Sādhanā (2019) 44:141
https://doi.org/10.1007/s12046-019-1126-9

Devanagari ancient documents recognition using statistical feature extraction techniques

SONIKA NARANG[1], M K JINDAL[2] and MUNISH KUMAR[3,*]

[1]Department of Computer Science, DAV College, Abohar, India
[2]Department of Computer Science and Applications, Panjab University Regional Centre, Muktsar, India
[3]Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India
e-mail: sonikanarang@davcollegeabohar.com; manishphd@rediffmail.com; munishcse@gmail.com

MS received 19 July 2018; revised 14 November 2018; accepted 4 March 2019; published online 13 May 2019

Abstract. Devanagari ancient document recognition process is drawing a lot of consideration from researchers nowadays. These ancient documents contain a wealth of knowledge. However, these documents are not available to all because of their fragile condition. A Devanagari ancient manuscript recognition system is designed for digital archiving. This system includes image binarization, character segmentation and recognition phases. It incorporates automatic recognition of scanned and segmented characters. Segmented characters may include basic characters (vowels and consonants), modifiers (*matras*) and various compound characters (characters formed by joining more than one basic characters). In this paper, handwritten Devanagari ancient manuscripts recognition system has been presented using statistical features extraction techniques. In feature extraction phase, intersection points, open endpoints, centroid, horizontal peak extent and vertical peak extent features are extracted. For classification, Convolutional Neural Network, Neural Network, Multilayer Perceptron, RBF-SVM and random forest techniques are considered in this work. Various feature extraction and classification techniques are considered and compared to the recognition of basic characters segmented from Devanagari ancient manuscripts. A data set, of 6152 pre-segmented samples of Devanagari ancient documents, is considered for experimental work. Authors have achieved 88.95% recognition accuracy using a combination of all features and a combination of all classifiers considered in this work by a simple majority voting scheme.

Keywords. Ancient manuscripts; Devanagari historical documents; feature extraction; classification.

1. Introduction

The recognition of characters from scanned printed or handwritten documents can be made using one major area of pattern recognition called Optical Character Recognition (OCR). OCR involves many steps like image pre-processing and segmentation followed by feature extraction, classification and post-processing. Text recognition of ancient documents poses many challenges because of the degraded quality of documents. Degradation of ancient documents may be due to age or writing style. There may be documents with ink stains, faded ink, uneven space between text lines, overlapping of text lines or characters, different layouts, broken characters or torn pages. India has a very rich history and culture and ancient documents of India are a wealth of knowledge. These documents are preserved by libraries and museums, but the purpose of the preservation has not been served [1]. These documents are not available for public owing to their delicate condition. To preserve our cultural heritage and for automated processing of documents, libraries and national archives have initiated work on digitizing historical documents [1]. Hence, this work has motivated us to offer a system for recognition of Devanagari ancient documents. In this paper, authors have recognized characters of Devanagari ancient documents using various features, namely, intersection and open endpoints (F1), centroid (F2), horizontal peak extent (F3) and vertical peak extent (F4). For classification, authors have considered five classifiers, namely, MLP (C1), Neural Network (C2), Convolution Neural Network (CNN, C3), RBF-SVM (C4) and random forest (C5), for performing the classification task. This paper is structured as follows. Characteristics of Devanagari script are discussed in section 2. Prior work is presented in section 3. Section 4 describes the proposed methodology. Results and discussion are given in section 5. Concluding notes and future directions are presented in section 6.

2. Characteristics of Devanagari script

India is a multi-script and multi-lingual country. One of the most popular scripts in India is the Devanagari script. Devanagari is used to write Hindi, Marathi, Nepali and

*For correspondence

Sanskrit languages. Devanagari has 11 vowels, 33 consonants and 3 common conjuncts. These are known as basic characters. We can write vowels as independent characters or by using some special diacritical marks. These diacritical marks are known as modifiers or *matras*. Characters formed using modifiers are called conjuncts. Some characters, formed by combining two or more consonants, are called compound characters. A set of the basic Devanagari characteristics is delineated in tables 1 and 2. There is a horizontal line at the upper part of every Devanagari character, known as *Shirorekha* or *headline*. Various characters of the same word are joined by the headline. Headline poses a challenge while segmenting characters. There are many similarly shaped characters, which make character recognition a challenging problem [2].

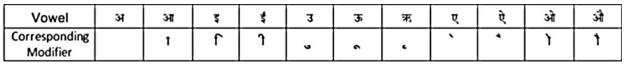
3. Related work

Kleber *et al* [3] presented a method to generalize missing parts of a degraded ancient document based on a priori knowledge. They introduced an algorithm for ruling estimation of Glagolitic texts that are based on the extraction of text line. The proposed algorithm is appropriate for degraded manuscripts. For line extraction, they have considered the approach based on connected component. Bansal and Sinha [4] developed a complete recognition system for text in Devanagari script. They observed real-life printed text in Devanagari that contained character fusions and noisy environment. For feature extraction, they employed region coverage of the core strip, vertical bar feature, horizontal zero crossings, number of positions of the vertex points, moments and structural descriptors of the characters as features. By employing a decision tree classifier, they attained an accuracy of about 93.0%. Kim *et al* [5] presented a dedicated OCR system for Hanja historical documents. Sousa *et al* [6] proposed an OCR system based on fuzzy logic for ancient printed documents. Their proposed OCR builds fuzzy membership functions from oriented features extracted using Gabor filter banks. Cecotti and Belaid [7] presented a hybrid combination approach complemented by specialized ICR for ancient documents. They proposed a model for combining several OCRs and specialized intelligent character recognition based on the CNN. Diem and Sablatnig [8] presented a work to recognize degraded characters using local features. This work was done in the ancient manuscript where the character was washed out (partially visible) due to age. Due to washout characters, it was not suitable for binarization. Hence, segmentation-free approach based on local descriptors was developed. Local descriptors are classified using Support Vector Machine (SVM) and then identified by a voting scheme of neighbouring regional descriptors.

Raghuraj *et al* [9] presented a scheme to develop complete OCR for five different fonts and sizes of Devanagari characters. They used various approaches like matrix matching, fuzzy logic, feature extraction, structural analysis and neural networks. They used a histogram-based threshold approach to convert images to two-tone images, median filters for salt and pepper noise and derivative operators to increase edges. They used three features: mean distance, histogram of projection based on the spatial position of pixels and the histogram of projection based on pixel value. They used artificial neural network (ANN) approach for classification. Holambe *et al* [10] presented an overview of feature extraction and selection methods for recognition of numerals and characters of the Devanagari script. They used Zernike moment for feature extraction and - k-NN classifier based on Euclidean distance. Yadav *et al* [11] proposed an OCR system for printed Hindi recognition, using ANN. They used projection profiles for segmentation and histograms of projection based on mean distance, histogram of projection based on pixel value, vertical zero crossing for feature extraction and back-propagation neural network with two hidden layers for classification. They achieved a recognition rate of about 90.0%. Yunxue *et al* [12] introduced restoration method for character image in order to recognize unconstrained character handwritten in Chinese. They modelled the character image by combining the ideal character image with two types of noise images, namely, omitted stroke noise image and added stroke noise image. Restoration is done in order to maintain the original gradient features. The determined features are then employed to differentiate similar characters.

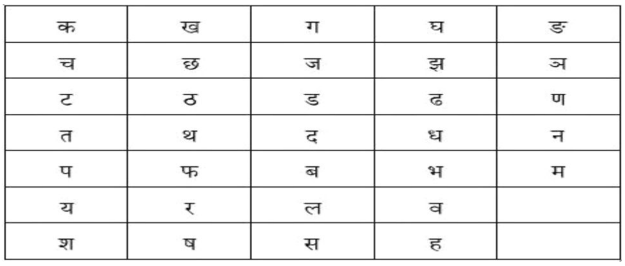
Katiyar and Mehfuz [13] developed a hybrid recognition system to recognize offline handwritten characters. They merged multiple features extracted using seven different approaches. In order to optimize the number of features, Genetic Algorithm is employed. Adaptive Multi-Layer Perceptron (MLP) classifier is employed for classification purpose. For experiments, CEDAR (Centre of Excellence for Document Analysis and Recognition) database on English alphabet is used. Belhe *et al* [14] developed a recognition system for handwritten words in Hindi. For

Table 1. Vowels and corresponding modifiers.



Vowel	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Corresponding Modifier		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ

Table 2. Consonants.



क	ख	ग	घ	ङ
च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	
श	ष	स	ह	

recognition, HMM and tree classifiers are employed. As a result, a recognition accuracy of 89.0% based on 10,000 Hindi words has been attained. Lehal and Singh [15] worked for feature extraction and classification for OCR of Gurmukhi script. They developed two sets of features: Primary feature set and Secondary feature set. They used binary tree classifier and nearest neighbours for classification. In order to recognize offline handwritten Gurmukhi characters, Kumar *et al* [16] proposed a novel feature extraction technique. Various feature extraction techniques based on curvature features have also been proposed by them for recognition of offline characters handwritten in Gurmukhi script [17]. Kumar *et al* [18] also presented a survey for character and numeral recognition of various non-Indic and Indic scripts. Based on the related work, authors noticed that a lot of work has been done for printed and handwritten text recognition of different scripts. However, text recognition system for ancient documents is not provided. Hence, in this paper, the authors have presented a Devanagari ancient character recognition that is a combination of multiple classifiers with the majority voting scheme.

4. Proposed system

For recognition of Devanagari ancient manuscripts, the proposed system consists of various phases like image acquisition, pre-processing, segmentation, feature extraction and classification. Block diagram of the proposed system is depicted in figure 1.

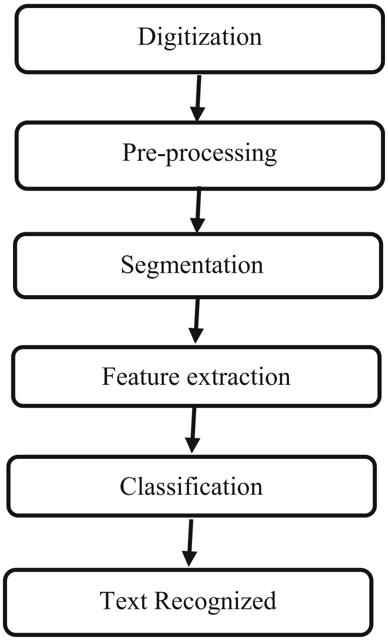


Figure 1. Block diagram of the proposed system.

4.1 *Image acquisition and digitization*

For this work, Devanagari ancient documents are collected from various libraries and museums. A sample of the Devanagari ancient document is shown in figure 2. Digitization means converting paper-based document into an electronic form. The electronic conversion is obtained by scanning or by using a digital camera. Bitmap image of the original document is produced in this phase.

4.2 *Image pre-processing*

In image pre-processing, three steps are considered. In the first step, the document image is enhanced using an auto-correct feature of Office image viewer. In the second step, the input image is transformed into a binary image. For binarization, the global threshold value is used. If the global threshold value does not give satisfactory results, then the local threshold value is used in the third step.

4.3 *Segmentation*

Segmentation phase is used to partition the input document into lines, words and characters.

4.3a *Line segmentation*: For line segmentation, piecewise projection profile is used. Devanagari ancient documents have a slant and touching/overlapping lines. Hence, authors have considered piecewise projection profiles to segment lines. In this method, the document image is partitioned into vertical strips and then piecewise horizontal projection profile (HPP) is used to segment lines. After this, average line height was used to check the correctness of segmentation. Based on average line height, some lines were found to be over-segmented and some lines were found to be under-segmented.

A general algorithm for line segmentation is given here.

Algorithm: for line_segmentation

Step 1: Divide the document image into fixed size vertical stripes.
Step 2: Calculate horizontal projection profile (HPP) in each row of each stripe.

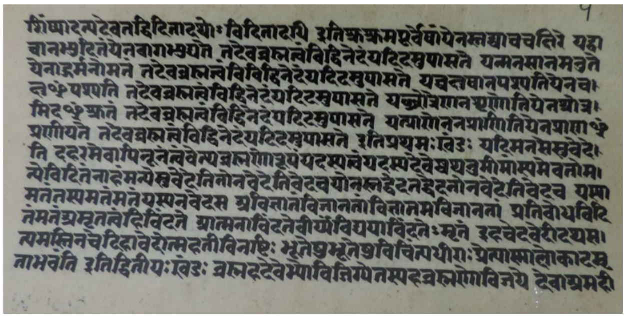

५
शिष्यादन्यदेवतद्विदितादथो॰विदितादधि इतिशुश्रुमपूर्वेषांयेनस्तद्व्याचचक्षिरे यद्वा
चानभ्युदितंयेनवागभ्युद्यते तदेवब्रह्मत्वंविद्धिनेदंयदिदमुपासते यन्मनसानमनुते
येनाहुर्मनोमतं तदेवब्रह्मत्वंविद्धिनेदंयदिदमुपासते यच्चक्षुषानपश्यतियेनच
क्षूंषिपश्यति तदेवब्रह्मत्वंविद्धिनेदंयदिदमुपासते यच्छ्रोत्रेणनशृणोतियेनश्रोत्र
मिदंश्रुतं तदेवब्रह्मत्वंविद्धिनेदंयदिदमुपासते यत्प्राणेननप्राणितियेनप्राणः
प्रणीयते तदेवब्रह्मत्वंविद्धिनेदंयदिदमुपासते इतिप्रथमःखंडः यदिमन्यसेसुवेदे
ति दह्रमेवापिनूनंत्वंवेत्थब्रह्मणोरूपंयदस्यत्वंयदस्यदेवेष्वथनुमीमांस्यमेवतेम
न्येविदितंनाहंमन्येसुवेदेतिनोनवेदेतिवेदच योनस्तद्वेदतद्वेदनोनवेदेतिवेदच यस्या
मतंतस्यमतंमतंयस्यनवेदसः अविज्ञातंविजानतांविज्ञातमविजानतां प्रतिबोधविदि
तंमतंअमृतत्वंहिविंदते आत्मनाविंदतेवीर्यंविद्ययाविंदतेऽमृतं इहचेदवेदीदथस
त्यमस्तिनचेदिहावेदीन्महतीविनष्टिः भूतेषुभूतेषुविचित्यधीराः प्रेत्यास्माल्लोकादमृ
ताभवंति इतिद्वितीयःखंडः ब्रह्महदेवेभ्योविजिग्येतस्यहब्रह्मणोविजये देवाअमही

Figure 2. A sample of Devanagari ancient document.

Step 3: If HPP<=threshold value for any row, then that row is considered as piece-wise separating line (PSL).
Step 4: Consecutive PSLs are reduced to one PSL only.
Step 5: Average line height (avg_line_height) is computed.
Step 6: Based on avg_line_height, over-segmentation is detected and handled.
Step 7: Based on avg_line_height, under-segmentation is detected and handled.
Step 8: Finally, the lines are separated.

4.3b *Word segmentation*: For word segmentation, vertical projection profiles (VPP) are used. If the number of consecutive columns with VPP=0 is greater than a given threshold value, then this is considered as a word boundary.
4.3c *Character segmentation*: It is very difficult to segment characters from a word because (i) neighbouring characters in a word may touch and (ii) neighbouring characters may not touch each other but they can overlap. Character segmentation in Devanagari documents becomes very easy if the headline (*Shirorekha*) is removed, but ancient Devanagari documents have a thick and uneven headline. Hence, it is very difficult to remove the headline from such documents. Characters are segmented without removing headline. Characters of the document image are segmented in multiple iterations. In the first iteration, connected components in the document image are found. Due to the writing style of ancient documents, many times, most of the characters have been segmented correctly as in most of the ancient documents, there is a slight break in the headline after every character as shown in figure 3.

To find touching/overlapping characters, the aspect ratio of all the connected components was found. If the aspect ratio of any component is greater than the threshold value, it is identified as having touching characters.
4.3d *Character normalization*: Initially, all the segmented images of Devanagari ancient documents are normalized into 64 × 64 using Nearest Neighbourhood Interpolation (NNI) algorithm.

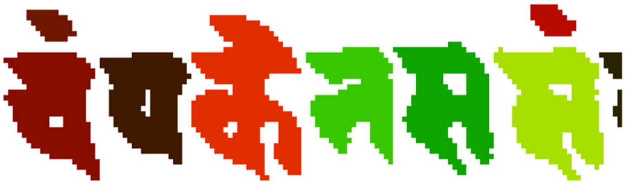


Figure 3. Characters already segmented because of writing style.

4.4 *Feature extraction*

In this work, authors have extracted four types of statistical features, namely, intersection and open endpoints, centroid, horizontal peak extent and vertical peak extent, for Devanagari ancient character recognition. Kumar *et al* [19] extracted various types of features for offline handwritten Gurmukhi character recognition and they presented a study of these features with different classifiers. They concluded that intersection and open endpoints, centroid, horizontal peak extent and vertical peak extent features perform better than other techniques for handwritten text recognition. A pixel that has more than one pixels in its neighbourhood is known as intersection point. A pixel that has only one pixel in its neighbourhood is known as open endpoint. The centroid is the point that can be considered as the centre of a two-dimensional image. In this work, authors have considered the centroid of foreground pixels in each zone of a character image as features. In horizontal peak extent based features, a character image is divided into 85 zones. Then the sum of successive black pixels in each row of a zone is computed. Maximum values (peak values) in each row of a zone are added. The resulting sum is the required feature value of that zone. Authors get a feature set of length 85 for 85 zones. Similarly, in features based on vertical peak extent the sum of successive foreground pixels in the vertical direction in each column of a zone is computed. Hence, authors considered these four techniques in the present work. These features are calculated on different zones of a character image of 64 × 64 size. Authors used the hierarchical zoning method to obtain zones in an image. For hierarchical zoning, first of all, the entire image is considered as one zone and features are calculated for this zone. After this, this image is divided into 4 zones of size 32 × 32 each and features of these 4 zones are calculated. Next, these 4 zones are further divided into 16 zones of size 16 × 16 each as depicted in figure 4. Features are calculated for these 16 zones. These 16 zones are further divided into 64 zones of size 8 × 8 each and features are calculated for these 64 zones. Hence, a feature vector of 85 (1 + 4 + 16 + 64) zones has been extracted.

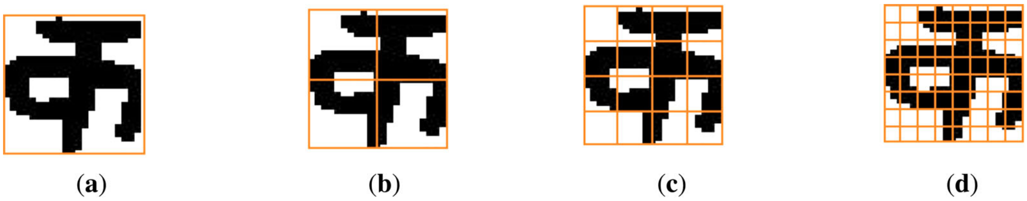


Figure 4. A character image: **a** one zone, **b** 4 zones, **c** 16 zones and **d** 64 zones.

4.5 *Classification*

Classification techniques are going to be described in this section. Features extracted in the feature extraction stage are used for classification. Classification decides the class membership. Classification makes decisions in a character recognition system. For classification task, MLP (C1), Neural Network (C2), CNNs (C3), RBF-SVM (C4) and random forest (C5) are considered. Finally, results are computed by simple majority voting scheme with all the classification techniques considered in this paper. Following subsections briefly describe the classifiers used in the present work.

4.5a *Multilayer perceptron*: MLP is a feed-forward neural network. It has one or more layers between the input and output layer. In a feed-forward, data flows from input layer to the output layer (forward). Back propagation learning algorithm is used to train such network. Multi-Layer Perceptrons are used to solve non-linearly separable problems.

4.5b *ANNs*: ANNs can be considered as structure that contain adaptive simple processing elements called artificial neurons, which are tightly inter-connected. These artificial neurons have the capacity to perform extensively parallel computations for data processing and knowledge representation [22]. Neural networks are typically layered structures. There are interconnected nodes in each layer. Nodes contain an activation function. Input layer receives input data, the output layer gives classification results and others are hidden layers present between the input and output layer. Connections between nodes have weights that are modified according to the input and output provided. Input is passed into the first layer. Individual neurons receive the inputs, with each of them receiving a specific value. After this, an output is produced based on these values. The outputs of the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is predefined. The network, in effect, trains itself.

4.5c *CNN*: CNN is a supervised deep learning algorithm that is trained using the back-propagation algorithm. CNNs can withdraw features automatically. CNN is utilized to learn complex, high-dimensional data, and diverge on the basis of investigation of convolutional and sub-sampling layers [20]. CNNs are considered as the hierarchical architecture of MLPs where the succeeding alternating layers are designed. These layers are designed to learn successively higher-level features, and the classification results are produced by the last layer [21]. The two basic operations, namely, convolution and sub-sampling, are provided by the alternating layers of CNNs.

4.5d *SVM*: SVM is a widely used technique for classification. SVMs are used for supervised learning. They are based on statistics. SVM classification is used to assign data to various classes. Authors have used SVM classifier for the same data set with linear, polynomial and RBF kernels. However, SVM with RBF kernel performs better than other kernels. Hence, in the present paper, authors have considered only RBF kernel of SVM.

4.5e *Random forests*: A random forest can be considered as a collection of decision trees. Decision trees are white box models, which implies that the inner workings of these models are clearly understood. In the case of classification, the data are segregated based on a series of questions. Any new data point is assigned to the selected leaf node. In a decision tree, start at the root of tree and use the decision algorithm to split the data on the feature, resulting in the largest information gain (IG). This splitting procedure is then repeated in an iterative process at each child node until the leaves are classes. This means that the data samples at each node belong to the same class. From the training set, a sample of size *n* is drawn randomly. A decision tree from the bootstrap sample is grown. Grow '*k*' such trees using a subset of samples. Aggregate the prediction of each tree for a new data point and use majority vote (pick the group selected by the most number of trees and assign new data point to that group) to assign the class label.

5. Experimental results and discussion

In this work, authors have considered 6152 characters (34-class problem), which are segmented from Devanagari ancient manuscripts collected from various libraries and museums. In recognition of vowels and consonants of ancient Devanagari manuscripts, Intersection and open endpoint features (F1), Centroid features (F2), Horizontal peak extent features (F3), Vertical peak extent features (F4) and all of their possible combinations in serial mode are also used to improve recognition accuracy. For classification, MLP (C1), Neural Network (C2), CNN (C3), RBF-SVM (C4) and random forest (C5) are used. CNN and random forest perform better than other classification techniques considered in this work, because CNN can extract topological properties of an image and they are learnt with a version of the back-propagation algorithm. They can recognize patterns with extreme variability. Random forest classifier achieves the best recognition accuracy because initially it does efficient feature selection for classification. It then builds trees based on good features and favours these trees over other trees that are built based on noisy features. Experimental results are obtained using 5-fold cross-validation technique. In the MLP classifier, learning rate is set to 0.3 and the momentum is set to 0.2. In CNN classifier, authors have taken the patch size as 3 × 3 and pool size as 2 × 2. For F1 feature set, the accuracy achieved is 39.59%. For F2 features, accuracy is slightly improved and an accuracy of 66.95% is achieved. For F3 set of features, achieved recognition accuracy is 74.64% and for F4 feature set, accuracy is 60.78%. Different combinations of these features have been experimented to

Table 3. Confusion matrix based on a combination of all features (F1 + F2 + F3 + F4) and majority voting scheme.

Character	Total samples	Accurately recognized	Confused with characters						
अ	56	27	च(4)	ह(4)	क(10)	म (1)	प(3)	व (7)	
इ	21	2	द(2)	ह(6)	र(9)	व(2)			
ए	30	13	ह(3)	र(7)	स(5)	य(2)			
क	277	268	च(1)	ह(3)	ज(2)	म(1)	र(1)	त(1)	
ग	56	28	च(1)	क(2)	म(17)	न(6)	र(2)		
घ	16	0	अ(1)	ध(1)	क(1)	स(1)	य(12)		
च	148	74	ह(1)	म(1)	न(2)	त(1)	व(69)		
छ	12	2	ध(1)	ह(5)	प(1)	थ(1)	व(1)	य(1)	
ज	89	51	क(2)	ल(2)	म(2)	न(11)	त(21)		
ट	59	46	द(3)	ह(4)	म(1)	र(2)	स(1)	त(2)	
ड	42	17	द(3)	ह(16)	र(3)	त(3)			
ण	25	7	ए(1)	ह(5)	क(4)	ल(2)	र(1)	स(5)	
त	705	668	ह(4)	Õ(3)	ल(3)	म(4)	न(18)	र(5)	
थ	84	40	ह(2)	क(1)	प(10)	त(1)	व(3)	य(27)	
द	202	142	ह(13)	न(1)	प(2)	र(15)	ट(2)	व(25)	य(2)
ध	80	33	ह(1)	प(11)	र(2)	थ(1)	व(8)	य(24)	
न	498	437	ध(1)	क(1)	म(9)	र(4)	त(36)	व(10)	
प	354	292	ध(2)	ह(5)	म(7)	य(48)			
फ	21	1	च(1)	ह(2)	क(17)				
भ	82	31	म(35)	न(7)	स(1)	त(7)	य(1)		
म	594	559	ह(1)	क(1)	न(8)	प(1)	र(4)	स(8)	त(12)
य	410	362	च(10)	द(3)	ध(1)	ह(2)	प(27)	म(5)	
र	445	426	द(3)	ह(3)	न(5)	प(2)	व(6)		
ल	158	106	द(1)	ह(2)	म(4)	न(12)	स(11)	त(22)	
व	693	665	च(5)	द(1)	ह(2)	म(2)	न(6)	र(6)	त(6)
श	23	2	ह(4)	ज(1)	क(5)	म(2)	स(9)		
ष	54	3	ह(1)	म(4)	प(25)	र(3)	व(9)	य(9)	
स	352	304	ह(3)	क(1)	ल(1)	म(28)	व(4)	त(11)	
ह	294	250	द(5)	क(1)	र(7)	स(6)	त(21)	व(4)	
ढ	36	17	द(1)	ह(2)	क(2)	न(1)	स(8)	त(4)	व(1)
त्र	87	40	च(3	ह(6)	ल(1)	म(2)	र(3)	त(3)	व(29)
ज्ञ	17	1	ह(2)	र(1)	स(7)	त(6)			
द्य	38	3	अ(3)	ह(4)	प(6)	त(1)	थ(1)	व(6)	य(14)
श	94	50	ह(1)	म(6)	प(9)	र(25)	स(1)	व(2)	

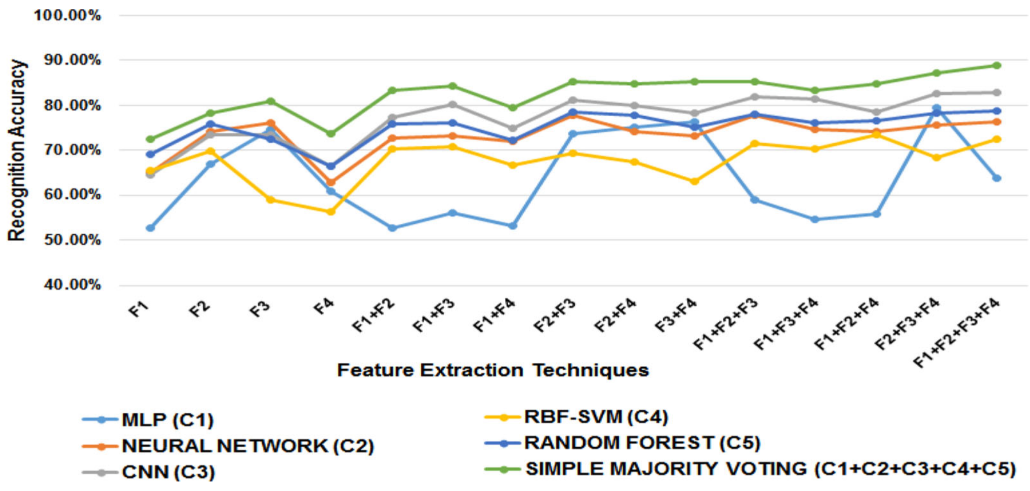


Figure 5. Recognition results using different features and classifiers.

improve the accuracy. Authors have achieved the recognition accuracy of 79.41% using a combination of features F2 + F3 + F4 with MLP classifier. A recognition accuracy of 82.80% using a neural network classifier with a combination of all features (F1 + F2 + F3 + F4) has been achieved. The CNN is the best classifier in the field of computer vision and pattern recognition.

In this work, authors have considered LeNet architecture for CNN. Recognition accuracy of 78.87% has been achieved using a combination of all features (F1 + F2 + F3 + F4) and random forest classifier. Finally, a combination of multiple classifiers considered in this paper by a majority voting scheme to test and improve the accuracy of Devanagari ancient character recognition has been used. Simple majority voting is a decision rule that chooses one of many alternatives. This selection is based on the predicted classes with the majority votes. Once the training of individual classifiers is done, majority voting does not require tuning of any parameter [23]. Using a majority voting scheme of classifiers (C1–C5), the maximum recognition accuracy of 88.95% has been achieved with a combination of all features (F1 + F2 + F3 + F4). Confusion matrix for this case is depicted in table 3. These recognition results are illustrated in figure 5.

Kumar *et al* [24] used diagonal features, centroid features, horizontal peak extent and vertical peak extent features with hierarchical zoning (similar to proposed work) for offline handwritten Gurmukhi character recognition. They experimented with different combinations for features. They used SVM for classification. Table 4 depicts the comparison of their best accuracy on full feature set with present work.

Table 4. Compared with existing work.

Authors	Database	Classifier	Accuracy (%)
Kumar *et al* [24]	Handwritten Gurmukhi text	Linear-SVM	90.08
Proposed work	Devanagari ancient text	Simple majority voting on MLP, NN, CNN, decision tree, random forest	83.92

6. Conclusion and future directions

Authors have presented a character recognition system for the ancient Devanagari documents. Ancient documents were collected from libraries and museums and characters are segmented from these documents. For the recognition of these characters, 4 features are extracted. These features are intersection and open endpoints, centroid, horizontal peak extent and vertical peak extent. All the possible combinations of these features are used. Authors have used five classifiers, namely, MLP (C1), Neural Network (C2), CNN (C3), RBF-SVM (C4) and Random Forest (C5) classifiers. Finally, a simple majority voting scheme with different combinations of these classifiers is used. Maximum accuracy of 88.95% was achieved when all the features were combined and simple majority voting was used for classification. This work considers only basic characters of the Devanagari script. In the future, authors will include modifiers and conjuncts for recognition. Also, more efficient features will be extracted to increase the recognition accuracy and other classification techniques will be experimented.

References

[1] Shah K R and Badgujar D D 2013 Devnagari handwritten character recognition (DHCR) for ancient documents: a review. In: *Proceedings of the 2013 IEEE Conference on Information and Communication Technology*. 656–660

[2] Sarkar R, Malakar S, Das N, Basu S and Nasipuri M 2010 A script independent technique for extraction of characters from handwritten word images. *Int. J. Comput. Appl.* 1(23): 83–88

[3] Kleber F, Sablatnig R, Gau M and Miklas H 2008 Ancient document analysis based on text line extraction. In: *Proceedings of the 19th International Conference on Pattern Recognition*. 1–4

[4] Bansal V and Sinha R M K 2001 A complete OCR for printed Hindi text in Devanagari script. In: *Proceedings of the 6th International Conference on Document Analysis and Recognition*. 800–804

[5] Kim M S, Jang M D, Choi H L, Rhee T H, Kim J H and Kwag H K 2004 Digitalizing scheme of handwritten Hanja historical documents. In: *Proceedings of the First International Workshop on Document Image Analysis for Libraries*. 321–327

[6] Sousa J M C, Pinto J R C, Ribeiro C S and Gil J M 2005 Ancient document recognition using fuzzy methods. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*. 833–836

[7] Cecotti H and Belaid A 2005 Hybrid OCR combination approach complemented by a specialized ICR applied on ancient documents. In: *Proceedings of the 8th International Conference on Document Analysis and Recognition*. 1045–1049

[8] Diem M and Sablatnig R 2009 Recognition of degraded handwritten characters using local features. In: *Proceedings of the 10th International Conference on Document Analysis and Recognition*. 221–225

[9] Raghuraj S, Yadav C S, Verma P and Yadav V 2010 Optical Character Recognition (OCR) for printed Devanagari script using artificial neural network. *Int. J.Computer Science & Communication* 1(1): 91–95

[10] Holambe A N, Thool R C and Jagade S M 2011 A brief review and survey of feature extraction methods for Devnagari OCR. In: *Proceedings of the 9th International Conference on ICT and Knowledge Engineering*. 99–104

[11] Yadav D, Sánchez-Cuadrado S and Morato J 2013 OCR for Hindi language using a neural network approach. *J. Inf. Process. Syst.* 9(1): 117–140

[12] Yunxue S Y, Wang C and Xiao B 2015 A character image restoration method for unconstrained handwritten Chinese character recognition. *Int. J. Doc. Anal. Recognit.* 18(1): 73–86

[13] Katiyar G and Mehfuz S 2016 A hybrid recognition system for off-line handwritten characters. *SpringerPlus* 5: 1–18

[14] Belhe S, Paulzagade C, Deshmukh A, Jetley S and Mehrotra K 2012 Hindi handwritten word recognition using HMM and symbol tree. In: *Proceedings of the Workshop on Document Analysis and Recognition* (*DAR*). 9–14

[15] Lehal G S and Singh C 1999 Feature extraction and classification for OCR of Gurmukhi script. *Vivek* 12(2): 2–12

[16] Kumar M, Sharma R K and Jindal M K 2013 A novel feature extraction technique for offline handwritten Gurmukhi character recognition. *IETE J. Res.* 59(6): 687–692

[17] Kumar M, Sharma R K and Jindal M K 2014 Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. *Natl. Acad. Sci. Lett.* 37(4): 381–391

[18] Kumar M, Sharma R K and Jindal M K 2018 Character and numeral recognition for non-Indic and Indic scripts: a survey. *Artif. Intell. Rev.* https://doi.org/10.1007/s10462-017-9607-x

[19] Kumar M, Jindal M K and Sharma R K 2012 Offline handwritten Gurmukhi character recognition: study of different features and classifiers combinations. In: *Proceedings of the International Workshop on Document Analysis and Recognition*, IIT Bombay. 94–99

[20] Elleuch M, Maalej R and Kherallah M 2016 A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition. *Procedia Computer Science* 80: 1712–1723

[21] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition. *Proc. IEEE* 86(11): 2278–2324

[22] Niu X Y, Xia L Y, Wang T X and Zhang X Y 2010 Application of BP-ANN and LS-SVM to discrimination of rice origin based on trace metals. *Proc. Int. Conf. Mach. Learn. Cybern.* 3: 1426–1430

[23] Zhang Y, Liu B and Yang F 2016 Differential evolution based selective ensemble of extreme learning machine. In: *IEEE Trustcom/Bgdatase/ispa*. 1327–1333

[24] Kumar M, Sharma R K and Jindal M K 2014 A novel hierarchical technique for offline handwritten Gurmukhi character recognition. *Natl. Acad. Sci. Lett.* 37(6): 567–572