



Improved Recognition Results of Medieval Handwritten Gurmukhi Manuscripts Using Boosting and Bagging Methodologies

Munish Kumar¹ · Simpel Rani Jindal² · M. K. Jindal³ · Gurpreet Singh Lehal⁴

Published online: 8 September 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Recognition of medieval handwritten Gurmukhi manuscripts is an essential process for resourceful contents exploitation of the priceless information contained in them. There are numerous Gurmukhi script ancient manuscripts from fifteenth to twentieth century's. In this paper, we have considered, work written by various persons from 18th to 20th centuries. For recognition, we have used various feature extraction techniques like zoning, discrete cosine transformations, and gradient features and different combinations of these features. For classification, four classifiers, namely, k -NN, SVM, Decision Tree, Random Forest individual and combinations of these four classifiers with voting scheme have been considered. Adaptive boosting and bagging have been explored for improving the recognition results and achieves the new state of the art for recognition of medieval handwritten Gurmukhi manuscripts recognition. Using this proposed framework, maximum recognition accuracy of 95.91% has been achieved using adaptive boosting technique and a combination of four different classifiers considered in this paper. To the best of our knowledge, this work is the successful attempt towards recognition of medieval handwritten Gurmukhi manuscripts and it can lead towards the development of optical character recognition systems for recognizing medieval handwritten documents in other Indic and non-Indic scripts as well.

Keywords Medieval handwriting recognition · Feature extraction · Classification · Adaptive boosting · Bootstrap aggregating

1 Introduction

Optical Character Recognition (OCR) is used for converting hard copy image containing machine-printed or handwritten text into a format which can be edited or processed by a computer. OCR frameworks have just accomplished impressive accuracy in fine printed text recognition. Historical books and medieval handwritten text documents containing bleed through text are yet to be recognized with acceptable accuracy. Gurmukhi is one of the

✉ Munish Kumar
munishcse@gmail.com

Extended author information available on the last page of the article

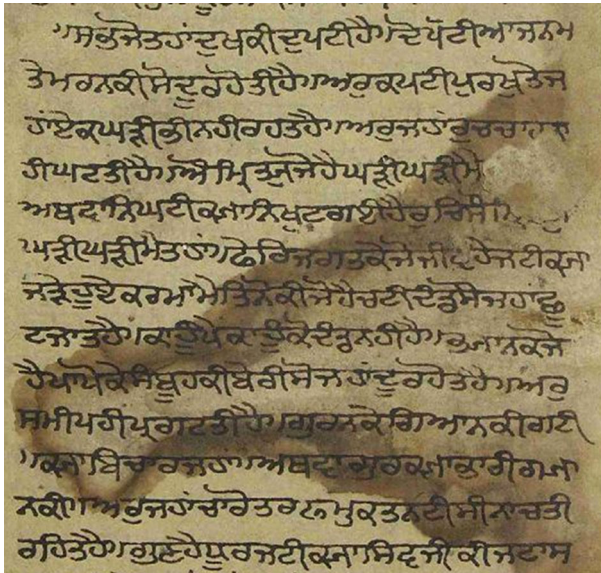


Fig. 1 A sample of medieval handwritten Gurmukhi script document

important Indian scripts used in north India. There are a number of medieval handwritten documents of the Gurmukhi script containing ancient manuscripts, printed books and typewritten documents of the sixteenth to the twentieth century. But, major composition is the handwritten manuscripts, mainly written by spiritual persons during that period. These records are profitable for the substance as well as particularly imperative for their physical appearance. A sample of the medieval handwritten document is shown in Fig. 1. To exploit the contents of these valuable Gurmukhi script manuscripts, recognition of these manuscripts is required. In this paper, an improved architecture has been proposed for recognition of isolated medieval handwritten Gurmukhi characters. Using this architecture, maximum recognition accuracy of 95.91% has been accomplished using AdaBoost (Adaptive Boosting) methodology. The organization of the paper is as follows: Sect. 2 illustrates the related work. Data collections are discussed in Sect. 3. Motivation and challenges for the proposed framework are discussed in Sect. 4. Section 5 describes the various phases of the proposed architecture. Section 6 presents the fundamentals of adaptive boosting and bootstrap aggregating methodologies. Experimental results and discussions are described in Sect. 7. Section 8, depicts the state-of-the-art work. Finally, inferences and future directions of the proposed work are presented in Sect. 9.

2 Related Work

Recognition of medieval handwritten manuscript documents is a challenging issue because of the unique properties of writing styles of individuals. More robust techniques are required to take care of this challenging issue as existing strategies for this design are not fruitful. Although handwritten document recognition is a timeworn problem and a considerable measure of research has just been done on it, but this issue is not completely tackled yet. Great outcomes have been accounted for online handwriting recognition because of the acces-

sibility of dynamic data, for example, the strokes of handwriting [1]. A lot of work has been accomplished for handwriting recognition at the character-level, which requires determining character boundaries [2]. As the complex writing styles make it difficult to extract character boundaries, the technique of jointly segmenting and recognizing the characters has been adopted. Lorigo and Govindaraju [3] have studied the different methodologies utilized for recognition of offline Arabic handwritten documents. Nikolos and Dimitrios [4] proposed basic and strong binarization technique for authentic historical documents. No reported work has been found in literature dealing with the problem of recognition of old handwritten manuscripts of any Indian script, in general. In particular, nothing has been done to recognize the old handwritten manuscripts of Gurmukhi script. Prominent work done in recognition of the Gurmukhi script documents is by Lehal and Singh [5, 6]. Alirezaee et al. [7, 8] built up a character recognition system for middle age Persian manuscripts. They have also proposed structural and statistical features based combined approach for character recognition of Greek and Egyptian documents. Feng and Manmatha [9] investigated different machine learning models to solve the historical handwritten manuscript recognition problem. They used Naive Bayes with kernel density estimates, SVM and conditional maximum entropy models for recognition of the 20 pages of George Washington's manuscripts. Kumar et al. [10] have proposed two efficient feature extraction techniques, namely, parabola fitting based and power curve fitting based for offline handwritten Gurmukhi character recognition. The classifiers that have been employed in their work are k -NN and SVM. Kumar et al. [11] have presented a hierarchical feature extraction technique for offline handwritten Gurmukhi character recognition. They have achieved a recognition accuracy of 91.80% with proposed hierarchical feature extraction technique while using the PCA feature set and SVM with a linear kernel classifier. They have also observed that PCA perform better than CFS and CON feature selection techniques for character recognition. Cheriet et al. [12] discussed the achievements of handwriting recognition research in the last twenty years and also discussed the challenges for handwriting recognition. Shayegan et al. [13] described a list of most widely used structural features, statistical features and transformation features in OCR applications. Tian et al. [14] have presented an efficient multilingual scene character recognition system based on co-occurrence of the histogram of oriented gradients. They have presented two new feature descriptors: Co-occurrence HOG (Co-HOG) and Convolutional Co-HOG (ConvCo-HOG) for accurate recognition of scene texts of different languages. They have experimented on five scene character datasets of three different languages, including, three sets in English, one set in Chinese and one set in Bengali. Sarkhel et al. [15] have proposed a novel multi-column multi-scale convolutional neural network (MMCNN) based architecture for the recognition of isolated handwritten characters of popular Indic scripts. They have evaluated the proposed framework on 10 publicly available datasets of isolated handwritten characters or digits including MNIST and promising results have been achieved by the proposed system for all of the datasets.

3 Data Set

Gurmukhi is the script used for writing the Punjabi language. Since, there is no standard database of Gurmukhi manuscripts, therefore, data has been collected for training and testing purpose. For that, pre-segmented characters of 150 documents of Gurmukhi manuscripts from different books have been scanned. This is the first of its kind of a database consisting of medieval handwritten manuscripts of any Indian script. In this work, we have taken 1140

samples of 43-classes (25 samples of each class) consisting of isolated medieval handwritten Gurmukhi characters. Medieval handwritten Gurmukhi characters are a little bit different from present day's handwriting because there may be disintegrating of pages, rust or defacing of ink etc. existing in the medieval document.

4 Motivations and Challenges

The physical condition of historical documents is affected by a number of reasons: ageing, rehashed utilize, staining, disintegrating of pages, rust, diffusing of ink and so on. Therefore, capturing text from such degraded historical documents and later recognizing the text is a crucial and typical task. There are many challenges also occurred during recognition of medieval handwritten Gurmukhi documents like in different phases of the recognition system. While digitizing the manuscripts, it is required to handle them very securely as most of the manuscript pages are fragile. Noise is always a major issue to be dealt with, which obstructs the recognition of the medieval historical documents. A historical document may contain a variety of slanting that is also an issue for text recognition. Segmentation is also an important issue because; most of the historical documents of the Gurmukhi script do not contain many word breaks. Text recognition of historical documents is a challenging problem as the shapes of the handwritten characters in historical documents are different from handwritten characters of modern time.

5 Proposed Architecture

The proposed system for medieval handwritten character recognition consists of various phases like digitization, pre-processing, feature extraction, and classification. These phases are briefly discussed in following sub-sections.

5.1 Digitization and Pre-processing

Foreground and background information is required to be preserved during the digitization of medieval handwritten documents. The quality of the scanner and camera influences the quality of the image. While digitizing the manuscripts, it is required to handle them very safely as most of the manuscript pages are fragile. The captured digital image of a document during digitization phase contains many problems, requiring digital enhancement of the image. The actual shape of the characters may change and different types of strokes can be found in a document. This occurs due to degradation of both the paper and the ink. Therefore, digital enhancement of the manuscript documents is essential and requires a good research work to manage this issue. Digitization creates the digital image which is sustained in the pre-processing phase. In this phase, the gray level character image is normalized into a window of size 64×64 using Nearest Neighborhood Interpolation (NNI) algorithm. After normalization, we produce a bitmap image of the normalized image.

5.2 Feature Extraction

Feature extraction is the most important phase of a character recognition system. The performance of a character recognition system depends heavily on the types and number of

feature values used. As far as it is concerned with handwritten historical documents, shape discrepancy among characters belonging to the same class is sometimes quite large because of the poor quality, low resolution of the document images, different styles of writing the characters, etc. In this paper, statistical feature extraction techniques, namely, zoning, discrete cosine transformations and gradient features have been used. These feature extraction techniques are briefly discussed in following sub-sections.

5.2.1 Zoning Features

This is one of the simplest and very powerful feature extraction methods. Using this method, the character image is divided into $N \times M$ ($N=8$ and $M=8$) zones. Features are extracted from each zone to generate the feature vector. Normally, the number of foreground pixels in each zone is considered a feature. Zoning is a local feature extraction technique to obtain the local characteristics of any character instead of the global characteristics of it. The number n_i of foreground pixels in each zone is counted and a feature vector is obtained by dividing n_i by N , where N is the total number of foreground pixels in the character image. Density values ($\text{Number_of_foreground_pixels}/\text{Total_number_of_pixels}$) are obtained for each zone. All density values are used to form the input feature vector for a particular character pattern.

5.2.2 Discrete Cosine Transformations (DCT) Features

DCT basically expresses a sequence of real data points into its real spectrum. The distribution of the coefficients of DCT depends on the nature of the studied image. For an image having low intensity with low spatial information, DCT provides good energy compaction in low frequency zones. In other words, conventional DCT coefficient distributions diminish while going to higher frequencies [16]. The DCT feature extraction technique has already been used for digit recognition by Ahmed [17] using four variations of DCT: DCT upper left corner coefficients, DCT zigzag coefficients, block based DCT upper left corner coefficients and block based DCT zigzag coefficients.

5.2.3 Gradient Features

The gradient of the image is the intensity at each point, giving the direction of the largest possible increase of light to dark and the rate of change in that direction. The gradient measures the magnitude and direction of the greatest change in intensity in a small neighborhood of each pixel. The rate of change of image/pixel intensity is modeled in gradient direction feature. The gradient direction feature is suitable for application using machine print/handwriting, binary/gray scale and low-resolution images [18]. A grayscale image is generated from an input binary image, and the gradient is calculated as described below.

1. Input normalized image of a character.
2. Mean filter of size 2×2 is repeatedly applied r times to obtain a grayscale image.
3. The gray scale image is normalized so that the mean and the maximum of the gray scale are 0 and 1, respectively.
4. Roberts filter given by Eqs. (1) and (2) is applied to each pixel $g(i, j)$ of the normalized image to calculate the gradient.

$$d_u = g(i + 1, j + 1) - g(i, j)$$

$$d_v = g(i + 1, j) - g(i, j + 1)$$

$$\text{Direction} : \theta(i, j) = \tan^{-1} \frac{d_v}{d_u} \quad (1)$$

$$\text{Strength} : f(i, j) = \sqrt{d_u^2 + d_v^2} \quad (2)$$

5.3 Classification

In this section, classification techniques used for character recognition have been discussed. Classification stage uses the features extracted in the feature extraction stage for deciding the class membership. Classification phase is the decision making phase of a character recognition system. In this study, k -NN, Linear-SVM, Decision Tree, and Random forest classifiers have been applied individually and in various combinations of these classifiers with a voting scheme for recognition.

5.3.1 k -NN Classifier

In the k -Nearest Neighbour (k -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

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

here N is the total number of features in the feature set, x_k is the library stored feature value and y_k is the candidate feature value. Let X_i be an input training sample with m features $\{x_{i1}, x_{i2}, x_{i3}, \dots, x_{im}\}$, n be the total number of input samples $\{i=1, 2, 3, \dots, n\}$ and m the total number of features $\{j=1, 2, 3, \dots, m\}$. It means lazy or instance learning is used in k -NN, while in other classifiers as SVM, eager learning is used. K specifies the number of nearest neighbours to be considered and the class of the majority of these neighbours is determined as the class of unknown pattern. More robust models can be achieved by locating k , where $k > 1$, neighbors and letting the majority vote decide the outcome of the class labeling. A higher value of k results in a smoother, less locally sensitive function. Generally, a larger value of k reduces the effect of noise on the classification, but makes the boundaries between classes less distinct. In this work, we have considered $k=1$.

5.3.2 Linear-SVM

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. SVM classifier has also been considered with a linear kernel. C-SVC type classifier in Lib-SVM tool has been used for classification purpose in this research work. In this work, initially we have computed recognition accuracy, using individuality of features and various combinations of these features like (zoning, DCT and gradient features) with a linear-SVM classifier.

5.3.3 Decision Tree and Random Forest (RF)

The attributes of the data are used by the decision tree learning algorithm for processing and decision making. Attributes in the decision tree are nodes and each leaf node is representing a classification. The ensemble for supervised learning method is called Random Forest. Random forest removed the over-fitting crisis of decision tree. Decision tree classifiers are used to classify various sub-samples of the dataset. The meta estimator that fits the number of decision tree classifiers for such design is called Random Forest. The random forest uses averaging to help in improving recognition accuracy and control over-fitting. Random forest is an outstanding algorithm in accuracy among existing supervised learning algorithms for classification and runs efficiently on large databases [19].

6 Adaptive Boosting and Bootstrap Aggregating

Boosting and Bagging are two techniques that can be used to improve the recognition results of classification.

6.1 Adaptive Boosting (AdaBoost)

Boosting is a way to manage machine learning in light of making a precise expectation rule by combining many moderately feeble and inaccurate rules. The AdaBoost algorithm of Freund and Schapire [20] was the principle useful boosting algorithm, and stays a standout amongst the most broadly utilized and contemplated, with applications in various fields. In this paper, we have used this methodology for improving the recognition results of medieval handwritten Gurmukhi manuscripts. AdaBoost is a sort of classifier with high precision. It gives the structure of order and makes the building up of sub-classifiers simple.

6.2 Bootstrap Aggregating (Bagging)

Bootstrap Aggregating (Bagging) is an ensemble method that creates separate samples of the training data set and creates a classifier for each sample. The results of these multiple classifiers are then combined (such as averaged or majority voting). Bagging, introduced by Breiman [21], takes bootstrap samples of objects and classifiers are trained on each sample. The classifier votes are then combined by majority voting. Breiman [21] shows that the predictors build using Bagging can make a weak unstable classifier significantly optimal. Bagging predictors have been used in various classifications and prediction problems in biology and social sciences. Diverse classifiers are generated by randomly selecting subsets of samples to train them.

7 Experimental Results

Experimental outcomes based on above mentioned features and classifiers with AdaBoost and Bagging algorithms are presented in this section. Various features are combined for improving the recognition accuracy. Different combinations of features and length of a combined feature vector, are also depicted in Tables 1, 2, 3 and 4. For classification, initially, we have

Table 1 Recognition accuracy using various features and classifiers

Features	Number of feature values	K-NN (%)	SVM (%)	Decision tree (%)	RF (%)
Zoning (F1)	100	88.47	84.38	52.04	82.15
Zoning (F2)	144	87.36	84.01	55.39	81.41
DCT (F3)	50	76.20	86.61	42.75	76.57
DCT (F4)	100	72.49	88.10	42.37	76.57
Gradient (F5)	200	66.91	84.38	34.20	66.17
Zoning+DCT (F1 + F3)	100 + 50 = 150	88.47	89.59	56.40	85.50
Zoning+DCT (F2 + F3)	144 + 50 = 194	88.47	90.70	56.13	83.64
Zoning+DCT (F1 + F4)	100 + 100 = 200	89.59	90.70	57.24	86.24
Zoning+DCT (F2 + F4)	144 + 100 = 244	87.36	91.82	55.76	86.98
Zoning+Gradient (F1 + F5)	100 + 200 = 300	84.38	86.98	53.90	85.50
Zoning+Gradient (F2 + F5)	144 + 200 = 344	86.24	88.47	58.36	84.75
DCT+Gradient (F3 + F5)	50 + 200 = 250	71.37	91.07	40.52	75.47
DCT+Gradient (F4 + F5)	100 + 200 = 300	72.49	91.44	39.03	75.83
Zoning+DCT+Gradient (F1 + F3 + F5)	100 + 50 + 200 = 350	84.01	90.70	54.64	85.13
Zoning+DCT+Gradient (F2 + F3 + F5)	144 + 50 + 200 = 394	86.24	90.70	56.87	84.75
Zoning + DCT + Gradient (F1 + F4 + F5)	100 + 100 + 200 = 400	84.75	91.07	54.64	85.87
Zoning + DCT + Gradient (F2 + F4 + F5)	144 + 100 + 200 = 444	86.98	90.33	56.87	84.38

considered individual classifiers as discussed in Sect. 5.3 and finally, we have also combined the classifiers and classification is done based on the voting scheme. For experimental work, the complete dataset is divided into training dataset and testing dataset. So, 70% data is considered as training dataset (798 samples) and remaining 30% considered as testing dataset (342 samples).

7.1 Recognition Accuracy Using Various Features and Classifiers

We have done experiments using zoning, DCT and gradient based features. Different combinations of these features are also experimenting. Classifier wise recognition accuracy is depicted in Table 1. As depicted in this Table 1, maximum recognition accuracy of 91.82% has been achieved using Linear-SVM classifier with a combination of zoning and DCT features. These results are graphically shown in Fig. 2.

Table 2 Recognition Accuracy using various features and classifiers with Bootstrap Aggregating methodology

Features	Number of feature values	K-NN (%)	SVM (%)	Decision tree (%)	RF (%)
Zoning (F1)	100	84.01	91.82	72.86	86.61
Zoning (F2)	144	85.13	86.24	66.54	82.15
DCT (F3)	50	77.32	87.73	62.08	77.32
DCT (F4)	100	71.37	86.98	59.10	76.57
Gradient (F5)	200	67.28	80.66	53.15	66.54
Zoning + DCT (F1 + F3)	100 + 50 = 150	86.24	88.84	71.74	85.87
Zoning + DCT (F2 + F3)	144 + 50 = 194	86.24	90.70	71.74	83.64
Zoning + DCT (F1 + F4)	100 + 100 = 200	87.73	90.33	72.49	85.87
Zoning + DCT (F2 + F4)	144 + 100 = 244	86.24	89.96	73.23	85.87
Zoning+Gradient (F1 + F5)	100+200= 300	84.01	85.13	72.49	82.89
Zoning+Gradient (F2 + F5)	144+200= 344	84.75	75.46	66.54	83.64
DCT+Gradient (F3 + F5)	50+200=250	71.74	90.70	66.17	76.20
DCT+Gradient (F4 + F5)	100+200= 300	71.74	91.07	63.94	77.32
Zoning+DCT + Gradient (F1 + F3 + F5)	100+50+ 200 = 350	84.75	91.82	72.11	85.50
Zoning + DCT + Gradient (F2 + F3 + F5)	144+50+ 200 = 394	85.13	92.19	69.88	86.98
Zoning+DCT+Gradient (F1 + F4 + F5)	100+100+ 200 = 400	84.01	91.82	72.86	86.61
Zoning+DCT+Gradient (F2 + F4 + F5)	144+100+ 200 = 444	86.24	92.19	70.63	87.36

7.2 Recognition Accuracy using Bootstrap Aggregating Methodology

In this sub-section, recognition results based on bootstrap aggregating are presented. Classifier wise recognition results using bootstrap aggregating are depicted in Table 2. As shown in Table 2, maximum recognition accuracy of 92.19% has been achieved using a combination of zoning, DCT and gradient as features and Linear-SVM as a classifier. Recognition accuracy achieved using bootstrap aggregating methodology is graphically depicted in Fig. 3.

7.3 Recognition Accuracy using Adaptive Boosting Methodology

Recognition results based on adaptive booting methodology are presented in this section (Table 3). It has been observed that the linear-SVM classifier with adaptive boosting methodology make this possible to achieve a maximum recognition accuracy of 91.07% when we use a combination of zoning, DCT and gradient features. Recognition accuracy achieved using adaptive boosting methodology is graphically depicted in Fig. 4.

Table 3 Recognition Accuracy using adaptive boosting methodology

Features	Number of feature values	K-NN (%)	SVM (%)	Decision tree (%)	RF (%)
Zoning (F1)	100	88.47	82.89	71.37	79.18
Zoning (F2)	144	87.36	84.38	75.09	82.15
DCT (F3)	50	76.20	85.50	63.94	76.95
DCT (F4)	100	72.49	87.36	61.33	72.49
Gradient (F5)	200	66.91	81.41	55.76	64.31
Zoning + DCT (F1 + F3)	100 + 50 = 150	88.47	88.47	74.72	85.50
Zoning + DCT (F2 + F3)	144 + 50 = 194	88.47	90.33	75.46	84.38
Zoning + DCT (F1 + F4)	100 + 100 = 200	84.75	89.21	76.57	84.75
Zoning + DCT (F2 + F4)	144 + 100 = 244	87.36	89.21	73.97	83.27
Zoning + Gradient (F1 + F5)	100 + 200 = 300	84.38	85.50	78.43	84.01
Zoning + Gradient (F2 + F5)	144 + 200 = 344	86.24	84.38	73.97	82.15
DCT + Gradient (F3 + F5)	50 + 200 = 250	71.37	88.10	66.91	75.09
DCT + Gradient (F4 + F5)	100 + 200 = 300	72.49	89.96	69.14	74.72
Zoning + DCT + Gradient (F1 + F3 + F5)	100 + 50 + 200 = 350	84.01	89.59	81.41	82.89
Zoning + DCT + Gradient (F2 + F3 + F5)	144 + 50 + 200 = 394	86.24	91.07	76.95	84.75
Zoning + DCT + Gradient (F1 + F4 + F5)	100 + 100 + 200 = 400	84.75	89.96	75.46	84.01
Zoning + DCT + Gradient (F2 + F4 + F5)	144 + 100 + 200 = 444	86.98	90.70	72.49	83.27

7.4 Recognition Accuracy Using Combination of Classifiers with Majority Voting Scheme

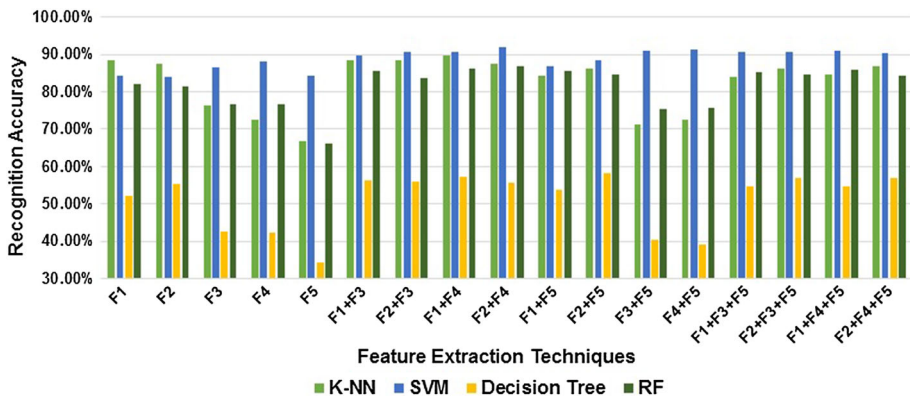
In this sub-section, recognition results based on a combination of all classifiers considered in this work with a majority voting scheme are illustrated (Table 4). These results are graphically shown in Fig. 5. It has been seen that a combination of all classifiers considered in this study with adaptive boosting methodology achieved maximum recognition accuracy of 95.91% when we use a combination of zoning and DCT features.

8 State of the Art Work

In this section, we have discussed about the state-of-the-art work related to the present study. Lehal and Singh [22] have achieved a recognition accuracy of 91.6% for printed *Gurmukhi* text recognition. Sharma and Jain [23] have extracted zoning features for handwritten *Gurmukhi* character recognition. They have employed two classifiers, namely, *k*-NN and SVM. They have achieved maximum recognition accuracy of 72.5% and 72.0%, respectively, with *k*-NN and SVM. Kumar et al. [24] have achieved a recognition accuracy of 94.29% using intersection and open end points based features and SVM classifier. In their work, they have considered 3500 samples for experimentation work. They have also concluded the recognition accuracy is depending upon the number of samples in training and testing dataset. Using

Table 4 Recognition Accuracy using a combination of classifiers with majority voting scheme

Features	Number of feature values	Combination of classifiers with voting scheme (%)	Adaptive boosting (%)	Bootstrap aggregating (%)
Zoning (F1)	100	88.47	86.98	88.10
Zoning (F2)	144	85.87	85.50	87.73
DCT (F3)	50	84.38	87.73	88.10
DCT (F4)	100	82.89	85.85	85.50
Gradient (F5)	200	79.18	79.55	78.43
Zoning+DCT (F1+F3)	100+50=150	91.07	93.68	92.56
Zoning+DCT (F2+F3)	144+50=194	90.70	91.07	93.30
Zoning+DCT (F1+F4)	100+100=200	92.56	90.33	93.30
Zoning+DCT (F2+F4)	144+100=244	91.82	95.91	92.93
Zoning+Gradient (F1+F5)	100+200=300	91.07	91.44	89.96
Zoning+Gradient (F2+F5)	144+200=344	90.33	92.93	89.96
DCT+Gradient (F3+F5)	50+200=250	82.89	86.24	88.10
DCT+Gradient (F4+F5)	100+200=300	88.47	89.21	89.59
Zoning+DCT+Gradient (F1+F3+F5)	100+50+200=350	91.44	94.05	93.30
Zoning+DCT+Gradient (F2+F3+F5)	144+50+200=394	91.44	91.79	92.93
Zoning+DCT+Gradient (F1+F4+F5)	100+100+200=400	92.56	93.68	93.68
Zoning+DCT+Gradient (F2+F4+F5)	144+100+200=444	92.56	94.79	92.93

**Fig. 2** Recognition accuracy using various features and classifiers

diagonal features with a k-NN classifier, they have achieved a recognition accuracy of 94.12% for the same dataset. Kumar et al. [25] have considered 10,500 samples for experimental work and used modified division points based features for offline handwritten Gurmukhi character recognition. They have achieved recognition accuracy of 84.57, 85.85 and 89.20% using Linear-SVM, MLP and k-NN classifiers, respectively. A few work has been done by Rani [26]

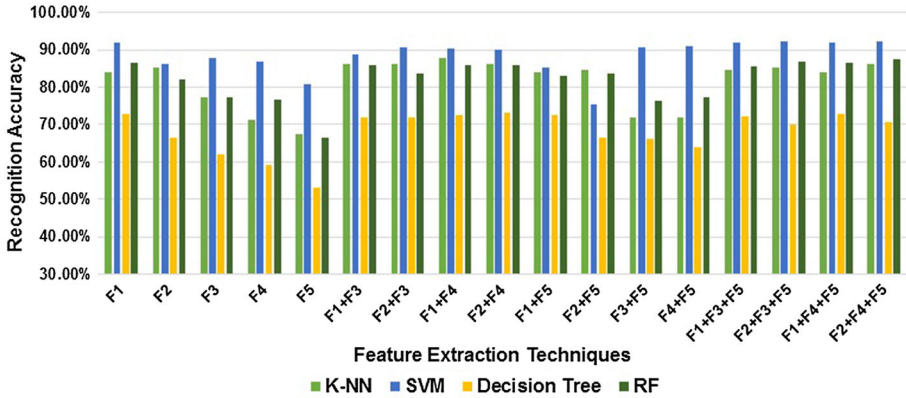


Fig. 3 Recognition accuracy achieved with bootstrap aggregating methodology

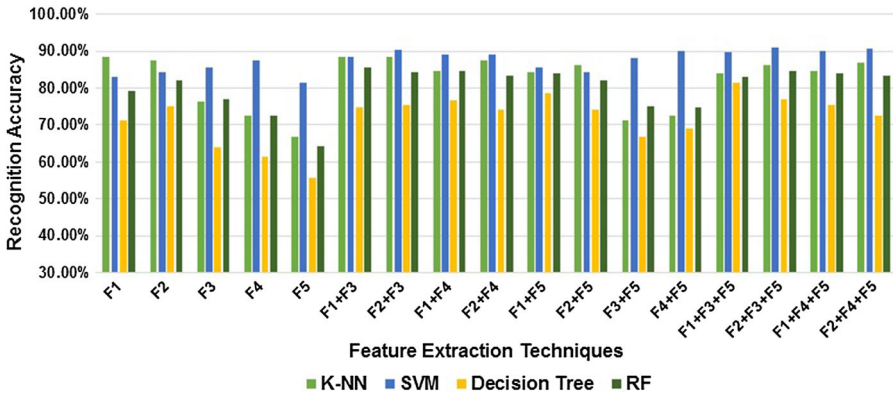


Fig. 4 Recognition accuracy achieved with adaptive boosting methodology

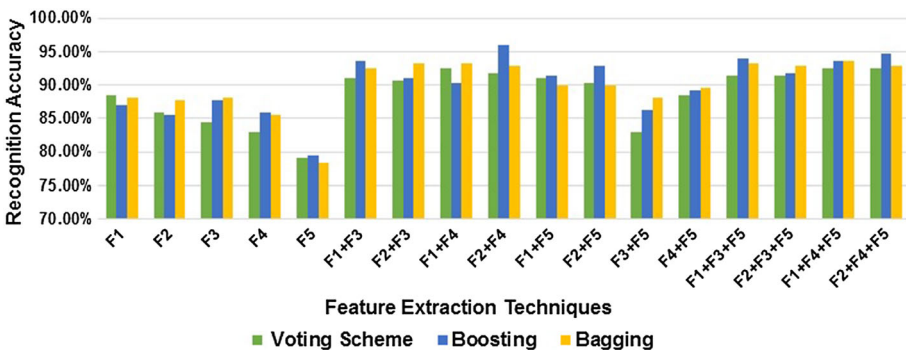


Fig. 5 Recognition accuracy achieved using a combination of classifiers with majority voting scheme

for medieval handwritten Gurmukhi character recognition. They have achieved maximum recognition accuracy of 93.31% recognition accuracy, using a combination of zoning, DCT and gradient based features and Linear-SVM classifier. Similarity in the shape of different characters in Gurmukhi script makes confusion during the recognition process as depicted in

Table 5 Similar shaped characters in Gurmukhi script

ਬ and ਥ	ੜ and ਝ	ੜ and ੜ	ੲ and ਟ
ਠ and ਨ	ਸ and ਮ	ਸ and ਗ	ਖ and ਪ
ਚ and ਜ	ਚ and ਦ	ਵ and ਫ	ਰ and ਕ
ੲ and ਲ	ਦ and ਢ	ਖ and ਬ	ਟ and ਠ

Table 5. In this paper, a framework for medieval handwritten Gurmukhi character recognition has been presented and achieved a recognition accuracy of 95.91% with a combination of all classifiers considered in this work with the adaptive boosting methodology.

9 Inferences and Future Directions

In this paper, an efficient recognition system has been presented for medieval handwritten Gurmukhi character recognition. Experimental results have affirmed the viability of the proposed approach and proved the characteristic advantages. We improved the recognition accuracy, using a combination of classifiers and majority voting scheme. Finally, with the help of adaptive boosting methodology we, achieved maximum recognition accuracy of 95.91%. We have also concluded that adaptive boosting and bootstrap aggregating methodologies can be widely used in pattern recognition field for improving the recognition results of various categories.

Compliance with ethical standards

Conflict of interest All authors declares that they have no conflict of interest.

References

1. Plamondon R, Srihari SN (2000) On-line and off-line handwritten recognition: a comprehensive survey. *IEEE Trans PAMI* 22(1):63–84
2. Bortolozzi F, Britto Jr A, Oliveria LS and Morita M (2005) Recent advances in handwriting recognition, In: *Proceedings of international workshop on document analysis (IWDA)*, pp 1–30
3. Lorigo LM, Govindaraju V (2006) Offline Arabic handwriting recognition: a survey. *IEEE Trans PAMI* 28(5):712–724
4. Nikolaos N, Dimitrios V (2008) A binarization algorithm for historical manuscripts, In: *Proceedings of the 12th WSEAS international conference on communications*, Heraklion, Greece, pp 41–51
5. Lehal GS (2001) Optical character recognition of machine printed Gurmukhi text. Ph.D. thesis, Punjabi University, Patiala, India
6. Lehal GS, Singh C (2006) A complete machine printed Gurmukhi OCR system. *Vivek* 16(3):10–17
7. Alirezaee S, Aghaeinia H, Faez K and Fard AS (2005) An efficient feature extraction method for the middle-age character recognition, In: *Proceedings of the international conference on intelligent computing*, pp 998–1006
8. Arrivault D, Richard N, Fernandez-Maloigne C, Bouyer P (2005) Collaboration between statistical and structural approaches for old handwritten characters recognition, In: *Proceedings of graph-based representations in pattern recognition*, pp 291–300
9. Feng SL, Manmatha R (2005) Classification models for historical manuscript recognition, In: *Proceedings of international conference on document analysis and recognition (ICDAR-05)*, pp 528–532
10. Kumar M, Sharma RK, Jindal MK (2014) Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. *Natl Acad Sci Lett* 37(4):381–391
11. Kumar M, Jindal MK, Sharma RK (2014) A novel hierarchical techniques for offline handwritten Gurmukhi character recognition. *Natl Acad Sci Lett* 37(6):567–572

12. Cheriet M, Yacoubi ME, Fujisawa H, Lopresti D, Lorette G (2009) Handwriting recognition research: twenty years of achievement and beyond. *Pattern Recognit* 42(12):3131–3135
13. Shayegan MA, Aghabozorgi S, Raj RG (2014) A novel two-stage spectrum-based approach for dimensionality reduction: a case study on the recognition of handwritten numerals. *J Appl Math* 2014:1–15
14. Tian S, Bhattacharya U, Lu S, Su B, Wang Q, Wei X, Lu Y, Tan CL (2016) Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. *Pattern Recognit* 51:125–134
15. Sarkhel R, Das N, Das A, Kundu M, Nasipuri M (2017) A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognit* 71:78–93
16. Mallikarjun H, Santosh KC, Pardeshi R (2013) Directional discrete cosine transforms for handwritten script identification, In: Proceedings of the 12th international conference on document analysis and recognition (ICDAR-13), pp 344–348
17. Ahmed H (2014) Handwritten digit recognition based on DCT features and SVM classifier, In: Proceedings of the 2nd world conference on complex systems (WCCS-14), pp 13–16
18. Aggarwal A, Rani R, Dhir R (2012) Recognition of Devanagari handwritten numerals using gradient features and SVM. *Int J Comput Appl* 48(8):39–44
19. Sharma DV, Lehal GS (2006) An iterative algorithm for segmentation of isolated handwritten words in Gurmukhi script, In: Proceedings of the 18th ICPR vol 2, pp 1022–1025
20. Freund Y, Schapire RE (1999) A short introduction to boosting. *J Jpn Soc Artif Intell* 14(5):771–780
21. Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
22. Lehal GS, Singh C (1999) Feature extraction and classification for OCR of Gurmukhi script. *Vivek* 12(2):2–12
23. Sharma DV, Jain U (2010) Recognition of isolated handwritten Characters of *Gurmukhi* script using neocognitron. *Int J Comput Appl* 10(8):10–16
24. Kumar M, Jindal MK, Sharma RK (2011) k-Nearest neighbor based offline handwritten Gurmukhi character recognition, In: The proceedings of the international conference on image information processing, Jaypee University of Information Technology, Wagnaghat (Shimla), pp 1–4
25. Kumar M, Sharma RK, Jindal MK (2013) A novel feature extraction technique for offline handwritten Gurmukhi character recognition. *IETE J Res* 59(6):687–692
26. Rani S (2016) Recognition of Gurmukhi handwritten manuscripts. Ph.D. thesis, Punjabi University, Patiala, India

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Munish Kumar¹ · Simpel Rani Jindal² · M. K. Jindal³ · Gurpreet Singh Lehal⁴

Simpel Rani Jindal
simpler_jindal@rediffmail.com

M. K. Jindal
manishphd@rediffmail.com

Gurpreet Singh Lehal
gslehal@gmail.com

¹ Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India

² Department of Computer Science & Engineering, Yadavindra College of Engineering, Talwandi Sabo, Bathinda, Punjab, India

³ Department of Computer Science & Applications, Panjab University Regional Centre, Muktsar, Punjab, India

⁴ Department of Computer Science, Punjabi University, Patiala, Punjab, India