A Novel Handwriting Grading System Using Gurmukhi Characters

Munish Kumar[1], Manish Jindal[2], and Rajendra Sharma[3]
[1]Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, India
[2]Department of Computer Science and Applications, Panjab University Regional Centre, India
[3]Department of Computer Science and Engineering, Thapar University, India

Abstract: *This paper presents a new technique for grading the writers based on their handwriting. This process of grading shall be helpful in organizing handwriting competitions and then deciding the winners on the basis of an automated process. For testing data set, we have collected samples from one hundred different writers. In order to establish the correctness of our approach, we have also considered these characters, taken from one printed Gurmukhi font (Anandpur Sahib) in testing data set. For training data set, we have considered these characters, taken from four printed Gurmukhi fonts, namely, Language Materials Project (LMP) Taran, Maharaja, Granthi and Gurmukhi_Lys. Nearest Neighbour classifier has been used for obtaining a classification score for each writer. Finally, the writers are graded based on their classification score.*

Keywords: *Gradation; feature extraction; peak extent based features; modified division point based features; NN.*

Received June 7, 2015; accepted January 13, 2016

1. Introduction

"Grading of Writers" means to judge the superiority of writing styles related to printed fonts. Handwriting grading systems can be used to grade the participants in a handwriting competition and can also be used for signature verification with suitable modifications. A handwriting grading system consists of the activities, namely, digitization, pre-processing, features extraction and grading based on the classification score. The activities in such a system have a close proximity with characters recognition system. A good number of researchers have done work for character recognition. For example, a printed Gurmukhi script recognition system has been proposed by Lehal and Singh [9].

Lorigo and Govindaraju [11] have presented a critical review on offline Arabic handwriting recognition. They have presented various techniques used for different stages of an offline handwritten Arabic character recognition system. Alaei *et al*. [1] have proposed an isolated handwritten Persian character recognition system. They employed SVM for classification and achieved a recognition accuracy of 98.1% with modified chain code based features. Dutta and Chaudhury [4] have presented a technique for isolated Bangla alphabets and numerals recognition using curvature features. Pal and Chaudhuri [12] have proposed a character recognition system using tree classifier. Their system was quite fast because pre-processing like thinning is not necessary in their scheme. They achieved a recognition accuracy of 96.0%. Li *et al*. [10] have proposed a hierarchical model for handwritten character recognition without consideration of orientation. They have proposed a few new models with multiple orientations at various positions. Bhattacharya *et al*. [3] have presented an approach for online Bangla handwritten character recognition. They developed a 50-class recognition problem and achieved an accuracy of 92.9% and 82.6% for training and testing, respectively. Kunte and Samuel [8] have presented efficient printed Kannada text recognition system. They considered invariant moments and Zernike moments as features and Artificial Neural Network (ANN) as classifier. They obtained a recognition accuracy of 96.8% using 2,500 characters.

Rampalli and Ramakrishnan [13] have presented an online handwritten Kannada character recognition system which works in combination with an offline handwriting recognition system. They improved the accuracy of online handwriting recognizer by 11% when its combination with offline handwriting recognition system is used. Alijla and Kwaik [2] have presented a handwritten character recognition using neural network for isolated Arabic characters. They achieved a recognition accuracy of 99.1% for trained writers. Kumar *et al*. [6] have presented a work on classification of characters and grading of writer based on offline Gurmukhi characters. They effectiveness of the system, however, was not established by them. In present work, we have tested the proposed grading system with a database collected from one hundred different writers (W_1, W_2, …, W_{100}) and in order to establish the reliability of this approach, we have also considered a character set taken from printed Gurmukhi font (Anandpur Sahib). This paper is divided into four sections. Section 2 presents feature

extraction techniques and classification process for obtaining classification score. Section 3 focuses on experimental results and section 4 presents conclusion of this paper.

2. Feature Extraction and Grading Based on Classification Score

A handwriting grading system consists of the activities, namely, digitization, pre-processing, features extraction, classification and grading based on the classification score as shown in Figure 1. The data considered in this work has already been digitized, pre-processed and segmented into isolated characters.

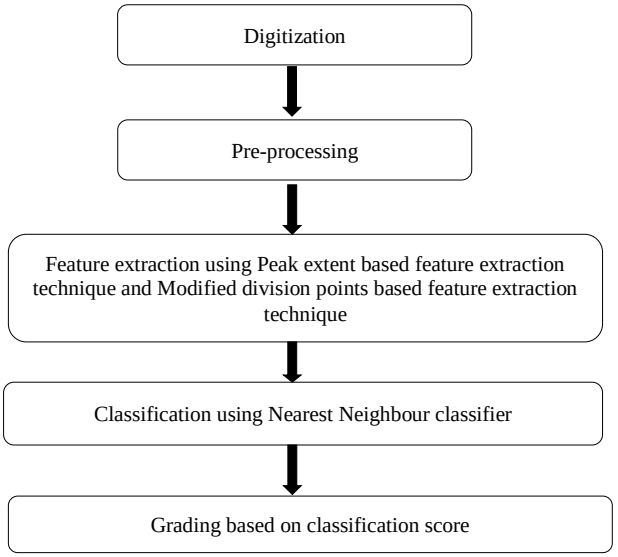
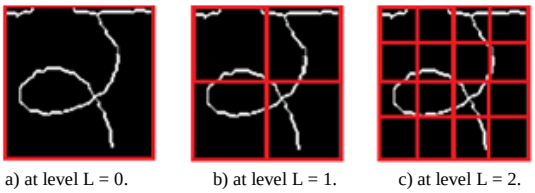


Figure 1. Block diagram of proposed handwriting grading system.

Feature extraction stage analyzes a handwritten character image and selects a set of features that can be used for grading the writers. In this work, for grading of the writers, we have used peak extent based features and modified division points based features as discussed in following sub-sections. In order to have an efficient recognition, in this work all the character images are reformed to the size of 88×88 pixels, using Nearest Neighborhood Interpolation (NNI) algorithm.

For extracting the features, initially, we have divided the digitized image into number of zones as shown in Figure 2. Let L be the current level of image. At this level, the number of the sub-images is 4^L. For example,when L = 1 the number of sub-images is 4 and when L = 2, it is 16. So, for every L a 4^L-dimensional feature vector is extracted. We have considered four levels of L(0, 1, 2 and 3) in this work.



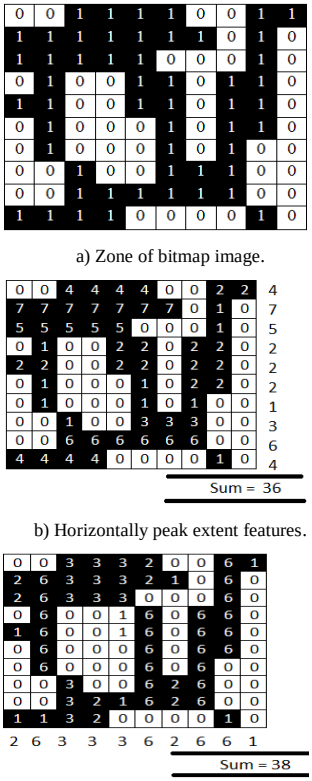
a) at level L = 0. b) at level L = 1. c) at level L = 2.

Figure 2. Digitized image of gurmukhi character (ਕ).

2.1. Peak Extent based Features

The peak extent based features [7] are extracted by taking into consideration the sum of the lengths of the peak extents that fit successive black pixels along each zone, as shown in Figure 3-a, b, and c. While fitting an extent along a series of successive black pixels within a region, the extent may be extended outside the boundary of the region if it continues in the next zone.

We have proposed a novel feature set, by using the horizontal peak extent features and the vertical peak extent features. In the horizontal peak extent features, we consider the sum of the lengths of the peak extents that fit successive black pixels horizontally in each row of a zone as shown in Figure 3-b, whereas in vertical peak extent features, we consider the sum of lengths of the peak extents that fit successive black pixels vertically in each column of a zone as depicted in Figure 3-c.



a) Zone of bitmap image.

b) Horizontally peak extent features.

c) Vertically peak extent features.

Figure 3. Peak extent based features.

The steps that have been used to extract horizontally peak extent features are given below:

Algorithm 1: Peak extent based features

Step 1: Input the initial value of L is 0.
Step 2: Divide a bitmap image into $4^{(L)}$ number of zones, each of equal sized (Figure 2).
Step 3: Find the peak extent as sum of successive foreground pixels in each row of a sub image at each level L.
Step 4: Replace the values of successive foreground pixels by peak extent value, in each row of a zone.
Step 5: Find the largest value of peak extent in each row.
Step 6: Obtain the sum of these largest peak extent sub-feature values for each sub-image and consider this as a

feature for the corresponding zone.
Step 7: For the zones that do not have a foreground pixel, take the feature value as zero.
Step 8: If L<3 then
(a) Set L=L+1
(b) Go to Step II
Else
Return
Step 9: Normalize the values in the feature vector in scale of 0 to 1.
Step 10: Return

These steps will give a feature set with 4^L elements at each level L. Similarly, for vertical peak extent features, we have extracted 4^L feature elements.

2.2. Modified Division Points based Features

In this technique, initially, we divide the character image into *n* (=100) zones, each of size 10×10 pixels. Let *Img*(*x*,*y*) be the character image having 1's for foreground pixels and 0's for background pixels. The proposed methodology is based on subdivisions of the character image so that the resulting sub-images have balanced numbers of foreground pixels. Let $V_p[xmax]$ be the vertical projection and H*p*[*ymax*] be the horizontal projection of the particular zone Z_1 as shown in Figure 4.

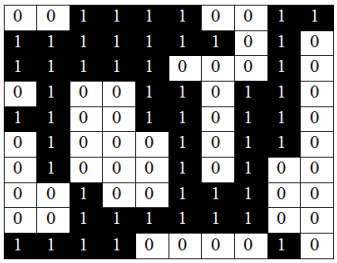


Figure 4. Bitmap of zone Z_1.

$$V_p = [4, 7, 6, 5, 6, 8, 3, 6, 7, 1]$$

$$H_p = [6, 8, 6, 5, 6, 4, 3, 4, 6, 5]$$

Here, in V_p the division point (d_v) of array is 5 (fifth element), because sum of the left sub-array elements and sum of the right sub-array elements is balanced as far as possible, if we consider fifth element into left sub-array. Similarly, calculate the division point (d_n) of H_p is taken as 4 (fourth element). The values of division points d_v and d_h of each zone are stored as features in the feature vector. This will give us 2*n* features for a character image [5].

The steps that have been used to extract these features are given below:

Algorithm 2: Modified division points based features

Step 1: Divide the bitmap image into n (=100) number of zones, each of size 10×10 pixels.
Step 2: Find the horizontal projection profiles H_p and vertical projection profiles V_p in each zone of a bitmap image.
Step 3: Store the horizontal projection profiles values in array H and vertical projection profiles values in array V.
Step 4: After that, calculate the value of division point (d_h) of array H and division point (d_v) of array V based on subdivisions of the arrays so that the resulting sub-arrays have balanced numbers of foreground pixels.
Step 5: Consider the values of (d_h) and (d_v) into left sub-array for make the possible balance between left sub-array and right sub-array.
Step 6: Calculate the values of (d_h) and (d_v) for each zone and placed in the corresponding zone as its feature.
Step 7: Corresponding to the zones that do not have a foreground pixel, the feature value is taken as zero.
Step 8: Normalize the values of feature vector.

These steps will give a feature set with 2*n* elements.

2.3. Grading based on Classification Score

Classification phase uses the features extracted in the feature extraction phase, for calculating classification score in the handwriting grading system. For classification, we have used Nearest Neighbours (NN) classifier. Then writers are graded based on their classification score. In the NN classifier, Euclidean distances from the candidate vector to stored vector are computed. The Euclidean distance between a candidate vector and a stored vector is given by,

$$d = \sqrt{\sum_{k=1}^{N} (x_k - y_k)^2} \quad (1)$$

Here, *N* is the total number of features in feature set, X_K is the library stored feature vector and y_K is the candidate feature vector. The class of the library stored feature producing the smallest Euclidean distance, when compared with the candidate feature vector, is assigned to the input character.

3. Experimental Results

As discussed in section 2, the gradation results, based on the values obtained by NN classifier are presented in this section. The classification scores obtained with NN classifier are normalized to [0, 100] in order to give the grade in percentage form. Feature-wise results of grading are presented in the following sub-sections. In the training data set of handwriting grading system, we have used four different printed Gurmukhi fonts, namely, LMP_Taran (F_2), Maharaja (F_3), Granthi (F_4) and Gurmukhi_Lys (F_5) as depicted in Table 1. In testing data set, we have considered these Gurmukhi characters written by one hundred different writers and one printed Gurmukhi font Anandpur Sahib to establish the effectiveness of the system. The experimental results in this paper have been presented in the form of graphs. These graphs present grading scores obtained for one hundred writers (W_1, W_2, …, W_{100}) and one printed Gurmukhi font F_1 (Anandpur Sahib). For the sake of better space usage, the calibration on x-axis does not include all the values. However, graphs contain the data for all 101 points of testing data set.

Table 1. Samples of printed characters from four Gurmukhi fonts (Training data set).

Script Character	LMP_Taran (F_2)	Maharaja (F_3)	Granthi (F_4)	Gurmukhi_Lys (F_5)
ੳ	ੳ	ੳ	ੳ	ੳ
ਅ	ਅ	ਅ	ਅ	ਅ
ੲ	ੲ	ੲ	ੲ	ੲ
ਸ	ਸ	ਸ	ਸ	ਸ

3.1. Grading using Peak Extent based Features

In this sub-section, gradation results of writers based on peak extent based features and *k*-NN classifier, are presented. Using this feature, it has been noted that font F_1 (with a score of 100) is the best font. On similar lines, it has also been observed that writer W_{64} (with a score of 41.84) is the best writer. The results of this classification process are presented in Figure 5.

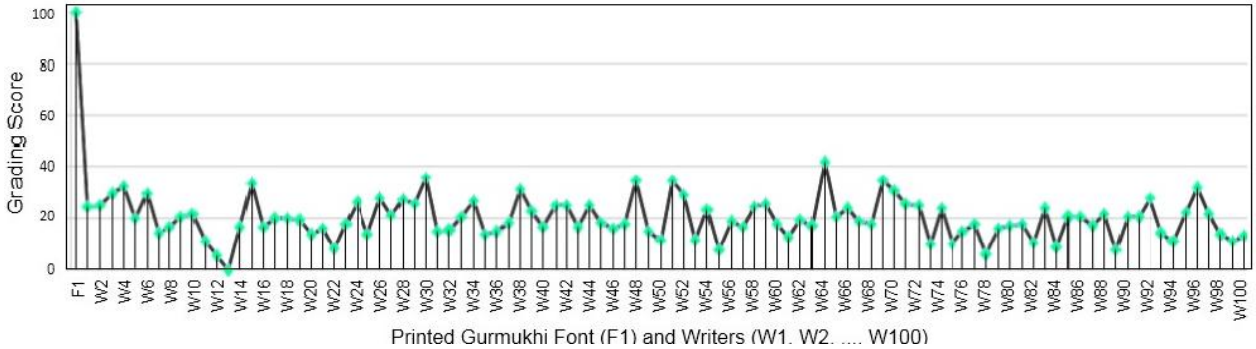


Figure 5. Grading using peak extent based features.

3.2. Grading using Modified Division Points based Features

When we use modified division points based features as an input to NN classifier, font F_1 (with a score of 100) comes out to be the best font. It has also been seen that writer W_{53} (with a score of 57.82) is the best writer amongst the one hundred writers taken in this study. MDP feature based grading results are depicted in Figure 6.

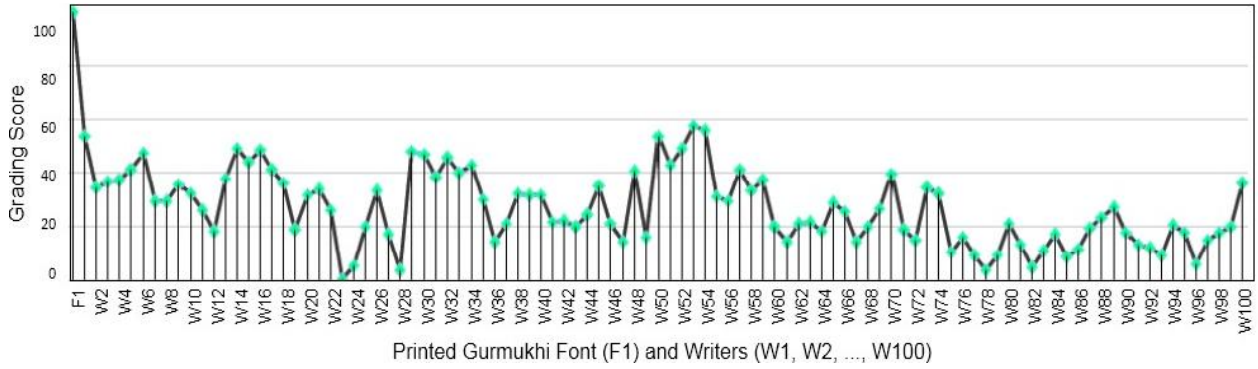


Figure 6. Grading using modified division points based features.

3.3. Final Grading using Average of Peak Extent based Features and Modified Divison Points based Features

Here, average grading, based on above mentioned two features has been presented. It has been observed that if we use average score of these two features, then font F_1 (with an average score of 100) is the best font and writer W_{30} (with an average score of 41.57) is the best writer. Final average grading scores of the font $F_1$and writers (W_1, W_2, …, W_{100}) considered in this study are shown in Figure 7.

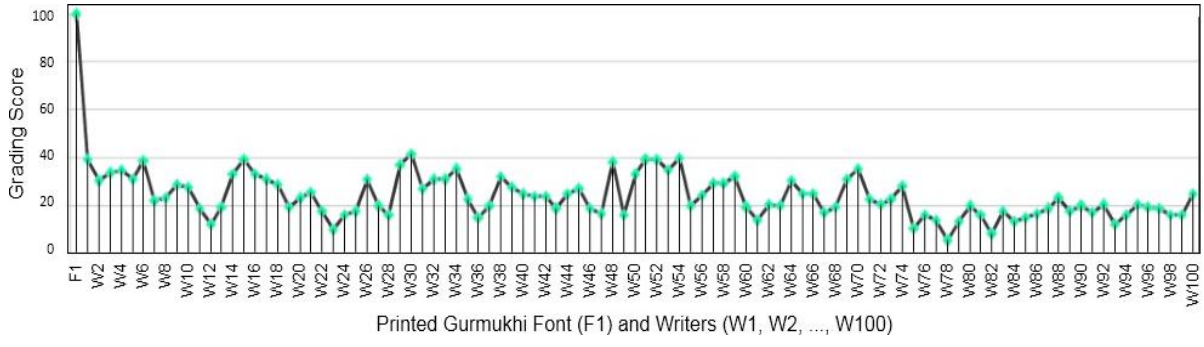


Figure 7. Final grading using average of peak extent based features and modified division points based features.

4. Conclusions

In this paper, offline Gurmukhi characters based novel handwriting grading system has been presented. Peak extent based features and Modified Division Points based features have been considered in this work. NN classifier has been used in the classification process. The system, proposed in present study, is tested with the help of four popular printed Gurmukhi fonts. As expected, in testing data set, the printed Gurmukhi font Anandpur Sahib has a better score of gradation in comparison with mortal writers, establishing the effectiveness of the proposed technique. The proposed grading system can be used as a decision support system for grading the handwritings in competitions.

This system can also be extended for grading writers using offline handwritten characters of other scripts after building the dataset of these scripts.

References

[1] Alaei A., Nagabhushan P., and Pal U., "A New Two-stage Scheme for the Recognition of Persian Handwritten Characters," *in Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition*, Kolkata, pp. 130-135, 2010.

[2] Alijla B. and Kwaik K., "OIAHCR: Online Isolated Arabic Handwritten Character Recognition Using Neural Network," *The International Arab Journal of Information Technology*, vol. 9, no. 4, pp. 343-351, 2012.

[3] Bhattacharya U., Gupta B., and Parui S., "Direction code based Features for Recognition of online Handwritten Characters of Bangla," *in Proceedings of the 9th International Conference on Document Analysis and Recognition*, Parana, pp. 58-62, 2007.

[4] Dutta A. and Chaudhury S., "Bengali Alpha-Numeric Character Recognition using Curvature Features," *Pattern Recognition*, vol. 26, no. 12, pp. 1757-1770, 1993.

[5] Kumar M., Jindal M., and Sharma R., "MDP Feature Extraction Technique for Offline Handwritten Gurmukhi Character Recognition," *Smart Computing Review*, vol. 3, no. 6, pp. 397-404, 2013.

[6] Kumar M., Sharma R., and Jindal M., "Classification of Characters and Grading Writers in Offline Handwritten Gurmukhi Script," *in Proceedings of International Conference on Image Information Processing*, Shimla, pp. 1-4, 2011.

[7] Kumar M., Sharma R., and Jindal M., "A Novel Feature Extraction Technique for Offline Handwritten Gurmukhi Character Recognition," *IETE Journal of Research*, vol. 59, no. 6, pp. 687-692, 2013.

[8] Kunte R. and Samuel R., "A Simple and Efficient Optical Character Recognition System for Basic Symbols in Printed Kannada Text," *Sadhana*, vol. 32, no. 5, pp. 521-533, 2007.

[9] Lehal G. and Singh C., "A Complete Machine Printed Gurmukhi OCR System," *Guide to OCR for Indic Scripts*, London, pp. 43-71, 2009.

[10] Li Z., Li H., Suen C., Wang H., and Liao S., "Recognition of Handwritten Characters by Parts with Multiple Orientations," *Mathematical and Computer Modeling*, vol. 35, no. 3-4, pp. 441-479, 2002.

[11] Lorigo L. and Govindaraju V., "Offline Arabic Handwriting Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712-724, 2006.

[12] Pal U. and Chaudhuri B., "OCR in Bangla: an Indo-Bangladeshi language," *in Proceedings of 12th International Conference on Pattern Recognition*, Jerusalem, pp. 269-274, 1994.

[13] Rampalli R. and Ramakrishnan A G., "Fusion of Complementary Online and Offline Strategies for Recognition of Handwritten Kannada Characters," *Journal of Universal Computer Science*, vol. 17, no. 1, pp. 81-93, 2011.

Munish Kumar received his Bachelors degree in Information Technology from Punjab Technical University, Jalandhar, India in 2006 and Post Graduate degree in Computer Science & Engineering from Thapar University, Patiala, India in 2008. He received his PhD degree in Computer Science from Thapar University, Patiala, India in 2015. He started his carrier as Assistant Professor in computer application at Jaito centre of Punjabi university, Patiala. He is working as Assistant Professor in Panjab University Rural Centre, Kauni, Muktsar, Punjab, INDIA. His research interests include Character Recognition and Handwriting Recognition.

Manish Jindal received his Bachelors degree in science in 1996 and Post Graduate degree in Computer Applications from Punjabi University, Patiala, India in 1999. He received his PhD degree in computer science & engineering from Thapar University, Patiala, India in 2008. He is working as Associate Professor in Panjab University Regional Centre, Muktsar, Punjab, INDIA. His research interests include Character Recognition.

Rajendra Sharma received his PhD degree in mathematics from the University of Roorkee (Now, IIT Roorkee), India in 1993. He is currently working as Professor at Thapar University, Patiala, India, where he teaches, among other things, queuing models and its usage in computer networks. He has been involved in the organization of a number of conferences and other courses at Thapar University, Patiala. His main research interests are in traffic analysis of Computer Networks, Neural Networks, and Pattern Recognition.