SVM Based Offline Handwritten Gurmukhi Character Recognition

Munish Kumar[1], M. K. Jindal[2], R. K. Sharma[3]
[1]Assistant Professor, Computer Science Department, GGS College for Women, Chandigarh, INDIA
[2]Associate Professor, Department of Computer Science and Applications, Panjab University Regional Centre, Muktsar, INDIA
[3]Professor, School of Mathematics & Computer Applications, Thapa r University, Patiala, INDIA
munishcse@gmail.com, manishphd@rediffmail.com, rksharma@thapar.edu

Abstract. Support Vector Machines (SVMs) have successfully been used in recognizing printed characters. In the present work, we have used this classification technique to recognize handwritten characters. Recognition of handwritten characters is a difficult task owing to various writing styles of individuals. A scheme for offline handwritten Gurmukhi character recognition based on SVMs is presented in this paper. The system first prepares a skeleton of the character, so that feature information about the character is extracted. Features of a character have been computed based on statistical measures of distribution of points on the bitmap image of character. SVM based approach has been used to classify a character based on the extracted features. In this work, we have taken the samples of offline handwritten Gurmukhi characters from one hundred different writers. The partition strategy for selecting the training and testing patterns has also been experimented in this work. We have used in all 3500 images of Gurmukhi characters for the purpose of training and testing. We have used diagonal and; intersection and open end points feature extraction techniques in order to find the feature sets for a given character. The proposed system achieves a maximum recognition accuracy of 94.29% with 90% training data and 10% testing data using intersection and open end points as features and SVM with polynomial kernel.

Keywords: Handwritten character recognition, Feature extraction, Diagonal features, Intersection and open end points features, SVM.

1. Introduction

Most of the published work on Indian scripts recognition deals with printed documents and very few articles deal with handwritten script problem. This has motivated us to consider the handwritten script recognition for Gurmukhi script. Handwritten Character Recognition, usually abbreviated as HCR, is the process of converting handwritten text into machine processable format. HCR is the field of research in pattern recognition and artificial intelligence. HCR can be online or offline. In online handwriting recognition, data are captured during the writing process with the help of a special pen and an electronic surface. Offline documents are scanned images of prewritten text, generally on a sheet of paper. Offline

handwriting recognition is significantly different from online handwriting recognition, because here, stroke information is not available [1, 2]. In this work, we have proposed a recognition system for offline handwritten Gurmukhi characters. A recognition system consists of the activities, namely, digitization, preprocessing, features extraction and classification. These activities in such a system have a close proximity with printed characters recognition system. A good number of researchers have already worked on the recognition problem of offline printed characters. For example, a printed Gurmukhi script recognition system has been proposed by Lehal and Singh [3]. Wen *et al.* [4] have proposed handwritten Bangla numerals recognition system for automatic letter sorting machine. Swethalakshmi *et al.* [5] have proposed handwritten Devanagri and Telugu character recognition system using SVM. The input to their recognition system consists of features of the stroke information in each character and SVM based stroke information module has been considered for generalization capability. Pal *et al.* [6, 7] have presented a technique for off-line Bangla handwritten compound characters recognition. They have used modified quadratic discriminant function for feature extraction. Pal *et al.* [8] have also used curvature feature for recognizing Oriya characters. Hanmandlu *et al.* [9] have reported grid based features for handwritten Hindi numerals. They have divided the input image into 24 zones. After that, they have computed the vector distance for each pixel position in the grid from the bottom left corner and normalized these distances to [0, 1] in order to obtain the features.

2. Gurmukhi script and data collection

Gurmukhi script is the script used for writing Punjabi language and is derived from the old Punjabi term "Guramukhi", which means "from the mouth of the Guru". Gurmukhi script has three vowel bearers, thirty two consonants, six additional consonants, nine vowel modifiers, three auxiliary signs and three half characters. Gurmukhi script is 12th most widely used script in the world. Writing style of Gurmukhi script is from top to bottom and left to right. In Gurmukhi script, there is no case sensitivity. The character set of Gurmukhi script is given in Figure 1. In Gurmukhi script, most of the characters have a horizontal line at the upper part called

The Consonants

headline and characters are connected with each other through this line.

ਸ ਹ ਕ ਖ ਗ ਘ ਙ ਚ ਛ ਜ ਝ ਞ ਟ ਠ ਡ ਢ ਣ ਤ ਥ ਦ ਧ ਨ ਪ ਫ ਬ ਭ ਮ ਯ ਰ ਲ

The Vowel Bearers

ੲ ਉ ਏ

The Additional Consonants (Multi Component Characters)

ਓ ਅ ਐ

The Vowel Modifiers

ਸ਼ ਜ਼ ਖ਼ ਫ਼ ਗ਼ ੍ਲ

Auxiliary Signs

ੇ ੈ ੋ ੌ ਿ ੀ ਾ ੁ ੂ

The Half Characters

ੱ ੰ ਂ

੍ਹ ੍ਰ ੍ਵ

Figure 1: Gurmukhi script character set.

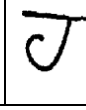
Script Character	W1	W2	W3	W4	W5
ੳ					
ਅ					
ੲ					
ਸ					
ਹ					

Figure 2: Samples of handwritten Gurmukhi characters.

For this work, a sample of 100 writers was selected from schools, colleges, government offices and other places. These writers were requested to write each Gurmukhi character. A sample of five handwritten Gurmukhi characters by five different writers (W1, W2, …, W5) is given in Figure 2.

3. The proposed recognition system

The proposed recognition system consists of the phases, namely, digitization, preprocessing, feature extraction and classification. The block diagram of proposed recognition system is given in Figure 3.

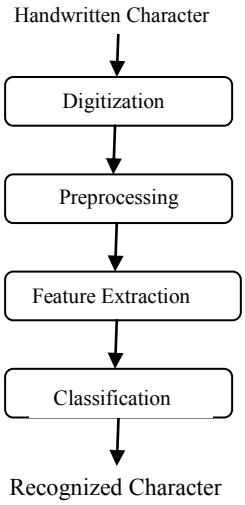


Figure 3: Block diagram of handwritten character recognition system.

3.1 Digitization

Digitization is the process of converting the paper based handwritten document into electronic form. The electronic conversion is accomplished using a process whereby a document is scanned and an electronic representation of the original document, in the form of a bitmap image, is produced. Digitization produces the digital image, which is fed to the pre-processing phase.

3.2 Preprocessing

Preprocessing is a series of operations performed on the digital image. Preprocessing is the initial stage of character recognition. In this phase, the character image is normalized into a window of size 100×100. After normalization, we produce bitmap image of normalized image. Now, the bitmap image is transformed into a contour image.

3.3 Feature extraction

The feature extraction stage analyzes a handwritten character image and selects a set of features that can be used for uniquely classifying the character. In this phase, the features of input characters are extracted. The performance of recognition system greatly depends on features that are being extracted. The extracted features

should be able to classify each character uniquely. We have used diagonal and intersection and open end points features for recognition of offline handwritten Gurmukhi characters.

3.3.1 Diagonal feature extraction

Diagonal features are playing an important role in order to achieve higher accuracy of the recognition system. Here, the skeletonized image of a character is divided into *n* (=100) zones. Now, diagonal features are extracted from the pixels of each zone by moving along its diagonals as shown in Figure 4. The steps that have been used to extract these features are given below.

Step I: Divide the skeletonized image into *n* (=100) number of zones, each of size 10×10 pixels.
Step II: Each zone has 19 diagonals; foreground pixels present along each diagonal is summed up in order to get a single sub-feature.
Step III: These 19 sub-feature values are averaged to form a single value and placed in corresponding zone as its feature.
Step IV: Corresponding to the zones whose diagonals do not have a foreground pixel, the feature value is taken as zero.

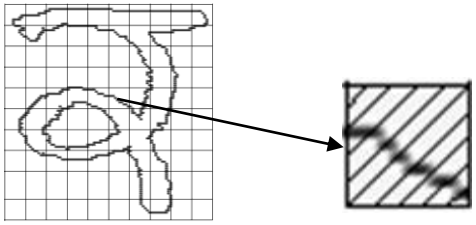


Figure 4: Diagonal feature extraction.

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

3.3.2 Intersection and open end points feature extraction

We have also extracted intersection and open end points for a character. An intersection point is the pixel that has more than one pixel in its neighborhood and an open end point is the pixel that has only one pixel in its neighborhood. Following steps have been implemented for extracting these features.

Step I: Divide the skeletonized image of a character into *n* (=100) zones, each of size 10×10 pixels (Figure 5).
Step II: Calculate number of intersection and open end points for each zone.

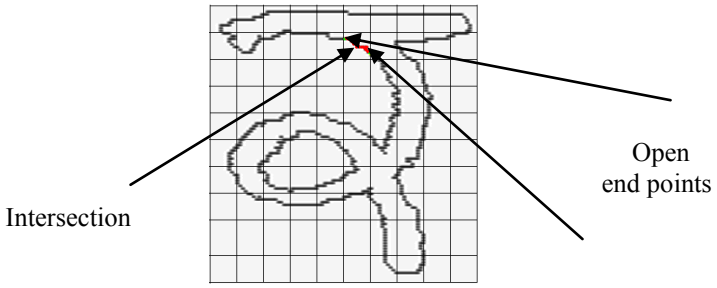


Figure 5: Intersection and open end point feature extraction.

This will give us 2*n* features for a character image.

3.4 Classification

Classification phase is the decision making phase of an HCR engine. This phase uses the features extracted in the previous stage for deciding the class membership. In this work, we have used Support Vector Machine (SVM) classifier for recognition. The SVM is a very useful technique for data classification. The SVM is a learning machine, which has been widely applied in pattern recognition. SVMs are based on statistical learning theory that uses supervised learning. In supervised learning, a machine is trained instead of programmed to perform a given task on a number of inputs/outputs pairs.

4. Experimental results and discussion

In this section, the results of recognition system for offline handwritten Gurmukhi characters are presented. The results are based on two feature extraction techniques, namely, diagonal and; intersection and open end point features. As stated earlier, we have also experimented some partitioning strategies while using the SVM as a classifier. We have divided the data set using five partitioning strategies. In the first strategy (strategy *a*), we have taken 50% data in training set and other 50% data in the testing set. In the second strategy (strategy *b*), we have considered 60% data in training set and remaining 40% data in the testing set. Strategy *c* has 70% data in training set and 30% data in testing set. Similarly, strategy *d* has 80% data in training set and 20% in testing set. Strategy *e* is formulated by taking 90% data in training set and remaining 10% data in testing set. SVM classifier has also been considered with three different kernels, namely, linear kernel, polynomial kernel and RBF kernel.

Feature-wise experimental results of testing are presented in the following sub-sections.

4.1 Diagonal features

In this section, the diagonal features have been considered to be taken as input to three types of SVM classifier, namely, SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel.

4.1.1 Recognition accuracy using SVM with linear kernel

In this sub-section, we have presented recognition results of five partitioning strategies (*a*, *b*, *c*, *d* and *e*) based on the diagonal features using SVM with linear kernel. Using this approach, *i.e.*, diagonal features and SVM with linear kernel, we achieved an accuracy of 81.83% when we use strategy *a* and achieved an accuracy of 90.29% when we used the strategy *e*. These results are depicted in Figure 6.

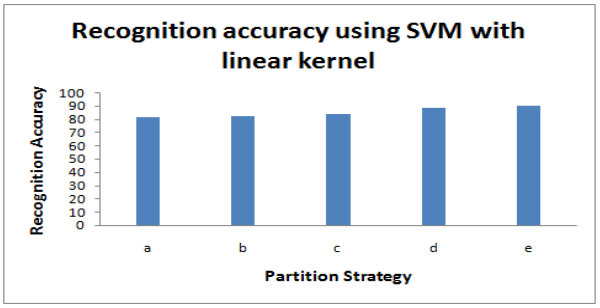


Figure 6: Recognition accuracy using SVM with linear kernel.

4.1.2 Recognition accuracy using SVM with polynomial kernel

When we use SVM with polynomial kernel, the results are not that encouraging. In partitioning strategy *a*, the accuracy that could be achieved was minimum at 43.6% and in strategy *e*, the accuracy achieved was maximum at 60.29%. These results are given in Figure 7.

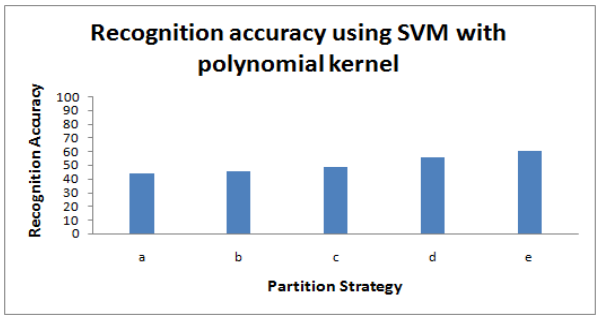


Figure 7: Recognition accuracy using SVM with polynomial kernel.

4.1.3 Recognition accuracy using SVM with RBF kernel

In this sub-section, recognition results using five partitioning strategies and based on the diagonal features using SVM with RBF kernel are presented. Here, partitioning strategy *a* gives the minimum accuracy (71.14%) and partitioning strategy *e* gives the maximum accuracy (84.29%). These results are given in Figure 8.

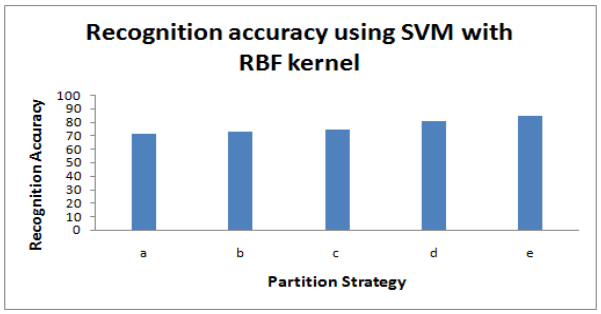


Figure 8: Recognition accuracy using SVM with RBF kernel.

4.2 Intersection and open end points features

In this subsection, the intersection and open end points features have been considered for inputting the classifier. Again three types of SVM as taken in 4.1 have been considered with these features.

4.2.1 Recognition accuracy using SVM with linear kernel

For the features under consideration and the SVM classifier with linear kernel, the minimum accuracy achieved is 81.26% in partitioning strategy *a* and the maximum accuracy achieved is 91.43% in partitioning strategy *e*. The results for this case are depicted in Figure 9.

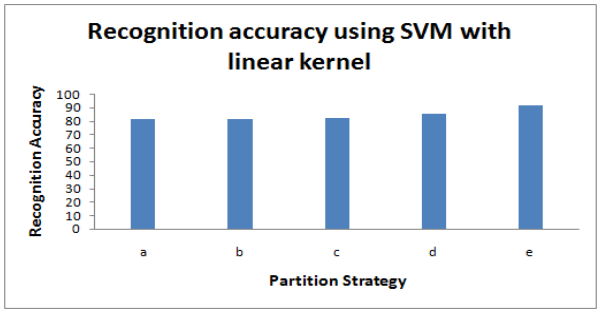


Figure 9: Recognition accuracy using SVM with linear kernel.

4.2.2 Recognition accuracy using SVM with polynomial kernel

In this sub-section, recognition results of five strategies and the SVM with polynomial kernel are presented. Again, the minimum accuracy is achieved when we use strategy *a* and the accuracy achieved is 82.69%. Maximum accuracy is again achieved when we use strategy *e* and the maximum accuracy achieved is 94.29%. These results are depicted in Figure 10.

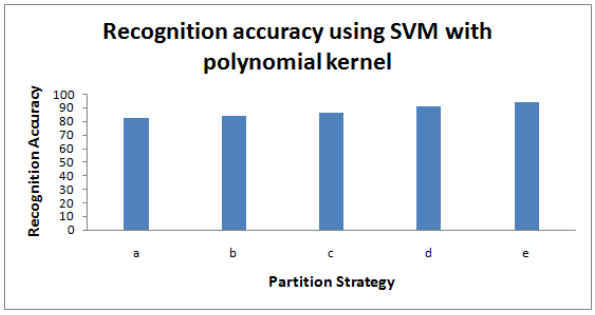


Figure 10: Recognition accuracy using SVM with polynomial kernel.

4.2.3 Recognition accuracy using SVM with RBF kernel

In this sub-section, recognition results of five partitioning strategies using SVM with RBF kernel are presented. Minimum accuracy achieved is 6% while using strategy *d* and maximum accuracy achieved is 22.23% while using strategy *a*. These results are depicted in Figure 11.

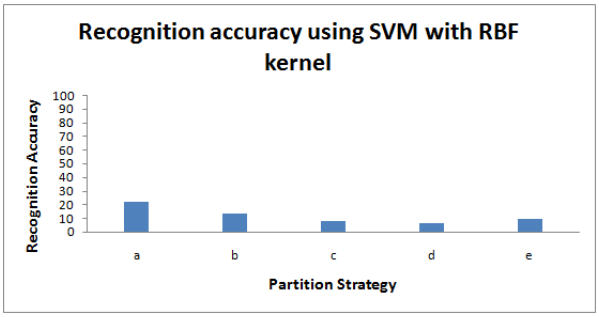


Figure 11: Recognition accuracy using SVM with RBF kernel.

4.3 Diagonal and intersection & open end points features

In this subsection, the diagonal and; intersection and open end points features simultaneously have been considered for inputting the classifier. Here, also again three types of SVM as taken in 4.1 have been considered with these features.

4.3.1 Recognition accuracy using SVM with linear kernel

In this sub-section, we have presented recognition results of five partitioning strategies based on the two features taken from 4.1 and 4.2 simultaneously using SVM with linear kernel. Using this approach, *i.e.*, SVM with linear kernel, we achieved an accuracy of 81.26% when we use strategy *a* and achieved an accuracy of 91.43% when we used the strategy *e*. These results are depicted in Figure 12.

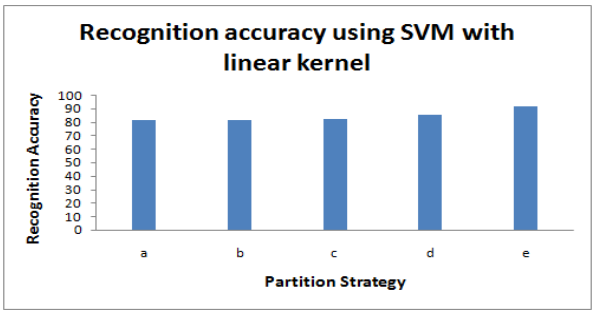


Figure 12: Recognition accuracy using SVM with linear kernel.

4.3.2 Recognition accuracy using SVM with polynomial kernel

In this sub-section, recognition results of five partitioning strategies and the SVM with polynomial kernel are presented. In partitioning strategy *a*, the accuracy that could be achieved was minimum at 82.69% and in strategy *e*, the accuracy achieved was maximum at 94.29%. These results are given in Figure 13.

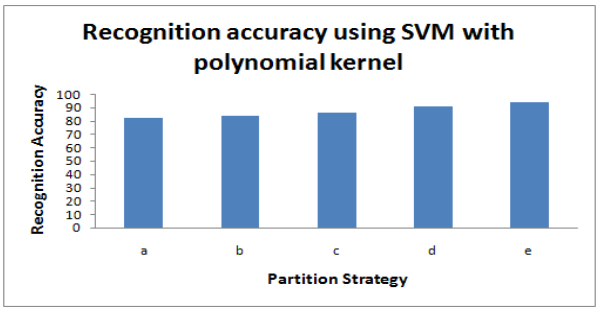


Figure 13: Recognition accuracy using SVM with polynomial kernel.

4.3.3 Recognition accuracy using SVM with RBF kernel

In this sub-section, recognition results of five partitioning strategies using SVM with RBF kernel are presented. Minimum accuracy achieved is 3.29% while using strategy *d* and maximum accuracy achieved is 19.37% while using strategy *a*. The results are depicted in Figure 14.

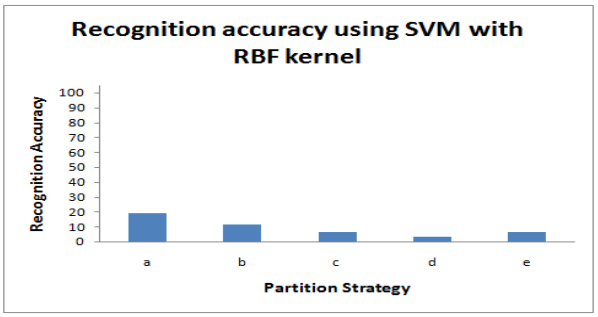


Figure 14: Recognition accuracy using SVM with RBF kernel.

5. Conclusion

The work presented in this paper proposes an offline handwritten Gurmukhi character recognition system. The features of a character that have been considered in this work include diagonal features and; intersection and open end points features. The classifier that has been employed in this work is SVM with three flavors, *i.e.*, SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel. The features have been inputted to the classifiers individually and have also been inputted simultaneously. The maximum recognition accuracy of 94.29% is achieved in this work for the case when we input the two features simultaneously to the SVM classifier with polynomial kernel. This accuracy can probably be increased by considering a larger data set while training the classifier. This work can also be extended for offline handwritten character recognition of other Indian scripts.

6. References

[1] Lorigo, L. M., and Govindaraju, V.: Offline Arabic handwriting recognition: a survey. IEEE Transactions on PAMI, 28, 5 (2006) 712-724

[2] Plamondon, R. and Srihari, S. N.: On-line and off- line handwritten character recognition: A comprehensive survey, IEEE Transactions on PAMI, 22, 1 (2000), 63-84

[3] Lehal, G. S. and Singh, C.: A Gurmukhi script recognition system, In Proceedings of 15th ICPR, 2 (2000), 557-560

[4] Wen, Y., Lu, Y. and Shi, P.: Handwritten Bangla numeral recognition system and its application to postal automation, Pattern Recognition, 40 (2007), 99-107

[5] Swethalakshmi, H., Jayaraman, A., Chakravarthy, V. S. and Sekhar, C. C.: Online handwritten character recognition of Devanagari and Telugu characters using support vector machine, In Proceedings of 10th IWFHR, (2006), 367-372

[6] Pal, U., Wakabayashi, T. and Kimura, F.: A system for off-line Oriya handwritten character recognition using curvature feature, In Proceedings of 10th ICIT, (2007), 227-229

[7] Hanmandlu, M., Grover, J., Madasu, V. K. and Vasikarla, S.: Input fuzzy for the recognition of handwritten Hindi numeral, In Proceedings of ITNG, (2007), 208-213

[8] Rajashekararadhya, S. V. and Ranjan, S. V.: Zone based Feature Extraction algorithm for Handwritten Numeral Recognition of Kannada Script, In Proceedings of IACC, (2009), 525-528

[9] Tripathy, J.: Reconstruction of Oriya alphabets using Zernike Moments, International Journal of Computer Applications, 8,8 (2010), 26-32

[10] Jindal, M. K.: Degraded Text Recognition of Gurmukhi Script", PhD Thesis, Thapar University, Patiala, India, 2008