Punjabi Handwritten Character Recognition Using Wavelet Based Features

Avneet Kaur[1], Sanya Malhotra[2]
[1]M. Tech., Computer Science & Applications, Thapar University, Patiala, Punjab 147004, India. Email: avneetgalhotra@gmail.com
[2]M. Tech., Computer Science, IIIT Delhi, Delhi 110020, India. Email: sanya13105@iiitd.ac.in

Abstract: In today's era, everyone is busy to the fullest and with increasing digitization everyone wants one's work to be completed easily and as fast as possible. If it was possible to give input to a digital machine in our own handwriting and in our own native language, it would have been easier and probably less time consuming. This paper thus focuses on Handwritten Character Recognition [7] of Punjabi Characters and Numerals. The steps [1] involved in this process are capturing the handwritten character or numeral given as input in the form of an image, preprocessing the image, extracting the features from the image and finally, the classification or recognition of the character or numeral captured. Feature extraction is a very important step in this process as with rapid growth and development in the field of Character Recognition, a number of methods have been proposed for Feature Extraction process [2], e. g. goemetric moments, color histogram, zoning, principal component analysis, etc. This paper focuses particularly on using wavelet coefficients as features that are achieved by wavelet decomposition [6] of the handwritten character's or numeral's image.

Keywords: wavelet coefficients; feature extraction through wavelet decomposition; handwritten character recognition; wavelet based features; SVM classifier.

I. INTRODUCTION

Handwritten Character Recognition [3], as the name suggests, refers to recognizing the input characters or numerals that are written by the user instead of typing it in. It is similar to the Optical Character Recognition [5] where instead of recognizing the printed characters, the handwritten characters are recognized. The research work presented in this paper was done on the Punjabi language. For this, inputs were collected from various users in their handwritings using digital input. The Punjabi Characters and Numerals taken as inputs were then converted to images. These images were then preprocessed to bring them to a standard size. Three different sizes have been tested in this work, i.e., 32 × 32, 64 × 64 and 128 × 128. The features were extracted through wavelet decomposition [6] of the preprocessed images. The features thus obtained are the wavelet coefficients. These coefficients have been used to train the SVM classifier [9] to get the accuracy of the overall recognition system, i.e., percentage of correct classification or recognition of the character or numeral given as input. Complete methodology of the system and its overall performance and effectiveness has been demonstrated with the help of suitable examples.

II. COLLECTION OF THE DATA

The handwritten character or numeral that is taken as an input from a user is to be represented in digital format so that the machine or the software can recognize it. In the work presented, the character or numeral taken as input was captured as a digital image of size 1000 × 1000. 51 Punjabi Characters and 40 Punjabi Numerals were taken as input from 178 different users using digital input. The inputs taken were divided into Punjabi Characters and Punjabi Numerals. Punjabi Characters were further divided into 3 categories, namely, Upper Zone Characters, Middle Zone Characters and Lower Zone Characters as shown in Figure 1.

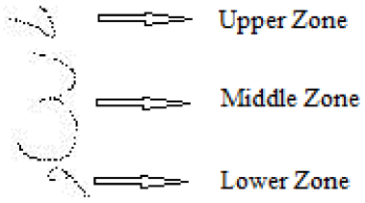

Figure 1: Zones of the Punjabi Characters

100 samples per character and 22 samples per numeral were collected. Number of characters and numerals tested from each category are shown in Table 1.

Table 1 Numbers of characters tested from each category

Category	Number of different classes tested	Total samples developed before pre-processing
Upper Zone Characters	11	1,100
Middle Zone Characters	33	3,300
Lower Zone Characters	7	700
Numerals	40	880

III. PRE-PROCESSING

As the characters and numerals written by different users were of different width and height, the images capturing them were thus pre-processed to bring them to a standard size. Pre-processing was done of every image representing any zone's Character or the Numeral. To pre-process the image was firstly windowed and then it was normalized to three different sizes, i.e., 32 × 32, 64 × 64 and 128 × 128, giving three different images. Thus, thrice the number of total samples discussed above, were developed for each category after pre-processing.

IV. EXTRACTING THE FEATURES

Feature extraction [4] involves highlighting the important information in the image representing the handwritten character or numeral that differentiates it from other characters and numerals. It thus discards the redundant and not useful information represented by it, thus, reducing the feature set that is built to describe a character uniquely in order to recognize it.

In the work presented, the features were extracted from the image representing character or numeral by Discrete Wavelet Decomposition of the image. The Discrete Wavelet Transform has been described in detail by Polikar [8]. The Wavelet Coefficients achieved afterwards were used as features in the Feature Set. The levels of decomposition and the number of wavelet coefficients achieved after applying wavelet decomposition on the character/numeral image are described in Table 2.

Table 2 Levels of decomposition used

Size of the normalized image	Number of levels of decomposition used	Number of Coefficients achieved respectively
32 × 32	2	256, 64
64 × 64	2	1,024, 256
128 × 128	3	4,096, 1,024, 256

The wavelet coefficients can be achieved by discrete wavelet decomposition of any normalized image by using any one of the wavelets [11] mentioned in Table 3. This was practically accomplished using Matlab.

Table 3 Wavelets used for decomposition

Wavelet family	Wavelet keyword	Wavelets belonging to the family
Bior Splines	bior	bior1.1, bior1.3, bior1.5, bior2.2, bior2.4, bior2.6, bior2.8, bior3.1, bior3.3, bior3.5, bior3.7, bior3.9, bior4.4, bior5.5, bior6.8
Coiflets	coif	coif1, coif2, coif3, coif4, coif5
Daubechies	db	db1, db2, db3, db4, db5, db6, db7, db8, db9, db10
DMeyer	dmey	Dmey
Haar	haar	Haar
Symlets	sym	sym2, sym3, sym4, sym5, sym6, sym7, sym8
Reverse Bior	rbio	rbio1.1, rbio1.3, rbio1.5, rbio2.2, rbio2.4, rbio2.6, rbio2.8, rbio3.1, rbio3.3, rbio3.5, rbio3.7, rbio3.9, rbio4.4, rbio5.5, rbio6.8

Thus, from every windowed image, 3 resized or normalized images were developed and from them 7 feature sets were developed as described in Table 3 using any one of the 54 wavelets described in Table 3. Similarly, from every windowed image, 7 feature sets each were developed using other wavelets as shown in Figure 2.

Thus in total 378 (54 × 7) feature sets were developed from each windowed image representing a Punjabi character or numeral. In all 6,781,320 feature sets were developed from three resized images of all the 100 samples of each class of characters from every zone and 22 samples of each class of numerals.

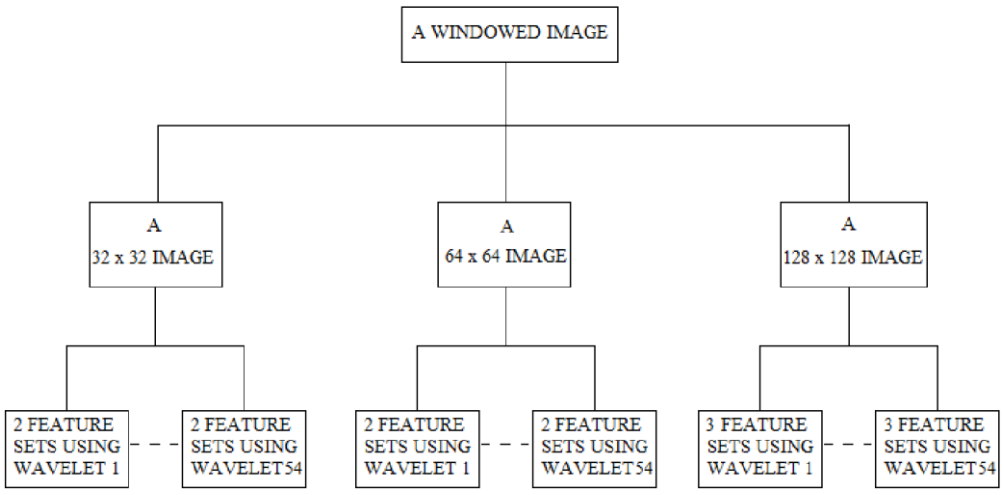

Figure 2: Feature Sets developed

V. RECOGNIZING THE CHARACTER OR NUMERAL

In order to recognize a character or numeral, every character from any zone and every numeral was given an ID that represented its class in order to differentiate it from other characters and numerals. Recognition involves recognizing a particular character/numeral written by the user that is captured as an image, by classifying the feature set obtained from the image (developed from the normalized image by wavelet decomposition of the image as discussed before). Classification of the feature set involves prediction of the class of the character/numeral to which it may belong depending on its feature set where a class is any ID from its zone if it is a character else any ID from numerals.

As discussed before, 378 different feature sets can be developed from a single character or numeral written. In order to find out the feature set that gives the maximum probability of character/numeral being recognized correctly, the classifier must be trained first with a number of feature sets of the

same character/numeral obtained from some of the samples of that character/numeral so that it classifies the feature set provided from the same ID's sample to the same ID. Afterwards, testing of the remaining samples of every ID was done to find out the accuracy of the recognition process. This training and testing was done in four steps, i.e., training of characters from upper zone, training of characters from middle zone, training of characters from lower zone and training of numerals. The classifier used was lib-svm [10]. A detailed description about SVM Classifiers is given by Thorsten [9].

For recognition, out of 100 samples of each ID of a particular size from any zone of characters, 80 samples were used for training which gave 80 feature sets using any one wavelet and they were kept in same training file and the rest corresponding 20 samples were used for testing from which the feature sets obtained using same wavelet were kept in one testing file corresponding to the training file obtained above. For the numerals out of 22 samples, 16 were used for training purposes and the rest 8 were used for testing. Similarly, for every 54 wavelets applied, 54 different training and 54 corresponding testing files were obtained.

But instead of keeping every ID's testing and training feature sets in different files, testing and training feature sets of all IDs from same category having same number of coefficients (same level of decomposition of same sized image) that were obtained using same wavelet were kept in the same testing and training files. Similarly 53 other training and testing files were obtained from all the samples of every ID. Thus, in total 378 (54 × 7) training and 378 testing files were obtained from one category of IDs.

Equation (1), (2) and (3) represents the total number of training and testing files and the number of feature sets per training/testing file.

$$\text{Total testing/training files} = \sum_{i=1}^{\text{number of noramlizations}} \text{number of decomposition levels} \times 54 \tag{1}$$

$$\text{Feature sets per testing file} = \text{number of IDs in zone/number of numerals} \times 20/8 \tag{2}$$

$$\text{Feature sets per training file} = \text{number of IDs in zone/number of numerals} \times 80/16 \tag{3}$$

Therefore, the number of feature sets per testing file = 11 × 20 = 220 and the number of feature sets per training file = 11 × 80 = 880 for the upper zone characters. Similarly, the number of feature sets per testing file = 33 × 20 = 660, 7 × 20 = 140 and 40 × 8 = 320 for the middle zone characters, lower zone characters and numerals respectively and the number of feature sets per training file = 33 × 80 = 2,640, 7 × 80 = 560 and 40 × 16 = 640 sets for the middle zone characters, lower zone characters and numerals respectively.

The whole recognition process involved four steps [10]. Figure 3 represents the process of recognition using the testing and training files developed above. One training file and its corresponding testing file are fed to the classifier at one time. The lib-svm classifier uses a special format for the training and testing files. The feature sets were converted to this format before giving them to the classifier for training and testing as the lib-svm only understands this format.

The format goes like,

ID 1:1st coefficient 2:2nd coefficient … *n*:*n*th coefficient (a line break)

ID 1:1st coefficient 2:2nd coefficient ... *n*:*n*th coefficient (a line break)

.

.

ID 1:1st coefficient 2:2nd coefficient … *n*:*n*th coefficient

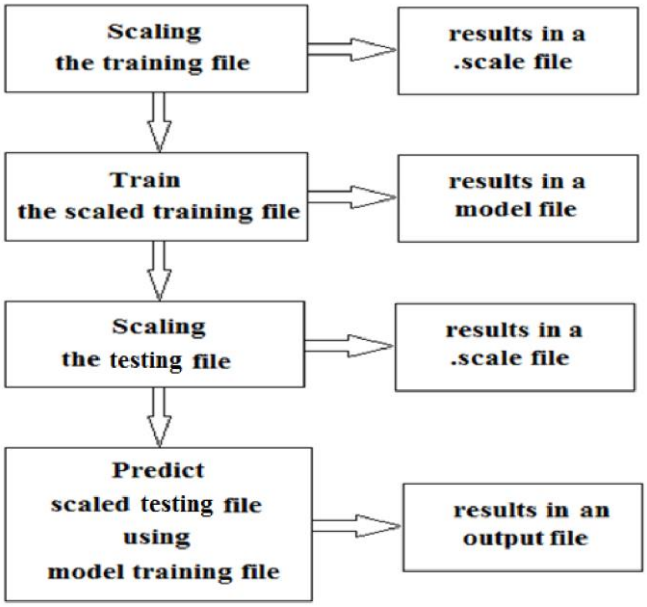

Figure 3: The recognition process

Scaling:

The scaling was done using the same parameters [10] for all the testing or training files used. The parameters taken were,

-l 1 -u 9

where 'l' represents the minimum value for scaling and 'u' represents the maximum limit for scaling.

Training:

Training was done twice for all the IDs of one category having same size of images and same number of coefficients.

Firstly the general parameters that are described below were taken for training of all the 54 training files with same number of coefficients obtained from same sized images that gave 54 model files correspondingly.

-s 0 –t 2 –d 3 –g 1 –r 0 –c 1

Then out of the 54 output files obtained using these 54 model files correspondingly, wavelet that corresponds to the maximum accuracy achieved for that number of coefficients was found and then all combinations were tried for the various parameters of svm-train on that particular wavelet training-testing file.

Various parameters combinations tried are shown in Table 4 and 5.

Table 4 Parameters tried for training

Parameter name	Parameter description	Parameter symbol	Values tried
svm type	type of the support vector machine	*-s*	0, 1, 2, 3, 4
cost	value for the parameter C for C-SVC, epsilon-SVR, and nu-SVR classifiers	*-t*	0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000
kernel type	type of the kernel	*-c*	0, 1, 2
nu	value for the parameter nu for nu-SVC, one-class SVM, and nu-SVR classifiers	*-n*	0.1, 0.2, 0.3,…0.9
epsilon	tolerance value for the termination criterion	*-e*	0.01, 0.05, 0.1, 0.5, 1
epsilon	epsilon value for the loss function of epsilon-SVR	*-p*	0.1
gamma	gamma value for the kernel function	*-g*	0, 0.01, 0.05, 0.1, 0.5, 1
coef0	coef0 value for the kernel function	*-r*	0, 1, 5, 10, 50, 75, 100
degree	degree value for the kernel function	*-d*	0, 1, 2, 3
cross validation	k-fold cross validation	*-v*	1, 5, 10, 20, 50, 80

Table 5 SVM type and kernel type parameters in detail

Parameter name	Parameter description	Value	Description
type of support vector machine	*-s*	0	C-SVC
		1	*nu*-SVC
		2	One class SVM
		3	Epsilon SVR
		4	*nu*-SVR
kernel type	*-t*	0	linear kernel = u × v
		1	polynomial = $(\text{gamma} \times u \times v + \text{coef0})^{\text{degree}}$
		2	radial basis = $e^{(-\text{gamma} \times \lvert u-v\rvert \hat{} 2)}$
		3	sigmoid = tanh(gamma × u × v + coef0)

All the parameters with all the possible combinations were not tried together. There are combinations of parameters that were tried together excluding few other parameters at that time. Table 6 represents the various parameters that were tested together for various combinations.

Table 6 Various combinations tried together

Parameter	Value	Parameters used along with it
-s	0	*c*
	1	*n*
	2	*n*
	3	*c, e, p*
	4	*c, n*
-t	0	*v*
	1	*g, r, d, v*
	2	*g, v*
	3	*g, r, v*

Testing:

The testing step takes as input a model training file and a scaled testing file and gives two output files, one gives the output analysing which it was concluded that which feature set was predicted with correct ID and which was not and the other file tells us the accuracy achieved in this process with that particular model file that was achieved with particular values for various parameters used.

VI. RECOGNIZING THE CHARACTER OR NUMERAL

In order to find out the best accuracy, every 54 files were first tested with the general parameters in order to find out the wavelet that gives the maximum accuracy for that size and that number of coefficients combination.

In all, for any zone or for numerals 2 (for images of 32 × 32) + 2 (for images of 64 × 64) + 3 (for images of 128 × 128) = 7 sets of 54 training files were tested to find out their set's wavelet that gave the best accuracy with general parameters for svm-train.

A single batch file was created with all the four commands of scaling the training file, training the training file, scaling the prediction file and prediction of the scaled prediction file using the model training file developed, for all 54 prediction-training files (54 × 4 = 216 commands). This batch file was executed 7 times for 7 pairs of 54 training-prediction files for one category of classes and for all IDs 7 × 4 = 28 times.

Table 7 represents the wavelets found for various sizes of images and various number of coefficients for all category of classes, that gave the best accuracy with general (default) parameters.

Table 7 Wavelets giving maximum accuracy with default/general parameters for training

Size	Coefficients	Wavelet	Accuracy
UPPER ZONE			
32 × 32	256	db2	91.3%
32 × 32	64	bior1.1	90.4%
64 × 64	1,024	sym4	90.4%
64 × 64	256	db1	90.4%
128 × 128	4,096	bior1.1	88.6%
128 × 128	1,024	bior1.1	90.0%
128 × 128	256	bior1.1	90.0%
MIDDLE ZONE			
32 × 32	256	rbio3.1	85.6%
32 × 32	64	coif5	59.7%
64 × 64	1,024	rbio3.1	86.3%
64 × 64	256	rbio3.1	86.0%
128 × 128	4,096	rbio3.3	84.8%
128 × 128	1,024	bior1.1	61.8%
128 × 128	256	bior1.1	38.4%
LOWER ZONE			
32 × 32	256	rbio3.3	81.4%
32 × 32	64	db2	82.1%
64 × 64	1,024	bior3.1	80.7%
64 × 64	256	rbio3.1	83.5%
128 × 128	4,096	bior3.1	75.7%
128 × 128	1,024	bior2.8	78.5%
128 × 128	256	rbio3.3	82.8%
NUMERALS			
32 × 32	256	rbio3.1	61.8%
32 × 32	64	rbio3.1	65.6%
64 × 64	1,024	rbio3.1	55.9%
64 × 64	256	rbio3.1	60.9%
128 × 128	4,096	bior1.1	52.8%
128 × 128	1,024	rbio3.1	54.0%
128 × 128	256	rbio3.1	61.5%

After finding the wavelets, all the parameter combinations discussed before were applied on the training-testing pair obtained from the specific wavelets that gave maximum accuracy for that zone or numerals.

Table 8 represents the results obtained after running the batch file of all combinations on the particular pair of training-testing files obtained using the particular wavelets determined earlier shown in Table 7.

Table 8 Maximum accuracy achieved for all sizes of character image and number of wavelet coefficients tried

Size	Coefficients	Accuracy
	UPPER ZONE	
32 × 32	256	91.8%
32 × 32	64	95.0%
64 × 64	1,024	93.6%
64 × 64	256	93.2%
128 × 128	4,096	90.0%
128 × 128	1,024	92.7%
128 × 128	256	93.6%
	MIDDLE ZONE	
32 × 32	256	90.7%
32 × 32	64	69.3%
64 × 64	1,024	90.3%
64 × 64	256	90.6%
128 × 128	4,096	70.3%
128 × 128	1,024	65.6%
128 × 128	256	65.6%
	LOWER ZONE	
32 × 32	256	89.2%
32 × 32	64	88.5%
64 × 64	1,024	87.8%
64 × 64	256	89.2%
128 × 128	4,096	88.5%
128 × 128	1,024	84.6%
128 × 128	256	83.5%
	NUMERALS	
32 × 32	256	71.8%
32 × 32	64	74.3%
64 × 64	1,024	67.1%
64 × 64	256	72.8%
128 × 128	4,096	61.5%
128 × 128	1,024	65.9%
128 × 128	256	70.6%

VII. CONCLUSION

After trying out all the combinations on features sets obtained from 32 × 32 sized images with 64 coefficients and 256 coefficients, from 64 × 64 sized images with 256 coefficients and 1,024 coefficients and from 128 × 128 sized images with 256 coefficients, 1,024 coefficients and 4,096 coefficients of upper zone, middle zone, lower zone and numerals, it was concluded that the maximum accuracies in all the zones and numerals were those shown in Table 9.

Table 9 Overall maximum accuracies achieved

Category	Size	Coefficients	Wavelet	Maximum Accuracy
Upper Zone	32 × 32	64	bior1.1	95.0%
Middle Zone	32 × 32	256	rbio3.1	90.7%
Lower Zone	32 × 32, 64 × 64	256	rbio3.3, rbio3.1	89.2%
Numerals	32 × 32	64	rbio3.1	74.3%

From above table it can be concluded that for upper zone maximum accuracy of 95.0% was achieved by feature sets containing 64 coefficients that were obtained by wavelet decomposition by bior1.1 wavelet of images from this zone of size 32 × 32. For middle zone maximum accuracy of 90.7% was achieved by feature sets containing 256 coefficients that were obtained by wavelet decomposition by rbio3.1 wavelet of images from this zone of size 32 × 32. For lower zone maximum accuracy of 89.2% was achieved by feature sets containing 256 coefficients that were obtained by wavelet decomposition by rbio3.3 and rbio3.1 wavelet of images from this zone of size 32 × 32 and 64 × 64 respectively. For numerals maximum accuracy of 74.3% was achieved by feature sets containing 64 coefficients that were obtained by wavelet decomposition by rbio3.1 wavelet of images from this zone of size 32 × 32.

Having a look at the majority, it can be concluded in general that the best accuracy was achieved through feature sets having 256 coefficients that were obtained by wavelet decomposition of 32 × 32 sized images using rbio3.1 wavelet.

REFERENCES

[1] P. Ahmed and Y. Al-Ohali, "Arabic Character Recognition: Progress and Challenges," J. King Saud Univ., Dept. of Comp. Sci., Technical Report Comp. and Info. Sci., pp. 85-116, March 16, 1999.

[2] A. Chadha, S. Mallik and R. Johar, "Comparative Study and Optimization of Feature Extraction Techniques for Content based Image," Int. Journal Comp. Applications, vol. 52, no. 20, pp. 975-8887, Aug. 2012..

[3] A. Choudhary, R. Rishi and S. Ahlawat, "Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique," in Proc. AASRI Conf Intelligent Systems and Control, pp. 306-312, 2013.

[4] S. Dalal and L. Malik, "A survey for Feature Extraction Methods in Handwritten Script Identification," IJSSST, vol. 10, no. 3.

[5] M. Z. Hossain, M. A. Amin and H. Yan, "Rapid Feature Extraction for Optical Character Recognition," North South Univ., Dept. of Comp. Sci., Technical Report, June 1, 2012.

[6] S. Lahmiri and M. Boukadoum, "Hybrid Discrete Wavelet Transform and Gabor Filter Banks Processing for Mammogram Features Extraction," IEEE, 2011.

[7] A. Lawgali, A. Bouridane, M. Angelova and Z. Ghassemlooy, "Handwritten Arabic Character Recognition: Which Feature Extraction Method?," Int. Journal Advanced Sci. and Tech., vol. 34, Sept. 2011.

[8] R. Polikar, "The Wavelet Tutorial," Rowan Univ., College of Engineering Web Servers, Technical Report, Jan. 21, 2001.

[9] J. Thorsten, Learning to Classify Text Using Support Vector Machines Methods, Theory and Algorithm, 1 st ed. Springer, 2002.

[10] "Watch parameters for lib-svm," www.csie.ntu.edu.tw/~r94100/libsvm-2.8/README

[11] "Wavelet Families," http://www.mathworks.in/help/wavelet/ref/waveletfamilies.html

AvneetKaur
avneetgalhotra@gmail.com

Research Interest: Pattern Recognition, Machine Learning, Data Mining
I have completed my M. Tech. from Thapar University, Patiala, Punjab, India in June, 2014. I have done my research work in the field of Pattern Recognition and Machine Learning. I have been associated with applications development since many years mainly in JAVA, XML, JavaScript and PL/SQL. During my Bachelor's I have been involved in web development using ASP.net, C, C++, PHP, SQL, HTML and CSS.

SanyaMalhotra
Sanya13105@iiitd.ac.in

Research Interest: Database Systems, Data Mining & Novelty Mining

I have completed my M. Tech. in Computer Science from IIIT D, Delhi, India in May, 2015. I have done my research work in the field of Data Analytics and Novelty Mining. I also have keen interest in web development using LAMP architecture, XML and PHP.