Unconstrained Kannada Handwritten Character Recognition Using Multi-level SVM Classifier

G.G. Rajput and Rajeshwari Horakeri

Department of Computer Science,
Gulbarga University, Gulbarga
Karnataka State, India
{ggrajput, rajeshwari_horakeri}@yahoo.co.in

Abstract. This paper presents an efficient zoning based method for recognition of handwritten Kannada characters using two sets of features, namely, crack codes (the line between the object pixel and background) and the density of the object pixels. A multi-level SVM is used for the classification purpose. The proposed method is implemented in two stages. In the first stage, similar shaped characters are combined into groups resulting in 22 classes instead of 49 classes, one class per character. Crack codes are used to assign the input character image to one of the groups. In the second stage, object pixel density is used to assign label to the input character image within that identified group. Experiments are performed on handwritten Kannada characters consisting of 24500 images with 500 samples for each character. Five-fold cross validation is used for result computation and average recognition rate of 91.02 % is obtained.

Keywords: Kannada, Crack codes, SVM, handwritten character, five-fold cross validation.

1 Introduction

Handwritten Character Recognition (HCR) is one of the important areas in pattern recognition field because it provides solution for document classifications, postal automation, bank check reading, automatic data entry, reading customer filled forms, etc. As a result of intensive research, several systems are available in many non-Indian scripts like Chinese, Arabic and English [1,2,3]. HCR is a challenging task as compared to recognition of printed characters due to unconstrained handwriting, touching of adjacent characters, broken strokes, poor binarization and noise. The task is more complicated for Indian languages due to complexity in the shape and number of characters compared to characters of non-Indian scripts like English and Chinese. A brief review of the work done towards the recognition of characters written in Indian scripts is presented below.

1.1 Related Works

A review on recognition of handwritten characters of south Indian scripts is presented in [4] and [5]. Printed and handwritten character recognition of Devanagari script using gradient features is presented in [6]. Gradient features are extracted using Sobel and Robert operators and fed to SVM classifier for recognition. Recognition of off-line Bangla handwritten compound characters using modified quadratic discriminant function (MQDF) is reported in [7]. Features used for recognition purpose are based on directional information obtained from the arc tangent of the gradient. A Robertss filter is applied on the normalized image to obtain gradient image. Next, the arc tangent of the gradient is initially quantized into 32 directions and the strength of the gradient is accumulated with each of the quantized direction. Finally, the frequencies of these directions are down sampled using Gaussian filter to obtain 392 dimensional feature vectors.

Recognition of handwritten Bangla basic characters is reported in [8]. The features are obtained by computing local chain code histograms of input image and the classification results are obtained after down sampling the histogram feature by applying Gaussian filter. Multilayer perceptrons (MLP) trained by back propagation (BP) algorithm is used for classification. Fuzzy model based recognition of handwritten Hindi characters is presented in [9]. Features consists of normalized distances obtained using the box approach. Pal et. al. [10] have proposed a quadratic classifier based scheme for the recognition of off-line handwritten characters of three popular south Indian scripts, namely, Kannada, Telugu, and Tamil. The bounding box of a character is segmented into blocks, and the directional features are computed in each block. These blocks are then down-sampled by a Gaussian filter, and the features so obtained are fed to a modified quadratic classifier for recognition. The mean accuracy of 90.34%, is reported for Kannada characters. FLD based unconstrained handwritten Kannada characters recognition is presented in [11]. The mean recognition accuracy of 68% is reported using Euclidean distance measure. In [12], handwritten Kannada character recognition system based on spatial features viz stroke density, stroke length and the number of stokes is proposed. Using k-NN classifier, the average recognition accuracy of 90.1% is reported for vowel characters. In [13], moment based features are used for recognition of Kagunita (the Kannada compound characters resulting from the consonant and vowel combination). These features are extracted using Gabor wavelets from the dynamically preprocessed original image. Multi-Layer Perceptron with Back Propagation Neural Networks are employed for character classification. Average recognition rate of 86% is reported for vowels and for consonants the average recognition reported is 65%. In [14], features are extracted using Fourier Transform and PCA (Principal Component Analysis) and recognition accuracy of 68.89% is obtained using PNN. A Neural network based Kannada character recognition system with limited set of characters with an accuracy of 99.17 % is presented in [15].

From the literature survey, it is clear that very few attempts have been made for handwritten Kannada characters with limited number of characters taken from the alphabet set (vowels and consonants). Further, the dataset used for

performing experiments is limited to small number of samples as compared to number of samples available in standard dataset of other languages like English. This motivated us to propose an efficient method for handwritten Kannada character recognition by including the entire alphabet set and creating a large dataset for performing experiments. This is challenging task as it results in more number of classes to be labeled.

The rest of the paper is organized as follows. In Section 2, handwritten Kannada characters database creation is presented. Section 3 deals with the feature extraction and section 4 describe classification. The experimental results are presented in Section 5 and conclusion is given in section 6.

2 Database Creation

Kannada, the official language of the southern Indian state of Karnataka, is a Dravidian language. The Kannada alphabet was developed from the Kadamba and Chalukya scripts used between the 5th and 7th centuries AD. Under the influence of Christian missionary organizations, Kannada script was standardized at the beginning of the 19th century. The writing style is from left to right. Kannada is built up from a base character set of 49 characters: 15 vowels and 34 consonants. Fig. 1 present a listing of the characters (vowels and consonants) used in Kannada scripts.

Fig. 1. (Kannada Character set (a)Printed (b)Handwritten (gray scale)

Since, the standard database for handwritten character images is not available in Kannada script; we created our own database for the proposed method. Dataset of handwritten Kannada characters (vowels and consonants) is created by collecting the handwritten documents from writers belonging to different professions. Sample image of Kannada handwritten vowels and consonants, in gray scale, is shown in Fig. 1(b). The collected data sheets were scanned using a flatbed scanner at a resolution of 300 dpi and stored as grayscale images. The noise, due to erratic hand movements and inaccuracies in digitization, present in the image is removed by applying median filter three times. Image binarization is performed using Otsus method. The noise at isolated locations and spikes around the end of the characters are removed using morphological open and close operations. A minimum bounding box is then fitted to the character and

the character is cropped. To bring uniformity among the handwritten characters, each image (located in bounding box) is size normalized to a size of 64×64 pixels. A total of 24500 binary digital images representing Kannada handwritten vowels are thus obtained. Each image represents a handwritten character (binary 1) that is unconstrained, isolated and clearly discriminated from the background (binary 0).

3 Feature Extraction

The proposed method is implemented in two stages. In Kannada character set, certain characters are similar in shape and hence are prone to misclassification. Hence, it is essential to perform recognition of a character in two stages. Firstly, similar shape characters are grouped together as shown in Table 1. This resulted in 22 classes as compared to original 49 classes [13]. For an input image, features are extracted using crack codes and fed to SVM classifier to classify the input character as to belong to one of these groups. In the second stage, pixel density of the input character is computed and fed to SVM classifier to label the character as to belong to one of the character in that group. In case, the input character belongs to a non-group character as shown in Table 1, the second stage of classification is not performed. The details of feature extraction method are given below.

Table 1. Group of similar shaped characters and other characters

Group of similar shaped characters		Non-group characters							
ಅ ಆ	ರ ಲ ಶ ಕ	ಇ	ಈ	ಉ	ಊ	ಋ	ೠ	ಃ	ಗ
ಚ ತ	ಹ ಡ ಘ ದ	ಞ	ಝ	ಣ	ಲ	ಳ			
ಎ ಏ ಐ ಒ ಟ	ಮ ಯ ಪು								
ಒ ಓ ಏ ಬ ಬ ಜ	ಸ ಸ								
ಥ ಡ ಧ ಧ ಧ ಧ ಧ ಧ ಷ									

3.1 Feature Extraction in the First Stage

The pre-processed character image is divided into non overlapping 64 zones of equal size. For each zone, compute crack code. Crack codes are efficient in representing the region borders. The crack belongs to the boundary that lies between foreground and background pixel. Encoding this line (a sequence of horizontal and vertical pixel edges) yields the crack code of the digitized object boundary (identified as the bold line in Figure 2). Codes are represented with 4 possible directions as shown in Fig. 2.

Fig. 2. Crack code (The "crack" is shown with the thick black line)

The crack code of the image is generated by traversing the character in each zone in anticlockwise direction. The method of computing the features using crack codes is presented in our earlier work [18]. The feature set of size 512 is obtained.

3.2 Feature Extraction in the Second Stage

To discriminate between similar shaped characters in the group (Table 1), we employed the concept of object pixel density. As discussed in previous paragraph, characters are grouped based on their shape. In each group, we observe that the common part in the shape does not contribute to the discriminating features. Hence, we decided to eliminate such common parts from the character in that group and compute the features for the remaining part of the character. Examples of such characters are shown in Table 2. Feature extraction is explained below.

Table 2. Character's groups after removal of common part

Original image	Common Part	Image after removal of common part	Original image	Common Part	Image after removal of common part	Original image	Common Part	Image after removal of common part

The common part of the character, for a specific group, is eliminated. Next, traversing the image column-wise from top left to bottom right in each zone, the occurrences of object pixel is counted which gives object pixel density for that zone. Obviously, certain zones have zero value. The feature vector of size 64 is obtained (Fig. 3).

4 Classification

Basically, Support Vector Machines (SVM) [19] is defined for two-class problem and it finds the optimal hyper-plane which maximizes the distance, the margin,

Fig. 3. Zoning of a character and object pixel density

between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data $\{x_m | m = 1, \dots, M\}$, the linear SVM classifier is defined as

$$F(x) = \sum \alpha_i x_i x + b \quad (1)$$

where x_j are the set of support vectors and the parameters j and b have been determined by solving a quadratic problem. The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs x_j , through a kernel function K defined as

$$K(x, y) = \phi(x)\phi(y) \quad (2)$$

This kernel function should satisfy the Mercer's Condition. The performance of SVM depends on the kernel used. A number of simple kernels include polynomial SVM, Radial Basis Function (RBF) SVM, and Two-layer neural network SVM. We have chosen RBF (Gaussian) kernel due to its better performance compared to other kernels. A SVM based classifier contracts all the information contained in the training set relevant for classification, into the support vectors. This procedure reduces the size of the training set by identifying its most important points. The SVM is applied to our multiclass character recognition problem by using one-versus-rest type method. More detailed treatment on principles and applications of SVM can be found elsewhere [20].

5 Experimental Results

Experiments are carried out on 24500 samples of handwritten Kannada characters. The database is divided in to train and test sets. Five-fold cross validation technique is used to test the performance of the proposed method. For each fold 19600 images are used for training and rests of the 4900 images are used for testing. Initially, character set consisting of entire 49 characters is considered and are labeled as 49 classes. Crack codes are used for feature extraction as explained in section 3 and the extracted features of the input character image are fed to trained SVM classifier for labeling the input character image. The average recognition accuracy of 87.24 % is obtained. The misclassification is mainly due to characters that are similar in shape. Taking into account of similar shaped characters, we then performed experiments in two stages as explained in section 3. Totally 9 groups are formed of 36 similar shaped characters and 13 individual classes of non-group characters, thus resulting in 22 classes instead of 49 classes

as shown in Table 1. In the first stage of classification we obtained 94.39% recognition accuracy for 22 classes. Once, the input character was classified to belong to a particular group, the pixel density features of the input character is fed to SVM classifier for character labeling in that group. An example of two stage feature extraction and subsequent classification for the first two vowel characters are shown in tables 3 to 6. It is clear from the tabulated results that, the method of grouping the characters and then implementing the character recognition in two stages provided acceptable recognition rate of 91.02% compared to that obtained without performing groups(Table 6). Further, to the best of our knowledge, results for the entire handwritten Kannada character set for the methods present in the literature are not available for comparison with our proposed method.

Table 3. Confusion matrix of first two vowel characters for 49 classes (not all characters are shown)

	ಅ	ಐ	ಏ	ಓ	ಔ	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ	ಠ	ಡ	ಢ	ತ	ಥ	ದ	ಧ	ನ	ಪ	ಫ	ಬ	ಭ	ಮ	ಯ	ರ	ಲ	ವ	ಶ	ಷ	ಸ	ಹ	ಳ	
ಅ	0	0	0	2	2	0	3	0	1	2	3	2	1	1	0	450	27	1	0	1	1	3	0																
ಐ	2	1	2	1	2	1	2	1	2	3	0	3	0	0	2	24	450	1	1	0	0	0	2																

Taking into account of similar shaped characters, we then performed experiments in two stages as explained in section 3. Totally 9 groups are formed of 36 similar shaped characters and 13 individual classes of non-group characters, thus resulting in 22 classes instead of 49 classes as shown in Table 1. In the first stage of classification we obtained 94.39% recognition accuracy for 22 classes. Once, the input character was classified to belong to a particular group, the pixel density features of the input character is fed to SVM classifier for character labeling in that group. An example of second stage feature extraction and subsequent classification for for the first two vowel characters are shown in tables 3 to 6. It is clear from the tabulated results that, the method of grouping the characters and then implementing the character recognition in two stages provided acceptable recognition rate of 91.02% compared to that obtained without performing groups(Table 6). Further, to the best of our knowledge, results for the entire handwritten Kannada character set for the methods present in the literature are not available for comparison with our proposed method.

Table 4. Confusion matrix of first two vowel characters at (a) stage-1 (obtained from table 4) (b) stage 2

		ಅ	ಐ	ಏ	ಓ	ಔ	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ	ಠ	ಡ	ಢ	ತ	ಥ	ದ	ಧ	ನ	ಪ	ಫ	ಬ	ಭ	ಮ	ಯ	ರ	ಲ	ವ	ಶ	ಷ	ಸ	ಹ	ಳ	
ಅ	ಐ	0	6	5	7	0	0	6	1	9	3	2	2	951	0	2	0	0	0	1	0	2	0	0																
		ಅ	ಐ	ಏ	ಓ	ಔ	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ	ಠ	ಡ	ಢ	ತ	ಥ	ದ	ಧ	ನ	ಪ	ಫ	ಬ	ಭ	ಮ	ಯ	ರ	ಲ	ವ	ಶ	ಷ	ಸ	ಹ	ಳ	
		0	6	5	7	0	0	6	1	9	3	2	2	951	0	2	0	0	0	1	0	2	0	0																

(a) stage-1

	ಅ	ಐ	ಏ
ಅ	467	10	
ಐ	9	465	

(b) stage 2

Table 5. Confusion matrix for first two vowel characters (consolidated from tables 4,5)

	ಅ	ಉ	ಊ	ಋ	ೠ	ಌ	಍	ಏ	ಐ	ಒ	ಓ	ಔ	ಋ	ೠ	ಌ	಍	ಏ	ಐ	ಒ	ಓ	ಔ		
ಅ	0	0	0	2	2	0	3	0	1	2	3	2	1	1	0	467	10	1	0	1	1	3	0
ಉ	2	1	2	1	2	1	2	1	2	3	0	3	0	0	2	9	465	1	1	0	0	0	2

Table 6. Comparison of recognition results of proposed method (consisting of 22 classes in the first stage) with 49 classes in terms of %

Class	49-Classes	2-level Classification	Class	49-Classes	2-level Classification	Class	49-Classes	2-level Classification
ಅ	90	93.4	ಋ	91.6	98.2	ಋ	93.04	93.06
ಉ	90	93	ಏ	86.2	93.2	ಋ	92.8	96.2
ಊ	82.8	98	ಏ	92.2	96.2	ಋ	93.4	98.4
ಋ	87.4	96	ಏ	90.4	92.8	ಋ	78.2	78.2
ಌ	81	86.2	ಏ	86.06	94	ಋ	81.6	81.6
಍	72.6	88.4	ಏ	93.8	93.8	ಋ	77.8	77.8
ಏ	81.4	88.8	ಏ	82	84.6	ಋ	62.8	62.8
ಏ	82.8	87.6	ಏ	84.8	94	ಋ	70.2	70.2
ಏ	87.8	84.4	ಏ	87.4	88.2	ಋ	68.2	68.2
ಏ	68.6	84.4	ಏ	90.4	92.4	ಋ	91	91
ಏ	79.6	88.4	ಏ	88.06	90	ಋ	77.6	77.6
ಏ	89.8	91.6	ಏ	90.2	91.8			
			ಏ	82.8	96.04			

6 Conclusion

In this paper we have presented a novel Kannada handwritten character recognition method using SVM classifier. Two types of feature extraction methods are proposed using used crack codes and pixel density. The proposed methodology is implemented in two stages and we have obtained encouraging results. The aim of the proposed system is to remove the confusions among similar shaped classes and thereby increase the recognition rate. This has been demonstrated by performing the experiments on the data set with and without grouping of characters. The proposed method can be extended for characters written in other scripts.

Acknowledgement. The authors are grateful to the referees for their helpful comments. The authors also thank Dr. P. S. Hiremath, Dr. B. V. Dhandra, GUG, and Dr. P. Nagbhushan, UOM, Mysore, for their guidance and support.

References

1. Nagy, G.: Chinese character recognition, A twenty five years retrospective. In: Proceedings of ICPR, pp. 109–114 (1988)
2. Lorigo, L.M., Govindaraju, V.: Offline Arabic handwriting recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(5), 712–724 (2006)
3. Pal, A., Singh, D.: Handwritten English Character Recognition Using Neural Network. International Journal of Computer Science & Communication 1, 141–144 (2010)

4. Indira, K., Sethu Selvi, S.: Kannada Character Recognition System: A Review. *Inter JRI Science and Technology* 1(2), 31–42 (2009)
5. Pal, U., Chaudhuri.: Indian script character recognition: a survey. *Pattern Recognition* 37(9), 1887–1899 (2004)
6. Holambe, A.N., Thool, R.C., Jagade, S.M.: Printed and Handwritten Character & Number Recognition of Devanagari Script using Gradient Features. *International Journal of Computer Applications* 2(9), 975–8887 (2010)
7. Pal, U., Wakabayashi, T., Kimura, F.: Handwritten Bangla Compound Character Recognition Using Gradient Feature. In: 10th International Conference on Information Technology (ICIT 2007), December 17–20, pp. 208–213 (2007)
8. Bhattacharya, U., Shridhar, M., Parui, S.K.: On Recognition of Handwritten Bangla Characters. In: Kalra, P.K., Peleg, S. (eds.) *ICVGIP 2006*. LNCS, vol. 4338, pp. 817–828. Springer, Heidelberg (2006)
9. Hanmandlu, M., Ramana Murthy, O.V., Madasu, V.K.: Fuzzy Model based recognition of handwritten Hindi characters. In: 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, December 3–5, pp. 454–461. IEEE (2007)
10. Pal, U., Sharma, N., Wakabayashi, T., Kimura, F.: Handwritten Character Recognition of Popular South Indian Scripts. In: Doermann, D., Jaeger, S. (eds.) *SACH 2006*. LNCS, vol. 4768, pp. 251–264. Springer, Heidelberg (2008)
11. Niranjana, S.K., Kumar, V., Hemantha Kumar, G., Aradhya, M.: FLD based Unconstrained Handwritten Kannada Character Recognition. *International Journal of Database Theory and Application* 2(3) (September 2009)
12. Dhandra, B.V., Hangarge, M., Mukarambi, G.: Spatial Features for Handwritten Kannada and English Character Recognition. *IJCA, Special Issue on RTIPPR* (3), 146–151 (2010)
13. Ragha, L.R., Sasikumar, M.: Feature Analysis for Handwritten Kannada Kagunita Recognition. *International Journal of Computer Theory and Engineering, IACSIT* 3(1), 1793–8201 (2011)
14. Aradhya, M., Niranjana, S.K., Hemant Kumar, G.: Probabilistic Neural Network based Approach for Handwritten Character Recognition. *Special Issue of IJCCT* 1(2,3,4); 2010 for International Conference (ACCTA 2010), August 3–5, pp. 9–13 (2010)
15. Kumar, S., Kumar, A., Kalyan, S.: Kannada Character Recognition System using Neural Network. *National Journal on Internet Computing* 1, 33–35
16. Gonzalez, R.C.G., Woods, R.E.: *Digital Image Processing*, 2nd edn. Pearson Education Asia (2002)
17. Alaei, A., Nagabhushan, P., Pal, U.: Fine classification of Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes. In: 10th International Conference on Document Analysis and Recognition, ICDAR, pp. 601–605
18. Rajput, G.G., Horakeri, R.: Handwritten Kannada Vowel Character Recognition System using Crack code and Fourier Descriptors. In: Sombattheera, C., Agarwal, A., Udgate, S.K., Lavangnananda, K. (eds.) *MIWAI 2011*. LNCS, vol. 7080, pp. 169–180. Springer, Heidelberg (2011)
19. Vapnik, V.N.: *The Statistical Learning Theory*. Springer, Berlin (1998)
20. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152 (1992)