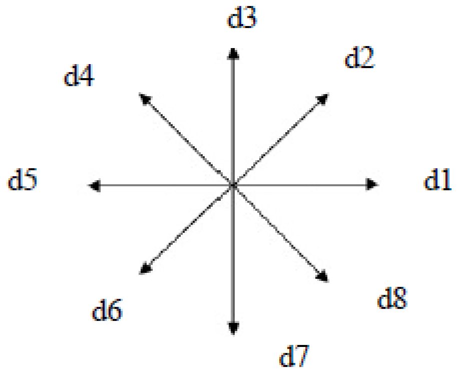
1. horizontal
2. vertical
3. Left diagonal and right diagonal.

2.3. Distance Profile Features:

Profile counts the number of pixels (distance) from bounding box of character image to outer edge of character. In this approach, profiles of four sides left, right, top and bottom were used [4].

2.4. Background Directional Distribution (BDD) 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. 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 [4].



(a)

0	0	1
0	X	2
0	0	1

d1

0	1	2
0	X	1
0	0	0

d2

1	2	1
0	X	0
0	0	0

d3

2	1	0
1	X	0
0	0	0

d4

1	0	0
2	X	0
1	0	0

d5

0	0	0
1	X	0
2	1	0

d6

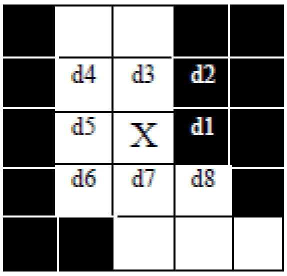
0	0	0
0	X	0
1	2	1

d7

0	0	0
0	X	1
0	1	2

d8

(b)



(c)

Fig. 2 (a) 8 directions used to compute directional distribution, (b) Masks used to compute directional distribution in different directions(c) An example of sample

2.5. Combination of various features:

Each feature is used to form a feature vector hence if we use a combination of features then it will help us to derive the feature vectors with more elements which are helpful to increase the efficiency of recognition.

3. CLASSIFICATION METHODS:

Classification determines the region of feature space in which an unknown pattern falls.

3.1. K-Nearest Neighbour:

In k-nearest neighbour algorithm (*k*-NN) [4], [5] is a method for classifying objects based on closest training examples in the feature space. The *k*-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its *k* nearest neighbours (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of its nearest neighbour. Generally we calculate the Euclidean distance between the test point and all the reference points in order to find K nearest neighbours, and then arrange the distances in ascending order and take the reference points corresponding to the k smallest Euclidean distances. A test sample is then attributed the same class label as the label of the majority of its K nearest (reference) neighbours.

3.2. SVM (Support Vector Machines):

Support vector machines (SVM) [4], [5] are a group of supervised learning methods that can be applied to classification. A classification task usually involves separating data into training and testing sets. The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. 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. Different types of kernel functions of SVM: *Linear kernel*, *Polynomial kernel*, Gaussian *Radial Basis Function* (RBF) and *Sigmoid* (*hyperbolic tangent*).

3.3. Probabilistic Neural Network (PNN) classifier:

A probabilistic neural network (PNN) [4] is a classifier which maps any input pattern to a number of classifications. If the probability density function (pdf) of each of the populations is known, then an unknown, X, belongs to class "i" if:

$f_i(X) > f_j(X)$, all $j \neq i$

f_k is the pdf for class k.

4. COMPARISON OF RESULTS AND CONCLUSION

Many researchers have proposed various techniques for handwritten gurmukhi script. The results are given below:

TABLE 2 COMPARISON OF RESULTS

S. No.	Feature Extraction Method	Classifier Used	Accuracy
1	Zoning	KNN	72.54%
2	Zoning	SVM (Poly. Kernel)	73.02%

3	Zoning density and background Directional distribution features	SVM with RBF kernel	95.04%

Zoning is used as feature extraction in [4]. Accuracy of 73.02% is obtained using SVM (Polynomial Kernel). In [5] using Zoning density and background directional distribution features and SVM with RBF kernel an accuracy of 95.04% is obtained which is the highest accuracy achieved. There is a need to extend this work to word and sentence level. There are many feature extraction techniques which are not implemented in case of handwritten gurmukhi script recognition e.g. wavelets, Fourier transform etc. So a lot of work can be done in field of Handwritten Gurmukhi Character Recognition.

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]G.S. Lehal and Chandan Singh, “A Gurmukhi Script Recognition System”, *Proceedings of the International Conference on Pattern Recognition (ICPR'00),* 1051-4651/00, 2000.

[3]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.

[4] Kartar Singh Siddharth , Mahesh Jangid, Renu Dhir, Rajneesh Rani, “Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features”, *International Journal on Computer Science and Engineering (0975-3397)*, Vol. 3 No. 6 June 2011.

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

[6] 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

[7] Rajiv Kumar and Amardeep Singh, “Hybrid Algorithm, to segment Character in Gurmukhi Handwritten Text, with a Comparative Study’’, *International Journal on Recent Trends in Engineering & Technology*, Vol. 05, No. 01, Mar 2011.