



Offline Handwritten Gurmukhi Numeral Recognition Using SVM and Different Feature Sets

Ashutosh Aggarwal* and Sukhpreet Singh
Department of Computer Science and Engineering
Dr.B.R. Ambedkar National Institute of Technology
Jalandhar- 144011, Punjab, India
er.ashutoshaggarwal@gmail.com*
sukhpreet87@gmail.com

Abstract-Isolated handwritten numerals recognition has been the subject of intensive research during last decades because it is useful in wide range of real world problems. It also provides a solution for processing large volumes of data automatically. Work has been done in recognizing handwritten characters and numerals in many languages like Chinese, Arabic, Devanagari, Urdu and English. 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. We have used three different feature sets. Out of three, two feature sets are based on the output of Gabor filters one being GABM having dimensionality 210 and other being GABN with dimensionality 200. Third feature set is comprised of Gradient Features forming 200 features. The SVM classifier with RBF (Radial Basis Function) kernel is used for classification. We have obtained the 5-fold cross validation accuracy as 99.7% using third feature set consisting of 200 gradient features. On second and first feature sets recognition rates 99.53% and 99% are observed. To obtain better results pre-processing of noise removal and normalization processes before feature extraction are recommended.

General Terms-Pattern Recognition, OCR, Indian Scripts, Gurmukhi script.

Keywords-Handwritten Gurmukhi numeral recognition, Gradient Feature Extraction, Gabor Filter, Gabor Feature Extraction, SVM classifier, RBF kernel.

I. INTRODUCTION

Recognition of handwritten characters has been a popular research area for many years because of its various application potentials, such as Postal automation, Bank cheque processing, Automatic data entry etc. There are many pieces of work towards handwritten recognition of Roman, Japanese, Chinese and Arabic scripts, and various approaches have been proposed by the researchers towards handwritten character recognition [1]. Many published paper are available towards postal automation of non-Indian language documents and sorting systems are also available for postal automation in several countries like USA, UK, Japan, Germany etc. But no such sorting system is available for Indian post. System development towards postal automation for a country like India is difficult because of its multi-lingual and multi-script behavior. India is a multi-lingual and multi-script country and not much work is done on Indian script character recognition [2]. Some pieces of work have been done on the recognition of Indian handwritten numerals [4-11] but not much work has been reported for Gurmukhi Language [12-14]. In this paper a SVM classifier based scheme is proposed towards off-line handwritten numeral recognition of Gurmukhi Language.

The features used in this paper are obtained from the directional information of the image points of the numerals. A Support Vector Machine (SVM) classifier has been used for the recognition of the numerals. In the proposed scheme, at first, the bounding box of a numeral is segmented into blocks and directional features are computed in each of these blocks. Next, these blocks are down sampled by a Gaussian filter. Finally, the features

obtained from the down sampled blocks are fed to the classifier for recognition.

An overview of the paper is as follows: In Section II, the Gurmukhi Numeral dataset which we have used in our experiment is described. Section III, covers our proposed feature extraction techniques for Gurmukhi Numeral Recognition right from pre-processing of images to use of Gradient as our Feature Extraction Technique and finally a brief introduction of SVM classifier. In Section IV, Experimental results and analysis are provided. Finally, the conclusion & acknowledgment have been offered in Section V.

II. 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. Among these samples, some distortions and irregularities are also incorporated by writers.

In pre-processing we applied many techniques like median filtration, dilation, isolated pixels removal and many other morphological operations to bridge unconnected pixels and to remove spur pixels etc. Before extracting the features we normalized the pre-processed numeral images to 32×32 pixel size.

III. FEATURE EXTRACTION

Feature extraction is an integral part of any recognition system. The aim of feature extraction is to describe the pattern by means of minimum number of features that are effective in discriminating pattern classes.

Table 1: Handwritten Samples of Gurumukhi Numerals

Digit	Samples				
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

We have used following three sets of features extracted to recognize Gurmukhi numerals.

- a) Gabor Features – GABM
- b) Gabor Features – GABN
- c) Gradient Features

A. Gabor Feature Extraction:

Gabor filters are defined by harmonic functions modulated by a Gaussian distribution. The use of the 2D Gabor filter in computer vision was introduced by Daugman in the late 1980s. Since that time it has been used in many computer vision applications including image compression, edge detection, texture analysis, object recognition and facial recognition.

Marcelja and Daugman discovered that simple cells in the visual cortex can be modeled by Gabor functions [15]. The 2D Gabor functions proposed by Daugman are local spatial band pass filters that achieve the theoretical limit for conjoint resolution of information in the 2D spatial and 2D Fourier domains.

Families of self-similar 2D Gabor wavelets have been proposed and adopted for image analysis, representation, and compression (e.g., [16]). Gabor filters have also been used extensively in various computer vision applications such as texture analysis, texture segmentation and classification, edge detection, etc. Furthermore, features extracted by using Gabor filters (we call them Gabor features) have been successfully applied to many pattern recognition applications such as face recognition, iris pattern

recognition, and fingerprint recognition. It is interesting to notice that in OCR area Gabor features have not become as popular as they have in face and iris pattern recognition areas. This situation is difficult for the new comers to understand, especially considering the following facts:

- a) Gabor features are well motivated and mathematically well-defined,
- b) They are easy to understand, fine-tune and implement,
- c) They have also been found less sensitive to noises, small range of translation, rotation, and scaling.

a. Introduction to Gabor Filter:

Gabor filters have been used extensively in image processing, texture analysis for their excellent properties: frequency and orientation representation of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination.

A Gabor Filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function.

$$h(x, y) = g(x, y) s(x, y)$$

Where $s(x,y)$ is a complex sinusoid, known as carrier and $g(x,y)$ is a Gaussian shaped function, known as envelope. The Gabor filters are self-similar, i.e. all filters can be generated from one mother wavelet by dilation and rotation. Thus the 2-D Gabor filter with the response in spatial domain is given by Eq. (1) and in spatial-frequency domain is given by Eq. (2).

Since Gaussian Function is a complex function so on convolving Gabor Filter with input image the output obtained can be used in various ways. Two of ways of manipulating the output of Gabor Filter to extract features are described below.

$$h(x, y; \lambda, \phi, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\frac{R_1^2}{\sigma_x^2} + \frac{R_2^2}{\sigma_y^2}\right]\right\} \times \exp\left[i\frac{2\pi R_1}{\lambda}\right] \quad (1)$$

Where

$$R_1 = x \cos \phi + y \sin \phi, \\ R_2 = -x \sin \phi + y \cos \phi.$$

$$h(u, v; \lambda, \phi, \sigma_x, \sigma_y)$$

$$= C \exp\left\{-2\pi^2\left(\sigma_x^2\left(F_1 - \frac{1}{\lambda}\right)^2 + \sigma_y^2(F_2)^2\right)\right\}, \quad (2)$$

Where

$$F_1 = u \cos \phi + v \sin \phi, \\ F_2 = -u \sin \phi + v \cos \phi. \\ C = const.$$

The other form of 2-D Gabor Filter in terms of frequency can be represented as:

$$h_{x,y,\theta,f} = e^{-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)} \cdot e^{i2\pi fx} \quad (3)$$

Where σ_x and σ_y explain the spatial spread and are the standard deviations of the Gaussian envelope along x and y directions. x' and y' are the x and y co-ordinates in the rotated rectangular co-ordinate system given as:

$$x' = x \cos \theta + y \sin \theta \\ y' = y \cos \theta - x \sin \theta$$

Any combination of θ and f , involves two filters, one corresponding to sine function and other corresponding to cosine function in exponential term in Eq. (3). The cosine filter, also known as the real part of the filter function, is an even-symmetric filter and acts like a low pass filter, while the sine part being odd-symmetric acts like a high pass filter.

Gabor filters having Spatial frequency ($f = 0.0625, 0.125, 0.25, 0.5, 1.0$) and orientation ($\theta = n\pi/6$) where n varies in the range 0 to 6, have been used in reported work.

b. Gabor Features-GABM:

This set of features is based on extracting features from Energy magnitudes of output of Gabor Filters.

In this the output of Gabor Features is divided into 3 parts.

- a) One part corresponds to the Real part (**Re**) of the Output,
- b) Other one corresponds to the Imaginary (**Im**) part of output,
- c) The last one corresponds to Absolute ($\sqrt{Re^2 + Im^2}$) of Complex Output of the Gaussian Gabor Filter.

After obtaining the required three forms of output, Energy magnitudes of these outputs are calculated. Energy magnitude of any output is nothing but square of that output.

In the proposed system, multi-bank Gabor filters having five different values for Spatial frequency ($f = 0.0625, 0.125, 0.25, 0.5, 1.0$) and seven different values for orientation $\theta = (0, 30, 60, 90, 120, 150, 180)$ are chosen thus giving total of 35 combinations of Gabor filters. From the output of each Gabor filter Real, Imaginary and Absolute part of output are calculated and then for each part mean (μ) and standard deviation (σ) are computed, which serves as Gabor features. Thus for each numeral image we get a feature vector of 210 values given by

$$F = \left[\begin{matrix} \mu_{(re)_1}, \sigma_{(re)_1}, \mu_{(im)_1}, \sigma_{(im)_1}, \mu_{(abs)_1}, \sigma_{(abs)_1}, \dots \\ \dots \mu_{(re)_{70}}, \sigma_{(re)_{70}}, \mu_{(im)_{70}}, \sigma_{(im)_{70}}, \mu_{(abs)_{70}}, \sigma_{(abs)_{70}} \end{matrix} \right]$$

c. Gabor Features-GABN:

This set of features is based on extracting features from real parts and imaginary parts of output of Gabor Filters.

In this also the output of Gabor Features is divided into 2 parts, Real part and Imaginary part. For this set of features we don't process the outputs further as we did in earlier technique rather we use the outputs as it is, as our feature extracted. One thing to note is that whenever the Image is convolved with Gabor Filter the size of output is similar to size of input image we have taken. Since size of image being 32×32 the output of convolution is also 32×32 thus making the feature extracted with dimensionality of 1024.

The processing time and storage increases proportionally with increase in dimensionality of feature vector. Since the size of feature is very high, the required processing time and storage can be reduced by the dimension reduction employing the principal component analysis (PCA transform). The principal component analysis is a typical dimension reduction procedure based on the orthonormal transformation which maximizes the total variances, and minimizes the mean square error due to the dimension reduction. It is shown that the dimensionality can be reduced to 1/5 without sacrificing the recognition accuracy. Thus by applying PCA we have reduced the dimensionality of feature vector from 1024 to 200. For this set of features we have to determine the optimum combination of θ & f out of the above mentioned ranges of θ and f . Along with varying values of both θ & f we also need to determine right pair of values of (σ_x, σ_y) to obtain the most suitable result as feature extracted. For our approach $\sigma_x = 7, \sigma_y = 6, \theta = \pi/6, f = 0.05$ serves as the optimum set of values.

B. Gradient Feature Extraction:

The gradient measures the magnitude and direction of the greatest change in intensity in a small neighborhood of each pixel. (In what follows, "gradient" refers to both the gradient magnitude and direction). Gradients are computed by means of the Sobel operator. The Sobel templates used to compute the horizontal (X) & vertical (Y) components of the gradient are shown in Fig.1.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Horizontal Component Vertical Component

Fig. 1: Sobel masks for gradient

Given an input image of size $D_1 \times D_2$, each pixel neighborhood is convolved with these templates to determine these X and Y components, S_x and S_y , respectively. Eq. (4) and (5) represents their mathematical representation :

$$S_x(i, j) = I(i-1, j+1) + 2 * I(i, j+1) + I(i+1, j+1) - I(i-1, j-1) - 2 * I(i, j-1) - I(i+1, j-1). \tag{4}$$

$$S_y(i, j) = I(i-1, j-1) + 2 * I(i-1, j) + I(i-1, j+1) - I(i+1, j-1) - 2 * I(i+1, j) - I(i+1, j+1) \tag{5}$$

Here, (i,j) range over the image rows (D_1) and columns (D_2), respectively. The gradient strength and direction can be computed from the gradient vector $[S_x, S_y]^T$ as shown below using Eq. (6) and (7):

The gradient magnitude is then calculated as:

$$r(i, j) = \sqrt{S_x^2(i, j) + S_y^2(i, j)} \tag{6}$$

Gradient direction is calculated as:

$$\theta(i, j) = \tan^{-1} \frac{S_y(i, j)}{S_x(i, j)} \tag{7}$$

After obtaining gradient vector of each pixel, the gradient image is decomposed into four orientation planes or eight direction planes (chaincode directions) as shown in Fig.3.

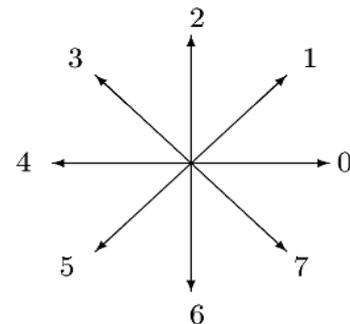


Fig. 2: 8 directions of Chaincodes

After calculation of gradient vector of each pixel, it is decomposed into components along these standard direction planes. If a gradient direction lies between two standard directions, it is decomposed into components in the two standard directions, as shown in Fig.3.

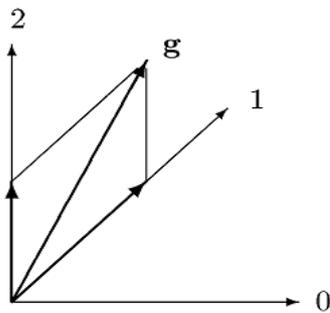


Fig.3: Decomposition of gradient direction.

a. Generation of Gradient Feature Vector:

A gradient feature vector is composed of the strength of gradient accumulated separately in different directions as described below:

- The direction of gradient detected at each image pixel as above is decomposed along 8 chaincode directions.
- The numeral image is divided into 81 (9 horizontal \times 9 vertical) blocks. The strength of the gradient is accumulated separately in each of 8 directions, in each block, to produce 81 local spectra of direction.
- The spatial resolution is reduced from 9×9 to 5×5 by down sampling every two horizontal and every two vertical blocks with 5×5 Gaussian Filter to produce a feature vector of size 200 (5 horizontal, 5 vertical, 8 directional resolution).
- The variable transformation ($y = x^{0.4}$) is applied to feature set to make the distribution of the features Gaussian-like.

The 5×5 Gaussian Filter used in this step is the high cut filter to reduce the aliasing due to the down sampling as done in paper [17].

IV. RECOGNITION RESULTS & ANALYSIS

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 [18]. 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).

In k-fold cross validation we first divide the training set into k equal subsets. Then one subset is used to test by classifier trained by other remaining k-1 subsets. By cross validation each sample of train data is predicted and it gives the percentage of correctly recognized dataset.

In our approach we have used 5-fold cross validation to obtain recognition rate.

We first tested on small samples on all possible parameters giving optimized results and then by refinement on complete dataset we finally discovered the parameters' combination giving optimum cross validation accuracy.

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

Table 2: Different Feature sets with their recognition rates

Feature Set	Recognition Rate	Parameters
Gabor Feature GABM(210)	99%	$C=4$; $\gamma = 256$
Gabor Feature GABN(200)	99.53%	$C=4$; $\gamma = 0.64-1.28$
Gradient Feature Vector(200)	99.7%	$C=4$; $\gamma = 0.002-0.005$

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

Thus we can conclude that we have obtained the maximum recognition rate as 99.7% by using Gradient as a Feature Extractor of dimensionality 200 among all three feature sets. Secondly, we obtained 99.53% recognition rate by using Gabor Feature Extractor of dimensionality 200. Thirdly, we obtained 99% recognition rate by using Gabor Feature Extractor of dimensionality 210. So from overall point of efficiency and performance, the recognition results using third feature set i.e. using Gradient features are best as it reduces computation complexity and hence time consumption while processing lesser number of features. In other way, comparing with first feature set, third features set provides significant increase in recognition rate (99% to 99.7%) while decreasing the number of features (210 to 200).

V. CONCLUSION

India is a multi-lingual and multi-script country comprising of eleven different scripts. But not much work has been done towards off-line handwriting recognition of Indian scripts. In this paper we presented an approach towards the recognition of off-line Gurmukhi handwritten numerals using multiple feature sets based on Gabor and Gradient information. We tested the proposed scheme on 2000 numeral images and obtained maximum 99.7% accuracy using Gradient feature extraction technique. The purpose of using Gradient and Gabor as mode of feature extractor is to promote its utility as major feature extraction technique in field of recognition of Indian Scripts like Gurmukhi, Malayalam etc. where not much research is conducted for their recognition. Very less literature is available on utilization of them on Indian Scripts. Both Techniques requires only a few simple arithmetic operations per image pixel which makes them suitable for real-time applications. Our approach is flexible in a way that the same algorithms can be used, without modification, for feature extraction in a variety of OCR problems. These include recognition from handwritten, machine-printed, grayscale/binary and low-resolution images. Due to its logical simplicity, ease of use and high recognition rate, more research work should be conducted in using Gradient & Gabor Features in combination with other feature extraction techniques & different-2 classifiers in order to improve recognition accuracy. More advanced classifiers as MQDF or MIL can be used and multiple classifiers can be combined to get better results.

VI. REFERENCES

- [1] R. Plamondon and S. N. Srihari, "On-Line and off-line handwritten recognition: A comprehensive survey", IEEE Trans on PAMI, vol.22, pp.62-84, 2000.
- [2] U. Pal, "Automatic Script Identification: A Survey", Vivek, vol.16, pp.26-35, 2006.
- [3] U. Pal and B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, vol. 37, pp. 1887-1899, 2004.
- [4] T. K. Bhowmick et al., "An HMM based recognition scheme for handwritten Oriya numerals", Proc.9th Int. Conf. on Information Technology, pp. 105-110, 2006.
- [5] N. Sharma, U. Pal, and F. Kimura, "Recognition of handwritten Kannada numerals", Proc.9th Int. Conf. On Information Technology, pp. 133-136, 2006.
- [6] M. Hanmandlu and O.V. Ramana Murthy, "Fuzzy model based recognition of handwritten numerals", Pattern Recognition, vol.40, pp. 1840-1854, 2007.
- [7] Y. Wen, Y. Lu and P. Shi, "Handwritten Bangla numeral recognition system and its application to postal automation", Pattern Recognition, vol.40, pp.99-107, 2007.
- [8] R. Bajaj, L. Dey, and S. Chaudhury, "Devanagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, vol.27, pp.-59-72, 2002.
- [9] S. Kumar and C. Singh, "A Study of Zernike Moments and its use in Devanagari Handwritten Character Recognition", Intl. Conf. on Cognition and Recognition, pp. 514-520, 2005.
- [10] U. Bhattacharya et al., "Neural combination of ANN and HMM for handwritten Devanagari Numeral Recognition", In Proc. 10th IWFHR, pp.613-618, 2006.
- [11] Mahesh Jangid, Kartar Singh, RenuDhir, Rajneesh Rani, "Performance Comparison of Devanagari Handwritten Numerals Recognition", International Journal of Computer Applications (IJCA), Vol. 22, No.1, May 2011.
- [12] D. Sharma, G. S. Lehal, PreetyKathuria, "Digit Extraction and Recognition from Machine Printed Gurmukhi Documents", MORC Spain, 2009.
- [13] Ubeeka Jain, D. Sharma, "Recognition of Isolated Handwritten Characters of Gurumukhi Script using Neocognitron", International Journal of Computer Applications (IJCA), Vol. 4, No. 8, 2010.
- [14] Kartar Singh, RenuDhir, Rajneesh Rani, "Handwritten Gurmukhi Numeral Recognition using Different feature sets", International Journal of Computer Applications (IJCA), Vol. 28, No.2, August 2011.
- [15] J.G. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, J. Opt. Soc. Am. A 2 (7) (1985) 1.
- [16] Daugman, J. G., IEEE Trans. On ASSP, Vol.36, No.7, pp.1169-1179, 1988. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression
- [17] Meng Shi, Yoshiharu Fujisawa, Tetsushi Wakabayashi, Fumitaka Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image" Pattern Recognition 35 (2002) 2051 – 2059.
- [18] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.