Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 1716 – 1723

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Use of Gradient Technique for extracting features from Handwritten Gurmukhi Characters and Numerals

Ashutosh Aggarwal[a,*], Karamjeet Singh[b], Kamalpreet Singh[c]

[a]*Department of Computer Science & Engg., Khalsa College of Engg. and Technology, Amritsar-143001, India.*
[b]*Department of Computer Science & Engg., Rayat & Bhara College of Engg. & Biotechnology for Women, Mohali-140104, India.*
[c]*Department of Computer Science & Engg., Lovely Professional University, Jalandhar-144411, India.*

Abstract

In this manuscript a recognition system for offline handwritten Gurmukhi characters and numerals using gradient information as mode of feature extraction technique is proposed. Two ways of extracting features using gradient information are explained in this paper. Both methods operate by accumulating gradient information from an image by dividing it into sub-images (blocks) and finally concatenating the obtained gradient features obtained from each block to form a vector of feature values with dimensionality 200. The efficiency of this feature vector is tested on two separate handwritten databases of Gurmukhi characters and Gurmukhi numerals containing 7000 & 2000 sample binary images respectively. Recognition rates of 97.38% for database of Gurmukhi characters and 99.65% for Gurmukhi numerals are obtained. Work has also been extended to test the effectiveness of the Gradient feature extraction technique on dataset of Gurmukhi characters and numerals combined together.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Handwritten Gurmukhi Script Recognition; Indian Script Recognition; Gradient Features; Pattern Recognition; OCR.

1. Introduction

India is a multilingual nation. Multiple scripts are used among different groups of people, mainly based on their geographic location. Gurmukhi script is among many of these scripts, which is very popular and widely used

* Corresponding author. Tel : +91 9988099471
E-mail address: er.ashutoshaggarwal@gmail.com.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.116

especially in the Northern region of India. So, there is an urgent need to develop a recognition system for Gurmukhi script that would ease the complexity of work to be done in government offices or in public banks in regions where the majority of the work is done in Gurmukhi. In this paper, an effort has been made to develop an automated system for recognition of handwritten Gurmukhi characters and numerals using gradient information. Various attempts have been made in past to recognize the handwritten Gurmukhi characters and numerals. Jhajj and Sharma[1] divided the image into 64 zones and combined the zoning densities calculated from each of those zones to form a feature vector for an entire image. Jain and Sharma[2] used neocognitron ANN for feature extraction and classification by computing horizontal and vertical profiles, and storing the height and width of each character under process. Siddarth et al.[3,4] used combination of zoning features, statistical features and background directional distribution (BDD) features for the recognition of Gurmukhi characters and the recognition accuracy of 95.071% was reported. Siddharth et al.[5] have proposed recognition of handwritten Gurmukhi numeral using three different feature sets. They used the features based on (first) distant profiling having 128 features, (second) projection histogram having 190 features, and (third) zonal density and background directional distribution (BDD) forming 144 features. They achieved recognition accuracy of 98%, 99.2%, and 99.13% respectively using the dataset of 1500 Gurmukhi numerals. Singh et al. have recognized Gurmukhi characters[6] and Gurmukhi numerals[7] using Gabor filters yielding recognition rates of 94.29% and 99.53% respectively. Other approaches for recognition of Gurmukhi Script can be found in[8,9,10,11,12,13,14,15,16].

In our proposed approach, gradient information of an image is used as mode of feature extraction technique. Two ways (Method 1 and Method 2) of extracting features using gradient information are presented. In both the methods, at all image pixels initially a gradient vector is computed and then the image is partitioned into 9×9 sub-images (blocks). Then in each sub-image (block) gradient strength is accumulated for each of "n" directions along which gradient direction is divided. For Method 1 "n" is 8 and for Method 2 "n" is 32. Finally, a Gaussian filter of size5×5 is applied to further downsample the image to 5×5 blocks (from 9×9 blocks), giving a feature vector of dimensionality 200 (5×5×8) in both the methods. Rest of the paper is organized as follows. Section 2 covers our proposed Gradient feature extraction technique. The experimental results are discussed in Section 3. Finally, conclusion is given in Section 4.

2. Gradient Feature Extraction

The goal of *feature extraction* is extracting information from raw data which is most suitable for classification purpose. Selection of an efficient and robust feature extraction method plays a very important role in achieving high recognition performance in character recognition systems.

During the literature survey we came across the use of directional information of character as a mode of feature extraction[17,20] and found out that their use yielded very promising accuracies that too very efficiently in terms of computing time. These drew our attention towards the use of Gradient information of sample under consideration as extracted feature and use it further for recognition purposes. This technique is successfully implemented for recognition of English, Chinese and Japanese characters as well as numerals[18,19,21] but not much literature is available towards their implementation for Indian Scripts.

2.1. Introduction to Gradient

The gradient is a vector quantity comprising of magnitude as well as directional component computed by applying its derivatives in both horizontal and vertical directions. For an image, gradient operator generates a 2D gradient vector at each image point such that it points in the direction of largest possible intensity increase, and its magnitude corresponds to the rate of change in that direction. For an image, gradient can be computed either by using Sobel operator or Robertz operator or Prewitt operator. In the proposed work we have used Sobel operator to determine **Gradient Vector** $\mathbf{[G_x,G_y]^T}$, where $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$ are the horizontal and vertical gradient components. Fig. 1 shows the horizontal and vertical masks of the Sobel operator used for calculating the horizontal & vertical components of the gradient.

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

Horizontal Component

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

Vertical Component

Fig. 1. Sobel masks

$(i-1,j-1)$	$(i-1,j)$	$(i-1,j+1)$
$(i,j-1)$	(i,j)	$(i,j+1)$
$(i+1,j-1)$	$(i+1,j)$	$(i+1,j+1)$

Fig. 2. 8-neighborhood of pixel (i,j)

Given an input image I with dimensions M and N, at each pixel (i,j), where $i = 1\ to\ M,\ j = 1\ to\ N$, an 8-pixel neighborhood is created which is then further convolved with these Sobel masks to determine $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$, respectively. The eight neighborhood of pixel (i,j) is shown in Fig. 2. Eq. (1)-(2) represents mathematical representation of $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$:

$$G_x(x,y) = I(i-1,j-1) + 2 * I(i-1,j) + I(i-1,j+1) - I(i+1,j-1) - 2 * I(i+1,j) - I(i+1,j+1) \tag{1}$$

$$G_y(x,y) = I(i-1,j-1) + 2 * I(i,j-1) + I(i+1,j-1) - I(i-1,j+1) - 2 * I(i,j+1) - I(i+1,j+1) \tag{2}$$

The Gradient Strength and Direction can be computed from the Gradient Vector $[G_x, G_y]^T$ as:

$$Gradient\ Magnitude = |G(i,j)| = \sqrt{(G_x(i,j))^2 + (G_y(i,j))^2} \tag{3}$$

$$theta(i,j) = tan^{-1}\{G_y(i,j)/G_x(i,j)\} \tag{4}$$

2.2. Calculation of Gradient Features

Gradient Feature Vector used in the proposed work is formed by accumulating the gradient strength separately along different directions. Two methods are proposed for extraction of this gradient information. Both methods generate a feature vector comprising of 200 features per image. Generation of this 200 dimensional gradient feature vector is explained below using two methods:

Method 1: Decomposition of Gradient Vector along 8 chaincode directions

STEP 1: After obtaining gradient vector of each pixel (as mentioned in section 2.1 equations (3) and (4)), the gradient image is decomposed into eight directions (chaincode directions) as shown in Fig. 3. Let E0, E1, E2, E3, E4, E5, E6 and E7 be the elementary vectors of each of 8 chaincode directions. The angle range $\{\theta | -\pi \le \theta < \pi\}$ is divided into eight equal sub-ranges as shown in Fig. 4 where $\Omega_1=\Omega_2=\Omega_3=\Omega_4=\Omega_5=\Omega_6=\Omega_7=\Omega_0=2\pi/8$.

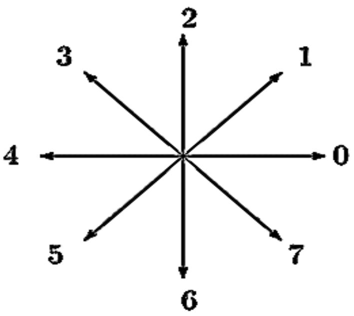


Fig. 3. 8 chaincode directions

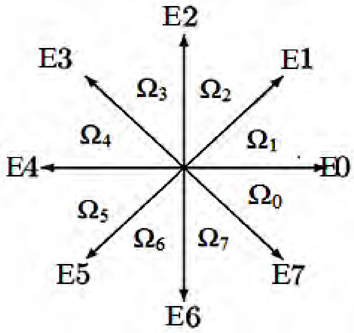
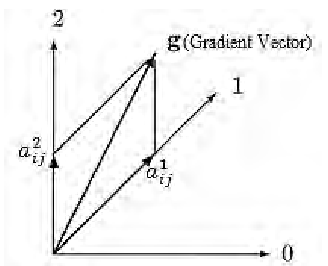


Fig. 4. 8 elementary vectors with corresponding angle areas



$$A_{ij} = \left(0, |a_{ij}^1|, |a_{ij}^2|, 0, 0, 0, 0, 0\right)$$

Fig.5.Decomposition of gradient direction.

The Directional Feature Vector of single pixel (i, j) is given by:

$$\boldsymbol{A_{ij}} = \left(|\boldsymbol{a_{ij}^0}|, |\boldsymbol{a_{ij}^1}|, |\boldsymbol{a_{ij}^2}|, |\boldsymbol{a_{ij}^3}|, |\boldsymbol{a_{ij}^4}|, |\boldsymbol{a_{ij}^5}|, |\boldsymbol{a_{ij}^6}|, |\boldsymbol{a_{ij}^7}|\right)$$

STEP 2: The Directional Feature Vector of each pixel is obtained by decomposing its Gradient Vector into components along these standard chaincode directions (shown in Fig. 3 and 4*)*. If a gradient direction lies between any two standard directions, it is decomposed into two components along those two directions resulting into generation of directional feature vector $\boldsymbol{A_{ij}}$ as shown in Fig. 5. Zeros (0) in directional feature vector $\boldsymbol{A_{ij}}$ are corresponding to directions other than those along which the gradient vector was decomposed. Following sets of equations represents the value of gradient vector g at any pixel (i, j) when it lies in between any of two elementary vectors E0, E1, E2, E3, E4, E5, E6 and E7.

$$\begin{aligned}
\boldsymbol{g(i,j)} &= a_{ij}^0 E0 + a_{ij}^1 E1,\ a_{ij}^2 = a_{ij}^3 = a_{ij}^4 = a_{ij}^5 = a_{ij}^6 = a_{ij}^7 = 0, \quad if\ \angle g(i,j) \in (0, \Omega_1) \\
&= a_{ij}^1 E1 + a_{ij}^2 E2,\ a_{ij}^0 = a_{ij}^3 = a_{ij}^4 = a_{ij}^5 = a_{ij}^6 = a_{ij}^7 = 0, \quad if\ \angle g(i,j) \in (\Omega_1, \omega_1) \quad where\ \omega_1 = \Omega_1 + \Omega_2 \\
&= a_{ij}^2 E2 + a_{ij}^3 E3,\ a_{ij}^0 = a_{ij}^1 = a_{ij}^4 = a_{ij}^5 = a_{ij}^6 = a_{ij}^7 = 0, \quad if\ \angle g(i,j) \in (\omega_1, \omega_2) \quad where\ \omega_2 = \omega_1 + \Omega_3 \\
&= a_{ij}^3 E3 + a_{ij}^4 E4,\ a_{ij}^0 = a_{ij}^1 = a_{ij}^2 = a_{ij}^5 = a_{ij}^6 = a_{ij}^7 = 0, \quad if\ \angle g(i,j) \in (\omega_2, \omega_3) \quad where\ \omega_3 = \omega_2 + \Omega_4\ \Rightarrow \omega_3 = \pi \\
&= a_{ij}^7 E7 + a_{ij}^0 E0,\ a_{ij}^2 = a_{ij}^3 = a_{ij}^4 = a_{ij}^5 = a_{ij}^6 = a_{ij}^1 = 0, \quad if\ \angle g(i,j) \in (0, -\Omega_0) \\
&= a_{ij}^6 E6 + a_{ij}^7 E7,\ a_{ij}^2 = a_{ij}^3 = a_{ij}^4 = a_{ij}^5 = a_{ij}^0 = a_{ij}^1 = 0, \quad if\ \angle g(i,j) \in (-\Omega_0, -\omega_4) \quad where\ \omega_4 = \Omega_0 + \Omega_7 \\
&= a_{ij}^4 E4 + a_{ij}^5 E5,\ a_{ij}^0 = a_{ij}^1 = a_{ij}^2 = a_{ij}^3 = a_{ij}^6 = a_{ij}^7 = 0, \quad if\ \angle g(i,j) \in (-\omega_4, -\omega_5) \quad where\ \omega_5 = \omega_4 + \Omega_6 \\
&= a_{ij}^5 E5 + a_{ij}^6 E6,\ a_{ij}^2 = a_{ij}^3 = a_{ij}^4 = a_{ij}^7 = a_{ij}^0 = a_{ij}^1 = 0, \quad if\ \angle g(i,j) \in (-\omega_5, -\omega_6) \quad where\ \omega_6 = \omega_5 + \Omega_5\ \Rightarrow \omega_6 = -\pi
\end{aligned}$$

STEP 3: The sample image is then divided into 81 blocks. The gradient magnitude is then calculated separately in each block along each of these 8 directions to produce a feature vector (with dimensionality 8) of that block. Finally all obtained 81 feature vectors corresponding to 81 blocks are merged together to form a single overall feature vector. Let $\boldsymbol{S_{rc}}$ be the square at row r and column c. Vector *F* of dimensionality 8×9×9 = 648 is defined as:

$$F = \{f_{rc}^l |\ l = 1,2,3,4,5,6,7,8;\ \ r, c = 1,2, \ldots\ldots, 9\}$$

$$f_{rc}^l = \sum_{P_{ij \in S_{rc}}} |a_{ij}^l|$$

STEP 4: In this step the image is further downsampled from 81blocks to 25 blocks using Gaussian filter of size 5 × 5, by down sampling every two horizontal and every two vertical blocks to produce a vector with 200 feature values. So, ***F*** *is* condensed into a 200-dimensional vector **GFV** (**G**radient **F**eature **V**ector) with a

Gaussian filter **W**.

$$W = \{w_{rc} \mid r, c = -2, -1, 0, 1, 2\}$$

$$GFV = \{u_{xy}^{l} \mid l = 1,2,3,4,5,6,7,8; \quad x, y = 1,2,3,4,5\}$$

$$u_{mn}^{l} = \sum_{r=-2}^{2} \sum_{c=-2}^{2} w_{rc} \, f_{2m-1+r,2n-1+c}^{l}$$

The 5 × 5 Gaussian Filter is used to counter the aliasing effects generated due to downsampling process.

STEP 5: Finally, all the obtained feature values are normalized by using a variable transformation ($y = x^{0.4}$).

This work on Gurmukhi Characters & Numerals using above explained Gradient Algorithm is inspired from the work done by[18,19,21]. So authors referring to this algorithm may refer to work done by[18,19,21] for more information on Gradient and its implementation.

Method 2: Non-Decomposition of Gradient Vector

This method is inspired from the work done by authors in[20]. This method is almost similar to method 1, the difference lies in the fact that, here instead of 8 chaincode directions 32 directions have been considered and unlike the method explained above, we haven't decomposed Gradient Direction along those directions rather the angle area in which the gradient vector lies in between two directions is considered to be part of directional feature vector of that pixel and rest all values corresponding to other 31 angle areas is taken to be 0.

Angle area corresponds to the angle between two adjacent directions. For example, α , β represent two angle areas between directions 0 & 1 and between directions 1 & 2 respectively as shown in Fig. 6. So corresponding to 32 directions i.e. from 0 to 32 we have 32 angle areas: $\{\theta_0, \theta_1, \theta_2, \dots \dots \dots \dots \dots, \theta_{31}\}$ such that $\{\theta_0 = \theta_1 = \theta_2 = \cdots = \cdots = \cdots = \theta_{31} = 2\pi/32\}$.

<u>The Directional Feature Vector of single pixel (i, j) is given by:</u>

$$\boldsymbol{A}_{ij} = \begin{pmatrix} |\boldsymbol{a}_{ij}^{0}|, |\boldsymbol{a}_{ij}^{1}|, |\boldsymbol{a}_{ij}^{2}|, |\boldsymbol{a}_{ij}^{3}|, |\boldsymbol{a}_{ij}^{4}|, |\boldsymbol{a}_{ij}^{5}|, |\boldsymbol{a}_{ij}^{6}|, |\boldsymbol{a}_{ij}^{7}|, \\ |\boldsymbol{a}_{ij}^{8}|, |\boldsymbol{a}_{ij}^{9}|, |\boldsymbol{a}_{ij}^{10}|, |\boldsymbol{a}_{ij}^{11}|, |\boldsymbol{a}_{ij}^{12}|, |\boldsymbol{a}_{ij}^{13}|, |\boldsymbol{a}_{ij}^{14}|, |\boldsymbol{a}_{ij}^{15}|, \\ |\boldsymbol{a}_{ij}^{16}|, |\boldsymbol{a}_{ij}^{17}|, |\boldsymbol{a}_{ij}^{18}|, |\boldsymbol{a}_{ij}^{19}|, |\boldsymbol{a}_{ij}^{20}|, |\boldsymbol{a}_{ij}^{21}|, |\boldsymbol{a}_{ij}^{22}|, |\boldsymbol{a}_{ij}^{23}|, \\ |\boldsymbol{a}_{ij}^{24}|, |\boldsymbol{a}_{ij}^{25}|, |\boldsymbol{a}_{ij}^{26}|, |\boldsymbol{a}_{ij}^{27}|, |\boldsymbol{a}_{ij}^{28}|, |\boldsymbol{a}_{ij}^{29}|, |\boldsymbol{a}_{ij}^{30}|, |\boldsymbol{a}_{ij}^{31}| \end{pmatrix}$$

where each $\left|\boldsymbol{a}_{ij}^{p}\right|$, $\boldsymbol{p} \in \{\boldsymbol{0}, \boldsymbol{1}, \boldsymbol{2}, \dots\dots\dots, \boldsymbol{31}\}$ can have value equal to modulus of gradient vector i.e. $|\mathbf{g}|$ (Gradient magnitude) depending upon in which angle area the Gradient Vector lies or in other terms in between which two directions the gradient vector falls. For e.g., in Fig. 7 the gradient vector g lies between direction 1 & 2 so the second component of directional vector $\boldsymbol{A}_{ij}$corresponding to angle between directions 1 & 2 would have value equal to $|\mathbf{g}|$ and rest all components of $\boldsymbol{A}_{ij}$ would be equal to 0.

Steps for calculating Gradient Feature Vector (GFV) are:

STEP 1: The gradient direction (theta) calculated in Section 2.1 are quantized to 32 levels with π/16 interval and directional feature similar to what explained above is obtained at each pixel.

STEP 2: The sample image is then divided into 81 blocks. The gradient magnitude is then calculated separately in each block along each of these 32 directions to produce a feature vector of that block. Finally 81 feature vectors corresponding to 81 blocks are obtained.

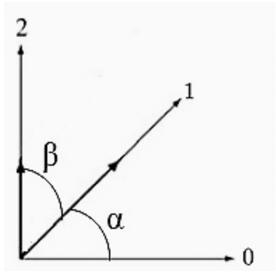


Fig. 6. Representation of angle area between two adjacent directions.

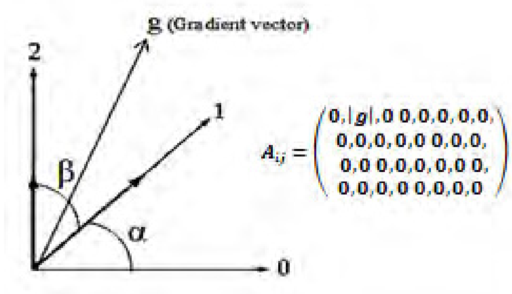


Fig. 7. Representation of directional vector

STEP 3: Similar to what is been done in Method 1, a 5×5 Gaussian filter is used to further reduce the spatial resolution from 81 blocks to 25 blocks by down sampling every two horizontal and every two vertical blocks.

STEP 4: In Method 2, not only spatial resolution but also the directional resolution is reduced from 32 to 16 directions and then from 16 to 8 directions by down sampling with a weight vectors $[1\ 4\ 6\ 4\ 1]^T$ and $[1\ 2\ 1]^T$ respectively to produce a vector with 200 feature values. Directional resolution is not performed in method 1.

STEP 5: Finally, all the obtained feature values are normalized by using a variable transformation ($y = x^{0.4}$), as done in Method 1.

The 5×5 Gaussian filter, the weight vector $[1\ 4\ 6\ 4\ 1]^T$ and $[1\ 2\ 1]^T$ used in the above steps are the high cut filters to reduce the effect of aliasing caused due to the down sampling, similar to what is been done in method 1.

3. Experiments and Results

To test the effectiveness of the proposed feature extraction technique based on gradient information, experiments were carried out on separate datasets of Gurmukhi characters and Gurmukhi numerals. The respective datasets of Gurmukhi characters and Gurmukhi numerals consists of 7000 and 2000 sample images each of size 63×63. This normalized size of sample images was empirically determined for experimental setup and for suitability with proposed feature extraction algorithm. Figure 8 shows handwritten sample images of Gurmukhi characters and numerals. SVM Classifier with RBF kernel[22,23] has been used for recognition purpose. The experimental framework uses 5-fold cross validation architecture for computing the recognition accuracy. In this framework, experimental database is divided into 5 equal subsets and testing is done on each subset using the remaining four subsets for training. The recognition rates obtained at each test subsets are averaged to calculate the overall recognition accuracy. All results are compiled in Tables 1-4 which are described below.

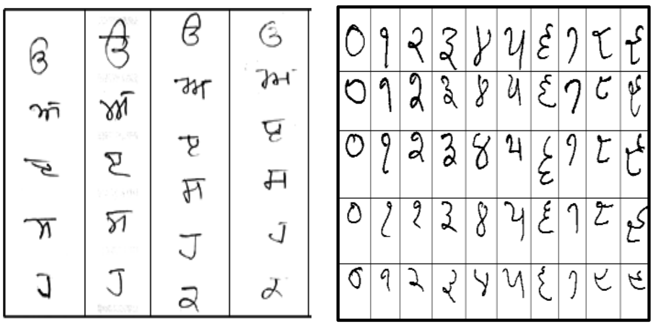


Fig. 8: Handwritten Samples of Gurmukhi Script

Table 1: Recognition Rates for Gurmukhi Characters and Numerals

Gradient Feature Vector	**Method 1**	**Method 2**
Recognition Rate for Gurmukhi Characters	97.04%	97.38%
Recognition Rate for Gurmukhi Numerals	99.60%	99.65%
Recognition Rate of Gurmukhi Characters and Numerals Combined together	97.80%	98.05%
Dimensionality of features	200	200

Table 2: Comparison with other existing state-of-art-approaches for Gurmukhi Numerals

Method	**Accuracy (%)**
Dharamveer Sharma, et al.	92.6
Ubeeka Jain et al.	92.78
Kartar Singh Siddharth, et al. (projection histogram)	99.13
Kartar Singh Siddharth, et al. (zonal density and BBD)	99.2
Sukhpreet Singh et al. (Gabor Filters)	99.53
Proposed Approach (Method 2)	**99.65**

Table 3. Comparison with other existing state-of-art-approaches for Gurmukhi characters

Proposed By	***Features Used***	***Classifier Used***	***Accuracy Obtained***
Naveen Garg et al.	Structural Features	Neural Network	*83.32%*
Anuj Sharma et al.	Strokes recognition and matching	Elastic matching	*90.08%*
Anuj Sharma et al.	Small line segments	Elastic matching	*94.59%*
Anuj Sharma et al.	HMM elements	Hidden Markov model	*91.59%*
Puneet Jhajj et al.	Zoning density (64)	SVM with RBF kernel	*73.83%*
Ubeeka Jain et al.	Profiles, width, height, aspect ratio	Neocognitron Neural Network	*92.78%*
Sukhpreet Singh et al.	Gabor Filters	SVM with RBF Kernel	*94.29%*
Kartar Siddarth et al.	Zoning density and BDD features	SVM with RBF kernel	*95.04%*
Proposed approach	**Gradient Features**	**SVM with RBF kernel**	***97.38%***

Also work was extended to recognize Gurmukhi numerals and characters combined in same dataset. The reason behind this work is that, in our day to day life while reading some book or newspaper we often encounter such documents which contains text containing both characters and numerals together. So the recognition system should be equipped with functionality to differentiate not only among various characters or numerals but also among characters and numerals when combined together. In this case too, testing was done on separate dataset containing Gurmukhi characters and numerals combined together. The combined dataset of Gurmukhi numerals and characters consists of 9000 (7000 + 2000) sample images which are formed by merging the Gurmukhi Character dataset with Gurmukhi Numeral dataset into a single dataset. The recognition rates of 98.055% and 97.8% were obtained using Method 1 and Method 2 respectively, as shown in Table 1.

4. Conclusion

Character Recognition field is vastly emerging field which is attracting researchers around the globe to contribute and come up with new ideas and innovations to create such type of systems that can give accuracies up in the range of 99-100% for all type of documents whether it is printed or handwritten or whether it is written in isolated manner or in continuous form. In this proposed work, recognition of handwritten characters and numerals of Gurmukhi script has been performed using gradient information and an effort is made to improve the already reported accuracy for Gurmukhi language. Gradient is a very effective and efficient feature extractor as it requires only a few simple operations per pixel, making it appropriate for real-time recognition applications. In this paper two methods (Method 1 and Method 2) are proposed for extracting gradient features. The effectiveness of both the methods is tested on two separate databases of Gurmukhi characters and numerals. The first method (Method 1) is found to be computationally faster and efficient than the other one (Method 2). From the experimental results it can be seen that the proposed methods yield very high and promising results both for Gurmukhi characters and numerals.

References

1. Jhajj P, Sharma D. Recognition of Isolated Handwritten Characters in Gurmukhi Script. *Int J Comput Appl* 2010; **4**: 9-17.
2. Jain U, Sharma D. Recognition of Isolated Handwritten Characters of Gurmukhi Script using Neocognitron. *Int J Comput Appl* 2010; **10**: 10-16.
3. Siddharth KS, Dhir R, Rani R. Handwritten Gurmukhi Character Recognition Using Zoning Density and Background Directional Distribution Features. *Int J Comput Sci Inf Technol* 2011; **2**: 1036-1041.
4. Siddharth KS, Jangid M, Dhir R, Rani R. Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features. *Int J Comp Sci Eng* 2011; **3**: 2332-2345.
5. Siddharth KS, Dhir R, Rani R. Handwritten Gurmukhi Numeral Recognition using Different Feature sets. *Int J Comput Appl* 2011; **28**: 20-24.
6. Singh S, Aggarwal A, Dhir R. Use of Gabor Filters for recognition of Handwritten Gurmukhi Characters. *Int J Adv Res Comput Sci Softw Eng* 2012; 2: 234-240.
7. Singh S, Dhir R. Recognition of Handwritten Gurmukhi Numerals using Gabor Filters. *Int J Comput Appl* 2012; 4**7**: 7-11.
8. Pal U, Sharma N, Jayadevan R. Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques. *ACM Transactions on Asian Language Information Processing* 2012; **11**: 1.
9. Pal U, Chaudhury BB. Indian Script Character Recognition: A Survey. *Pattern Recogn* 2004; **37**: 1887-1899.
10. Lehal GS, Singh C. A Gurmukhi Script Recognition System. In: *Proceedings of 15th International Conference on Pattern Recognition* 2000; pp. 557-560.
11. Lehal GS, Singh C. A Complete Machine printed Gurmukhi OCR. *Vivek* 2006; **16**: 10.
12. Lehal GS, Singh C. Feature Extraction and Classification for OCR of Gurmukhi Script. *Vivek* 1999; **12**: 2-12.
13. Lehal GS, Singh C. A Post Processor for Gurmukhi OCR. *Sadhana* 2002; **27**: 99-111.
14. Goyal V, Lehal GS. Comparative Study of Hindi and Punjabi Language Scripts. *Nepalese Linguistics* 2008; **23**: 67-82.
15. Sharma A, Kumar R, Sharma RK. Online Handwritten Gurmukhi Character Recognition Using Elastic Matching. In: *IEEE Congress on Image and Signal Processing* 2008; pp. 391-396.
16. Sharma D, Lehal GS, Kathuria P. Digit Extraction and Recognition from Machine Printed Gurmukhi Documents. In: *Proceedings of the International Workshop on Multilingual* 2009.
17. Kawamura A, Yura K, Hayama T, Hidai Y, Minamikawa T, Tanaka A, Masuda S. On-line Recognition of Freely Handwritten Japanese Characters Using Directional Feature Densities. In: *Proceedings of 11th IAPR International Conference on Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems* 1992; pp. 183-186.
18. Srikantan G, Lam SW, Srihari SN. Gradient-Based Contour Encoding For Character Recognition. *Pattern Recogn* 1996; **29**: 1147-1160.
19. Shi M, Fujisawa Y, Wakabayashi T, Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. *Pattern Recogn* 2002; **35**: 2051-2059.
20. Bai ZL, Huo Q. A Study on the Use of 8-Directional Features for Online Handwritten Chinese Character Recognition. In: *Proceedings of 8th IEEE International Conference on Document Analysis and Recognition* 2005; pp. 262-266.
21. Liu CL, Nakashima K, Sako H, Fujisawa H. Handwritten digit recognition: investigation of normalization and feature extraction techniques. *Pattern Recogn* 2004; **37**: 265 – 279.
22. Hsu CW, Chang CC, Lin CJ. A Practical Guide to Support Vector Classification. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
23. Chang CC, Lin CJ. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.