Artificial Intelligence Review (2020) 53:2075–2097
https://doi.org/10.1007/s10462-019-09727-2

Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: a study

Munish Kumar[1] · M. K. Jindal[2] · R. K. Sharma[3] · Simpel Rani Jindal[4]

Published online: 11 June 2019

Abstract

Classification is a process to pull out patterns from a number of classes by using various statistical properties and artificial intelligence techniques.The problem of classification is considered as one of the important problems for the development of applications and for efficient data analysis. Based on the learning adaptability and capability to solve complex computations, classifiers are always the best suited for the pattern recognition problems. This paper presents a comparative study of various classifiers and the results achieved for offline handwritten Gurmukhi characters and numerals recognition. Various classifiers used and evaluated in this study include *k*-nearest neighbors, linear-support vector machine (SVM), RBF-SVM, Naive Bayes, decision tree, convolution neural network and random forest classifier. For the experimental work, authors used a balanced data set of 13,000 samples that includes 7000 characters and 6000 numerals. To assess the performance of classifiers, authors have used the Waikato Environment for Knowledge Analysis which is an open source tool for machine learning. The performance is assessed by considering various parameters such as accuracy rate, size of the dataset, time taken to train the model, false acceptance rate, false rejection rate and area under receiver operating characteristic Curve. The paper also highlights the comparison of correctness of tests obtained by applying the selected classifiers. Based on the experimental results, it is clear that classifiers considered in this study have complementary rewards and they should be implemented in a hybrid manner to achieve higher accuracy rates. After executing the experimental work, their comparison and analysis, it is concluded that the Random Forest classifier is performing better than other recently used classifiers for character and numeral recognition of offline handwritten Gurmukhi characters and numerals with the recognition accuracy of 87.9% for 13,000 samples.

Keywords Artificial intelligence · Classification algorithms · Supervised learning · Performance measurement · Comparative studies

✉ Munish Kumar
munishcse@gmail.com

Extended author information available on the last page of the article

1 Introduction

Classification is one of the important step for the document analysis and recognition. In recent years, the machine learning approaches are progressively in demand and receiving great attention by the researchers for the statistical validation of the received outcomes. This can be credited to the development of the range, the expanding number of real life applications and the accessibility of the open machine learning systems that make it simple to propose new algorithms or change the existing ones. In computer vision and pattern recognition fields, various classifiers are generally used for the classification because of their learning adaptability and ability to handle complex situations. The decision about which strategy to use for classifier execution assessment is reliant of many qualities and it is contended that no technique fulfills all the desired requirements. This implies, for some applications, researchers have to utilize more than one classification technique to achieve a reliable assessment. Sometime bad selection of classification methods yields less accurate results, so great care must be given for the selection purpose. Recognition accuracy, training time to build classification model is also depends upon the quality of features and number of classes in the dataset for classification when someone use same classifier for different scripts recognition or different datasets like Gurmukhi script consisting of 56 classes, Devanagari script consisting of 49 classes etc.

Researchers in the area of character/numeral recognition have been presenting lots of work using different classifiers. In this paper, we have evaluated performance of various classifiers for Gurmukhi character/numeral recognition in such a way so that efficient classifier can work for other scripts also with similar structure as that of the Gurmukhi script. Our work progresses by processing the characters and numerals of the dataset using various classification techniques, namely, *k*-NN, Linear-SVM, RBF-SVM, Naïve Bayes, Decision tree, Convolution Neural Network (CNN) and Random forest. The goal is to develop a system that is able to recognize the characters and numerals of Gurmukhi script efficiently with promising accuracy rates. The classification evaluation metrics considered are accuracy, training sample size, False Acceptance Rate (FAR), False Rejection Rate (FRR) and Area Under Receiver Operating Characteristic (AUROC) Curve.

The paper is structured into seven sections. Introduction to the present work has been discussed in Sect. 1. Section 2 presents related work and the collection of the dataset. This section presents the background work of character/numeral recognition and depicts the various methodologies used by different researchers for character recognition. Section 3 focuses on the feature extraction phase used for extracting the properties of character and numeral recognition. Feature extraction is an important phase of an optical character recognition system. In this section, authors have presented a brief introduction about the feature considered in this work. In Sect. 4, authors are focusing the classifiers evaluated in this work. Classification phase is basically used for decided the class membership based on the features extracted from samples. Section 4 presents the detailed introduction and block diagrams of classifiers considered in this work for performance evaluation. Section 5 presents different evaluation metrics. Authors have evaluated the performance of various classifiers based on these performance evaluation metrics. Section 6 depicts experimental work performed using different classifiers. In this section, authors have analyzed the performance of used classifiers for the work based on the parameters such as recognition accuracy, time taken to build training model, False Acceptance Rate (FAR), False Rejection Rate (FRR), Area Under Receiver Operating Characteristics (AUROC) curve. In this section, the authors have finally, presented the performance based on individual features with the best classifier evaluated in this work. Finally, concluding notes and future directions of the present study are presented in Sect. 7.

2 Related work and data set

Literature shows that a good amount of work has already been done on the performance evaluation of a few classifiers for character and numeral recognition. For digit recognition, various methods of feature extraction and classifiers have been studied and compared by Lee and Srihari (1993). The results obtained claimed high accuracy with the chain code feature, the gradient feature, stroke-level, and concavity features (Favata et al. 1994). Jeong et al. (1999) have presented a correlation of different classifiers for digit recognition. For fingerprint and digit recognition, Blue et al. (1994) have analyzed a few classifiers and subsequently by studying the classifiers it was found that there was no problem in the execution of Probabilistic Neural Network (PNN) and the *k*-NN rule. Jain et al. (2000) have presented a study based on little dataset including a digit's dataset. Zhu et al. (1999) differentiated between connected character images and typical images using the Fourier Transform. By comparing Decision Tree, Artificial Neural Network and Logistic Regression, Kim has presented effectiveness of these classifiers based on Root Mean Square Error (Kim 2008). In this article, the impact of the sort of traits and the span of the dataset on the classification methods have been examined and the outcomes have been accounted for regression. Artificial Neural Network (ANN) has been applied to the real and simulated data. These reported results proved that if the data include errors and if the real values of attributes are not available, then the statistical method of regression could act better than the ANN method and produces superior performance. Huang et al. (2003) have taken into consideration Naïve Bayes (NB), Decision Tree (DT) and SVM collectively using Area Under Curve (AUC) paradigm. After applying specified techniques on the genuine information, they noticed that the AUC measure is superior to attaining the precision for comparing the classification methods. Moreover, it was observed that C4.5 execution of the decision tree has a higher Area Under Curve (AUC) as compared to Naive Bayes and SVM. A standout contribution amongst the most cited papers in this area is one by Dietterich (1998). Subsequent to depict the scientific categorization of statistical questions in machine learning, he concentrates on the subject of selecting the algorithm from the two algorithms under consideration, which produces more precise results for a given data collection. Liu et al. (2002) have presented a performance evaluation study in which some efficient classifiers have been used for handwritten digit recognition. They have also indicated that multiple classifiers should be used with great care to acquire high performance.

Kumar et al. (2018) have presented a review for character recognition of non-Indic and Indic scripts. In this review, they have also examined major challenges/issues for character/numeral recognition. Sharma et al. (2009) have expounded a method to rectify the recognition results of handwritten and machine printed Gurmukhi OCR systems. Sharma and Lehal (2009) have proposed an algorithm for removal of the field frame boundary of the hand-filled forms in Gurmukhi script. Sharma and Jhajj (2010) have extracted zoning features for handwritten Gurmukhi character recognition. They have employed two classifiers, namely, *k*-NN and SVM in their work. They could achieve a maximum recognition accuracy of 72.5% and 72.0%, respectively, with *k*-NN and SVM classifiers. Kumar et al. (2013a) have presented a novel feature extraction technique for offline handwritten Gurmukhi character recognition. They have also presented efficient feature extraction techniques based on the curvature features for offline handwritten Gurmukhi character recognition (Kumar et al. 2014a). Table 1 contains some of the studies that have used existing features and classifiers for character and numeral recognition.

For the experimental work in this paper, we have used a balanced primary dataset. This data set consists of 13,000 handwritten samples of 45 classes (7000 samples of handwritten Gurmukhi characters for 35-class problem and 6000 samples of handwritten numerals for 10-class problem). Dataset of characters (7000 samples) is a collection of 35 classes and each class contains 200 samples. Dataset of 6000 samples is a collection of 10 classes and each class contains 600 samples.

Kumar et al. (2013b) have noticed that irrespective of the features, few classifiers perform consistently better if the number of samples in the training data set are increased. Therefore, for experimental work, data set is divided using different partitioning strategies for training dataset and testing dataset as presented in Table 2.

Partitioning Strategy *f* and *g* presents the standard k-fold cross validation. In general, k-fold cross validation divides, complete data set for each category into k equal subsets. Then one subset is taken as testing data and the remaining k-1 subsets are taken as training data. By cross validation, each sample of training data is also predicted and it gives the percentage of correctly recognized testing dataset.

3 Feature extraction

For evaluating the performance of a recognition system, the feature extraction plays an important role. The essential logic behind the feature extraction stage is to extract important properties of a digitized character image, which boosts the recognition accuracy. In this work, at first Nearest Neighborhood Interpolation (NNI) technique has been used to change the digitized images into a size of 88 × 88. A feature vector of 105 elements is extracted by using a hierarchical technique, this feature vector comprises of horizontally and vertically peak extent features (Kumar et al. 2012), diagonal features (Kumar et al. 2012), and centroid features (Kumar et al. 2014b).

3.1 Peak extent based features

In this technique, features are extracted by taking into account the sum of the peak extents, that fit successive black pixels along each zone. Peak extent based features can be extracted horizontally and vertically. In the horizontal peak extent features, they considered the sum of the peak extents that fit successive black pixels horizontally in each row of a zone, whereas in vertical peak extent features they considered the sum of the peak extents that fit successive black pixels vertically in each column of a zone. So, using this technique, authors have obtained $2n$ features corresponding to each character.

3.2 Diagonal features

In this technique, authors have divided the original thinned image of a character into n number of proportionate evaluated zones. These features are taken out by moving along diagonals of the pixels of each zone. Each zone has $2n - 1$ diagonals and ON (foreground) pixels activated along each diagonal are computed up in order to acquire a single

Table 1 Studies on numeral and character recognition

Author	Script	Features	Classifier	Accuracy (%)
Desai (2010)	Gujarati	Projection profiles	Feed forward neural network	81.7
Bhowmik et al. (2004)	Bangla	Stroke	MLP	84.3
Rampalli and Ramakrishnan (2011)	Kannada	Transitions, projection profiles	SVM	87.7
John et al. (2007)	Malayalam	Wavelet transform	MLP	73.8
Lajish (2007)	Malayalam	Fuzzy zoning	Class modular NN	78.9
Raju (2008)	Malayalam	Wavelet	MLP	81.3
Shanthi and Duraiswamy (2010)	Tamil	Pixel density	SVM	82.0
Sundaram and Ramakrishnan (2008)	Tamil	2-D PCA global features	Modified Mahalanobis distance measure	83.4
Lehal et al. (2001)	Gurmukhi	Zoning, local features and global features	Binary decision tree and NN	97.0
Sharma et al. (2008)	Gurmukhi	Elastic matching	*k*-Means	87.4
Sharma and Jhajj (2010)	Gurmukhi	Zoning	SVM	72.0
Jindal et al. (2008)	Gurmukhi	Structural features	SVM	92.5
Kumar et al. (2013a)	Gurmukhi	Peak extent based features	SVM	95.6
Kumar et al. (2014a)	Gurmukhi	Hierarchical Features	SVM	91.8

Table 2 Data set partitioning strategies

Partitioning strategy	Training data	Testing data
a	50%	50%
b	60%	40%
c	70%	30%
d	80%	20%
e	90%	10%
f	Fivefold cross validation	
g	Tenfold cross validation	

sub-feature. These $2n - 1$ sub-features values are averaged to form a single value and put into comparing zone as its feature. Here, we will get n features relating to each sample.

3.3 Centroid feature

For centroid feature extraction, divide the bitmap image into n number of zones. After that, find the coordinates of foreground pixels in each zone and calculate the centroid of these foreground pixels and store the coordinates of these foreground pixels as a feature value. Corresponding to the zones that do not have a foreground pixel, take the feature value as zero. Using this methodology, authors have achieved 2n features elements for each character image.

4 List of classifiers employed for the experimental work

4.1 Convolution neural network (CNN)

Convolutional Neural Network (CNN) or ConvNet is a special kind of multi-layer neural network that is the most suitable classifier in the ground of pattern recognition. In 1990, LeCun and Bengio introduced the concept of CNNs (1990). CNNs are made up of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product and optionally follows it with non-linearity. The whole network expresses a single differentiable score function from the raw image pixels on one end to class score at the other end and they have a loss function (e.g. Softmax) on the last (fully-connected) layer. CNN is a feed-forward network that can extract topological properties of an image and they are learned with a version of the back-propagation algorithm. They can recognize patterns with extreme variability (such as handwritten characters). Block diagram of CNN classification process for numeral recognition is illustrated in Fig. 1.

4.1.1 Layers used to build CNN

CNN is a sequence of layers and every layer of CNN transforms one volume of activations to another through a differentiable function. There are three main types of layers to build CNN architecture, which are convolutional layer, pooling layer and fully-connected layer. The description of these layers are:

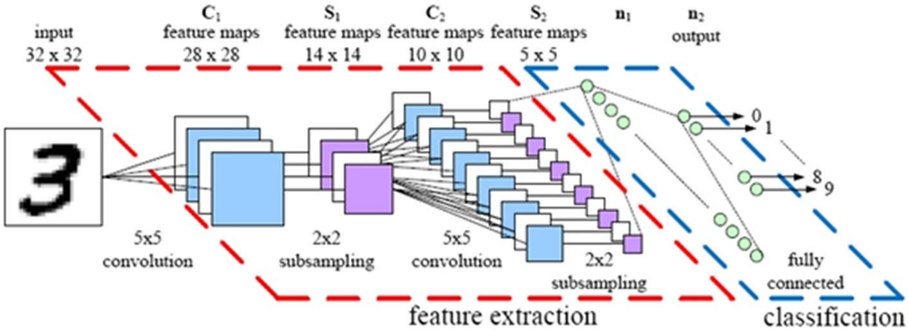

Fig. 1 Block diagram of CNN classification

- Convolutional layer is the core building block of CNN that does most of the computational heavy lifting.
- The pooling layer is placed between successive Convolutional layers of CNN architecture. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control over-fitting. The pooling layer operates independently on each depth slice of the input and resizes it spatially, using the MAX operation.
- In fully-connected layer, neurons have full connection to all activations in the previous layer. Their activations can be computed with a matrix multiplication followed by a bias offset.

There are several architectures available, which are helping in the working of CNN. These are:

- *LeNet* The first successful application of CNNs was developed by LeCun and Bengio in 1990s and the best known is the LeNet (1998) architecture that was used to read zip codes, digits etc.
- *AlexNet* The first work that popularized Convolutional Networks in Computer Vision was the AlexNet (Krizhevsky et al. 2012). The AlexNet was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up (top 5 errors of 16% compared to runner-up with 26% error).
- *ZFNet* The ILSVRC 2013 winner was a Convolutional Network from Matthew Zeiler and Rob Fergus that became known as the ZFNet (Zeiler and Fergus 2014). It was an improvement on AlexNet by modifying the architecture, hyper-parameters, in particular by expanding the size of the convolutional middle layers and making the stride and filter size.
- *GoogLeNet* The ILSVRC 2014 winner was a Convolutional Network from Szegedy et al. (2015) from Google. Its main contribution was the development of an inception module that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M).
- *VGGNet* The runner-up in ILSVRC 2014 was the network from Simonyan and Zisserman that became known as the VGGNet (2015). Its main contribution was in showing that the depth of the network is a critical component for good performance.

- *ResNet* Residual Network developed by He et al. (2016) was the winner of ILSVRC 2015. Its features include special skip connection and a heavy use of batch normalization. ResNet architecture is also missing fully-connected layers at the end of the network.

It is observed that a lot of findings and studies have been presented in the field of pattern recognition using Convolutional neural network. For example, Yuan et al. (2012) have applied CNNs for offline handwritten English character recognition and used modified LeNet-5 CNN model. Liu et al. (2013) proposed a hybrid model with a combination of CNN and Conditional Random Field (CRF) for handwritten English character recognition. CNN is used as a trainable topology-sensitive hierarchical feature extractor and CRF is trained to model the dependency between characters. Anil et al. (2015) have used LeNet-5, CNN is trained with gradient based learning and back propagation algorithm for the recognition of Malayalam characters. Wu et al. (2014) proposed a handwritten Chinese character recognition method based on the relaxation Convolutional Neural Network (R-CNN) and Alternately Trained Relaxation Convolutional Neural Network (ATR-CNN). In this paper, they have used LeNet (the First successful application of Convolution Networks) of CNN for script classification with dropout rate = 0.2, patch size = 3 × 3, pool width and height 2. CNN achieved the third rank among the top seven supervised learning algorithm for handwritten character and numeral recognition work considered in the present paper.

4.2 Decision tree

Various attributes of the data are used by the decision tree algorithm for processing and decision making. Attributes in the decision tree are nodes and each leaf node is representing a classification. Decision tree is a type of supervised machine learning algorithms where the data is continuously divided according to certain parameters. Block diagram of decision tree classification for fruit classification is illustrated in Fig. 2.

The decision tree classifiers organized a series of test questions and conditions in a tree structure. In the decision tree, the root and internal nodes contain attribute test conditions to separate records that have different characteristics. All the terminal nodes are given class labels, Yes or No. After construction of the decision tree, the classification of the test record starts from the root node and then apply the test condition to the record and follow the

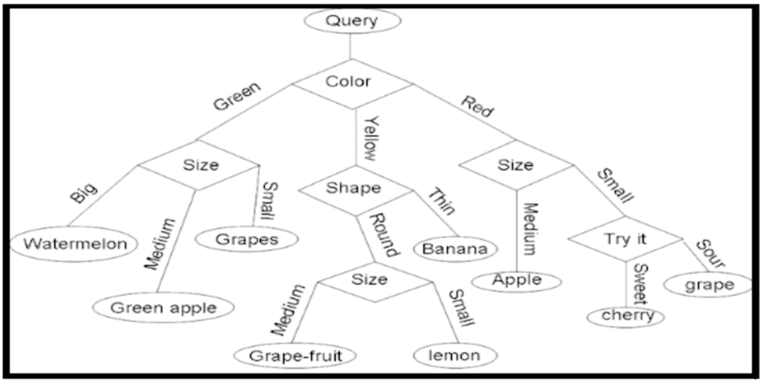

Fig. 2 Block diagram of decision tree classification

appropriate branch based on the outcome of the test. It then leads to either another internal node, for which a new test condition is applied, or a leaf node. When the leaf node is reached, the class label associated with the leaf node is then assigned to the record. The building of an optimal decision tree is the key problem in the decision tree classifier. Various efficient algorithms have been developed to construct a reasonably accurate decision tree in a reasonable amount of time. These algorithms usually employed a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example, Hunt's algorithm, ID3, C4.5, CART, SPRINT are greedy decision tree induction algorithms. Few finding and related work in the field of character recognition or pattern recognition based on the decision tree algorithm are discussed in this section. For example, Amin and Singh (1998) have presented a new technique for the recognition of hand-printed Chinese characters using the Decision trees/C4.5 machine learning system. Sastry et al. (2010) have proposed a system to identify and classify Telugu characters extracted from the palm leaves, using a decision tree approach. Ramanan et al. (2015) proposed a novel hybrid decision tree for printed Tamil character recognition using Directed Acyclic Graph (DAG) and Unbalanced Decision Tree (UDT) classifiers. As per a comparative study of different classification methods presented in this paper for character/numeral recognition, decision tree got the fifth rank among the top seven supervised learning algorithms for character/numeral recognition.

4.3 *k*-NN

k-NN is considered as a lazy learning algorithm that classifies the data sets based on their similarity with the neighbors. Here *k* stands for the number of dataset items that are considered for the classification. A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its *k* nearest neighbors measured by a distance function. If $k=1$, then the case is simply assigned to the class of its nearest neighbor. Usually Euclidean distance is used for calculating the distance between stored feature vector and candidate feature vector in *k*-Nearest Neighbor algorithm. Block diagram of k-NN classifier is depicted in Fig. 3.

For the given attributes,

$$A = \{X1, X2, \ldots, XD\},$$

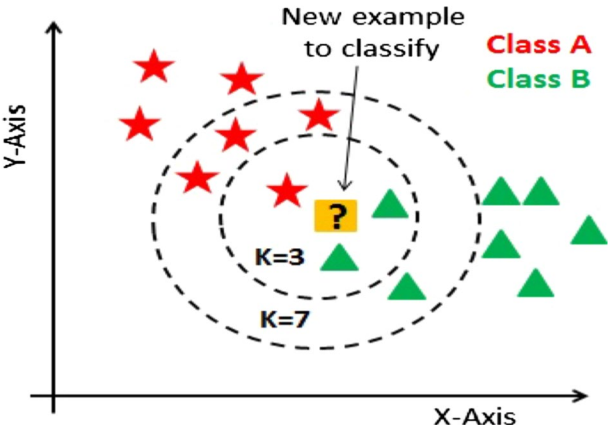

Fig. 3 Block diagram of *k*-NN classification

where D is the dimension of the data, we need to predict the corresponding classification group,

$$G = \{Y1, Y2, \ldots, Yn\}$$

using the proximity metric over *k* items in D dimension that defines the closeness of the association such that $X \in R^D$ and $Yp \in G$.

We choose the optimal value of *k* by first inspecting the data. In general, a large *k* value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to determine a good *k* value by using an independent dataset to validate the *k* value. Rathi et al. (2012) proposed an approach to the recognition of offline handwritten Devanagari vowels by means of *k*-NN classifier and achieved a recognition rate of 96.1%. Rashad and Semary (2014) have developed a system for isolated printed Arabic character recognition using *k*-NN and Random Forest classifiers. Hazra et al. (2017) have presented an application of pattern recognition using *k*-NN to recognize handwritten or printed text. Elakkiya et al. (2017) have developed a system for offline handwritten Tamil character recognition using *k*-NN. *k*-NN is a method for classifying characters/numerals in view of neighboring samples in the training feature space. This classifier got the 4th rank among the seven classification algorithms for character/numeral recognition experimented in this paper.

4.4 Naive Bayes

The Naive Bayes (John and Langley 1995) classifier is a basic method, which has a very clear semantics representing a probabilistic knowledge. This classifier is simple or naive with important and simple assumptions. It expects that in a given class, predicative quality is restrictively autonomous. It also assumes that the prediction process is not influenced by any hidden or latent attributes. Naive Bayes classifier is a family of probabilistic algorithms that takes advantage of probability theory and Bayes' theorem to predict the category of a sample. It is particularly suited when the dimensionality of the input is high. This algorithm is probabilistic, which means that it calculates the probability of each category for a given sample, and then output the category with the highest probability. These probabilities can be achieved by using Bayes' theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature. Naive Bayes classifier assumes that all the features are not related to each other. The presence or absence of a feature does not influence the presence or absence of any other feature. It also assumes that each feature is given the same weight or importance. This method achieved the sixth rank in the seven algorithms for recognition of handwritten characters and numerals considered in this study.

4.5 Random forest

The ensemble for supervised learning method is called the Random Forest (RF) method. Random forest removes 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. Block diagram of random forest classifier is shown in Fig. 4 The random forest uses averaging that helps in improving prescient exactness and control over-fitting. Random forest is unexcelled in accuracy among other existing supervised learning

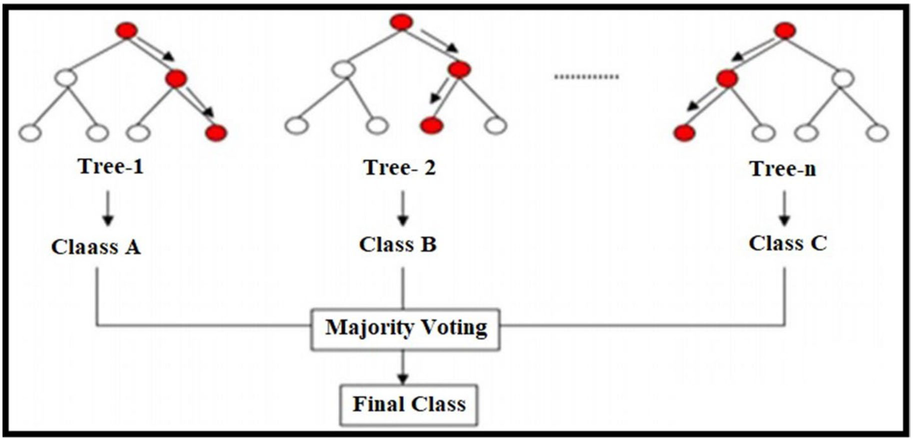

Fig. 4 Block diagram of random forest classification

algorithms for classification and runs efficiently on large databases (Breiman 2001). Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It then aggregates the votes from different decision trees to decide the final class of the test object. Alternatively, the random forest can apply the weight concept for considering the impact of the result of any decision tree. Tree with a high error rate is given low weight value and vice versa. This would increase the decisive impact of trees with low error rate. The basic parameters to random forest classifiers can be the total number of trees to be generated and decision tree related parameters like minimum split, split criteria etc. Random Forest classifier consists of a collection of tree-structured classifiers $\{h(x, \Theta k), k=1, \ldots\}$, where the Θk are independently, identically distributed random trees and each tree casts a unit vote for the final classification of input x. Like CART, Random Forest uses the Gini index for determining the final class of each tree. The Gini index of node impurity is the most commonly useful for classification-type problems.

Homenda and Lesinski (2011) have presented a study on influence of features selection techniques for effectiveness of different classifiers. Their experimental results show that random forest classifier achieves the better results as compared to other methods. Zahedi and Eslami (2012) have discussed the use of the Random Forest classifier in the field of Persian handwritten character recognition. Cordella et al. (2014) have proposed an experimental study of Random Forest classifier reliability in handwritten character recognition using two real world datasets, namely NIST and PD datasets. Rachidi and Mahani (2017) have presented a system of automatic recognition of Amazigh characters using Random Forest method for images obtained by camera phone. Random Forest is the best classification algorithm for character and numeral recognition among the top seven algorithms considered in this paper. 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 those trees over other trees that are built based on noisy features.

4.6 Support vector machine (SVM)

SVM is a supervised learning algorithm for the classification of both linear and non-linear data. It maps the genuine data in large dimensions from where it can find a hyper-plane for the division of the data using imperative training samples called as support vectors. Block diagram of SVM classifier is shown in Fig. 5. A hyper-plane is a "decision boundary" that splits one class from another (Han and Kamber 2001). Using support vectors and margins defined by the support vectors, the SVM locates the hyper-plane. In this work, the authors have considered SVM with linear kernel, namely linear-SVM and SVM with RBF kernel, namely RBF-SVM for classification. Kernel parameter for RBF-SVM is considered as $\gamma = 0.01$ and $c = 1$. The random state value is taken as zero in both kernels (Linear-SVM and RBF-SVM). Linear-SVM achieved the second rank and RBF-SVM achieved seventh rank in the seven supervised learning algorithms for recognition of offline handwritten Gurmukhi characters and numerals in this work.

5 Performance metrics

The performance of the classifiers has been measured with respect to different performance metrics like training sample size, recognition accuracy, False Acceptance Rate (FAR), False Rejection Rate (FRR) and Area Under Receiver Operating Characteristic (AUROC) Curve. The False Acceptance Rate (FAR) is the measure of the probability that the recognition system will inaccurately recognize test information dataset. FAR represents the proportion of the quantity of false acknowledgments partitioned by the aggregate number of mistaken examples. Similarly, False Rejection Rate (FRR) is the measure of the probability that the recognition system will mistakenly dismiss test information. Mutual relationship between FAR and FRR is shown in Fig. 6. FAR and FRR can be calculated as follows.

$$FAR = \frac{Wrongly\ accepted\ samples}{Total\ number\ of\ wrong\ samples}$$

$$FRR = \frac{Wrongly\ rejected\ samples}{Total\ number\ of\ correct\ samples}$$

Fig. 5 Block diagram of support vector machine classification

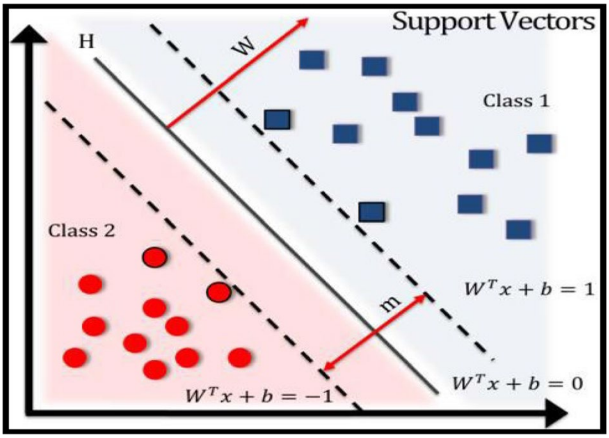

Fig. 6 Relationship between FAR and FRR

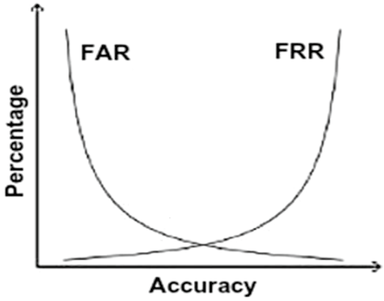

Table 3 Time taken to build training model (in seconds)

Classification technique	Data set partitioning strategy						
	a	*b*	*c*	*d*	*e*	*f*	*g*
CNN	1764.65	1104.34	1117.92	1067.41	1224.91	1124.13	1500.59
Decision tree	8.48	7.14	7.05	7.13	7.04	7.02	7.17
k-NN	0.01	0.01	0.01	0.01	0.01	0.01	0.04
Linear-SVM	30.93	33.89	34.41	28.46	33.96	30.32	30.65
Naïve Bayes	0.28	0.29	0.31	0.29	0.28	0.3	0.41
Random forest	30.34	32.24	29.44	32.08	31.56	29.13	33.2
RBF-SVM	88.8	75.3	73.44	67.85	88.95	73.31	75.28

Table 4 Recognition accuracy achieved using the classifiers

Classification technique	Data set partitioning strategies						
	a (%)	*b* (%)	*c* (%)	*d* (%)	*e* (%)	*f* (%)	*g* (%)
CNN	71.9	72.1	73.2	75.0	75.2	74.6	75.4
Decision tree	63.9	65.6	68.3	69.1	70.6	69.2	70.5
k-NN	68.1	70.5	71.9	73.7	73.8	73.5	74.7
Linear-SVM	79.0	80.7	82.0	81.1	82.0	82.0	82.5
Naïve Bayes	63.7	63.6	64.8	65.1	65.9	65.1	66.3
Random forest	83.4	85.1	86.6	86.8	87.2	87.0	87.9
RBF-SVM	59.7	62.3	63.7	64.6	62.9	64.7	64.9

Area Under Receiver Operating Characteristic (AUROC) Curve is used in classification analysis in order to determine which of the used models predicts the classes best. The classifiers considered in this work are trained with a variable number of samples as discussed in Table 2. We have additionally presented a performance metric of these classifiers in the light of time taken to assemble the model (Table 3). Recognition accuracies accomplished using different classification methods considered in this work are depicted in Table 4.

6 Experimental results

In this section, the authors have presented experimental results of the assessment study for the Convolution Neural Network (CNN), decision tree, *k*-NN, Linear-SVM, Naïve Bayes, RBF-SVM and random forest classifiers. A dataset of 13,000 samples for experimental results (7000 characters and 6000 numerals) has been considered for experimental work. The authors used a variable number of training samples to train the seven classifiers as discussed in Table 2. Time taken to train the proposed model is presented in Table 3. As shown in Table 3, one can see that *k*-NN classifier is taking minimum time when compared with other classifiers for the training of the model.

In Table 4, we have presented recognition accuracies achieved with different classifiers for offline handwritten Gurmukhi characters and numeral recognition. The recognition accuracy achieved with various classifiers is graphically plotted in Fig. 7. As depicted in Table 4 and Fig. 7, the recognition accuracies of 87.9%, 82.5%, 75.4%, 74.7%, 70.5%, 66.3%, and 64.9% has been achieved with Random Forest, Linear-SVM, CNN, *k*-NN, Decision Tree, Naïve Bayes and RBF-SVM classifiers, respectively.

The FAR, FRR and AUROC values of the seven classifiers considered in this work are depicted in Tables 5, 6 and 7 and graphically plotted in Figs. 8, 9 and 10, respectively.

Authors have also calculated one of the most widely used loss function is mean squared error for all classifiers considered in this study, which calculates the square of difference between actual value and predicted value. MSE values of the seven classifiers considered in this work are depicted in Table 8 and graphically plotted in Fig. 11, respectively.

Comparing the results based on recognition accuracy, we can see that the recognition accuracy achieved by the Random Forest classifier is noticeably higher than the other classifiers considered in this work. It has also been noticed that FAR, FRR, MSE and AUROC values of Random Forest classifier are also comparable to other classifier as depicted in Tables 5, 6, 7 and 8. Recognition results of individual features with Random Forest classifier and tenfold cross validation methodology are depicted in Table 9. These features

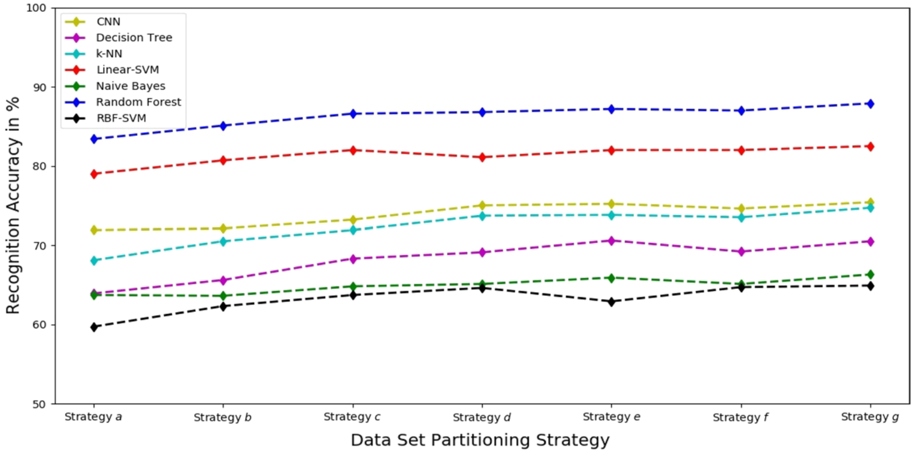

Fig. 7 Recognition accuracies attained using evaluated classifiers

Table 5 False acceptance rate (FAR) for the classifiers

Classification technique	Data set partitioning strategies						
	a (%)	*b* (%)	*c* (%)	*d* (%)	*e* (%)	*f* (%)	*g* (%)
CNN	0.7	0.7	0.7	0.7	0.6	0.6	0.6
Decision tree	0.9	0.8	0.8	0.7	0.7	0.8	0.7
k-NN	0.8	0.7	0.7	0.6	0.6	0.7	0.6
Linear-SVM	0.5	0.5	0.4	0.5	0.4	0.5	0.4
Naïve Bayes	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Random forest	0.5	0.5	0.4	0.4	0.4	0.4	0.4
RBF-SVM	1.3	1.2	1.1	1.1	1.1	1.1	1.0

Table 6 False rejection rate (FRR) for the classifiers

Classification technique	Data set partitioning strategies						
	a (%)	*b* (%)	*c* (%)	*d* (%)	*e* (%)	*f* (%)	*g* (%)
CNN	28.1	27.9	26.8	25.0	24.9	25.4	24.7
Decision tree	36.1	34.4	31.7	30.9	29.4	30.8	29.5
k-NN	31.9	29.5	28.1	26.3	26.2	26.5	25.3
Linear-SVM	21.0	19.3	18.0	18.9	18.0	18.0	17.5
Naïve Bayes	36.3	36.5	36.3	35.4	37.1	35.3	35.1
Random forest	16.7	14.9	13.4	13.2	12.8	12.9	12.0
RBF-SVM	40.3	37.7	35.2	34.9	34.1	34.9	33.7

Table 7 Area under receiver operating characteristic (AUROC) curve for the classifiers

Classification technique	Data set partitioning strategies						
	a	*b*	*c*	*d*	*e*	*f*	*g*
CNN	0.980	0.985	0.988	0.986	0.988	0.987	0.988
Decision tree	0.834	0.844	0.858	0.856	0.871	0.861	0.868
k-NN	0.844	0.856	0.864	0.874	0.877	0.872	0.879
Linear-SVM	0.892	0.901	0.908	0.903	0.908	0.908	0.910
Naïve Bayes	0.968	0.970	0.971	0.970	0.971	0.969	0.969
Random forest	0.993	0.994	0.994	0.994	0.994	0.995	0.995
RBF-SVM	0.792	0.805	0.792	0.820	0.824	0.820	0.826

are performing well for Gurmukhi character recognition (Sundaram and Ramakrishnan 2008; Kumar et al. 2012, 2013b, 2014b). These features are also useful for other types of scripts, which are structurally akin to the Gurmukhi script. As depicted in Table 9, recognition accuracy of 87.9%, FAR of 0.4%, and FRR of 12.0%, has been attained. Confusion

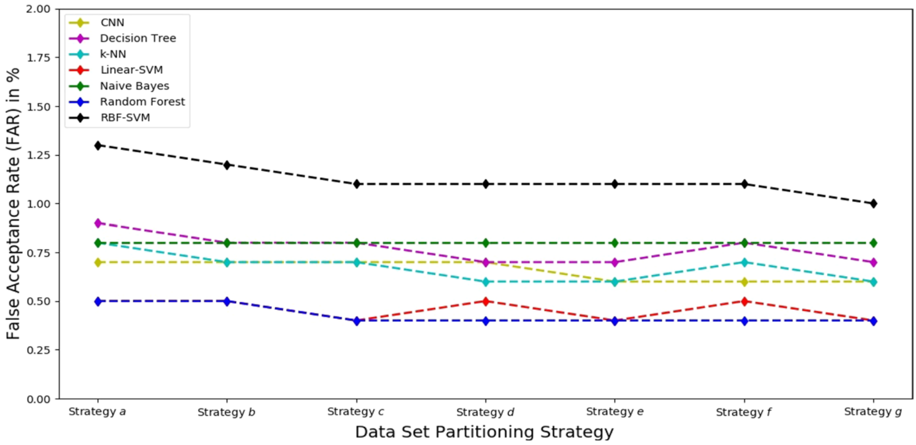

Fig. 8 False acceptance rate (FAR) for the classifiers

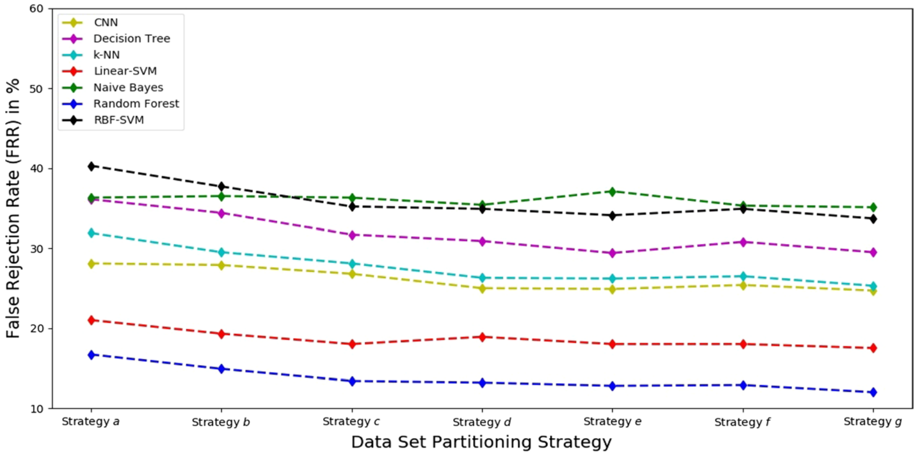

Fig. 9 False rejection rate (FRR) for the classifiers

matrix of this case using random forest classifier and tenfold cross validation is depicted in Table 10.

7 Inferences and observations

For developing successful applications under document analysis and recognition, many directions and alternatives are possible for selecting feature extraction, and classification methods in order to improve the recognition accuracy. Number of researchers has proposed feature extraction/selection techniques and classification techniques for the

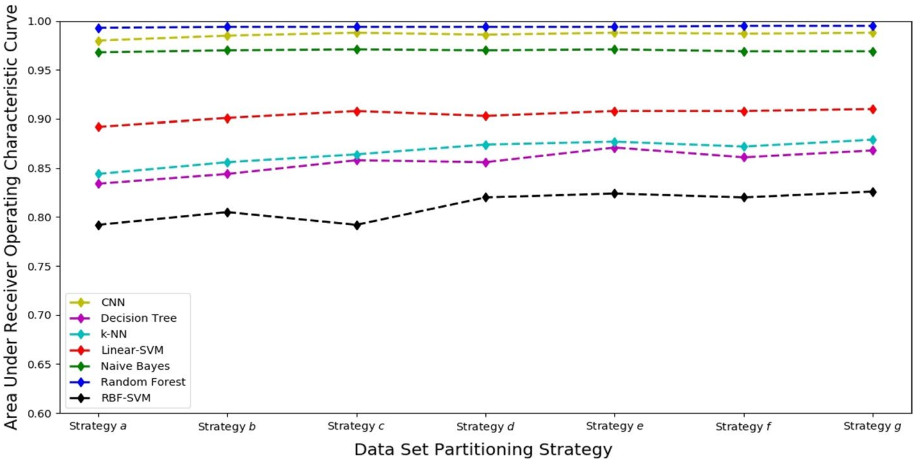

Fig. 10 Area under receiver operating characteristic (AUROC) curve for the classifiers

Table 8 Mean squared error (MSE) for the classifiers

Classification technique	Data set partitioning strategies						
	a	*b*	*c*	*d*	*e*	*f*	*g*
CNN	0.0094	0.0091	0.0091	0.0088	0.0086	0.0087	0.0086
Decision tree	0.0144	0.0138	0.0127	0.0125	0.0118	0.0124	0.0118
k-NN	0.0144	0.0133	0.0126	0.0118	0.0118	0.0119	0.0114
Linear-SVM	0.0095	0.0087	0.0081	0.0085	0.0081	0.0081	0.0085
Naïve Bayes	0.0151	0.0152	0.0150	0.0148	0.0155	0.0147	0.0147
Random forest	0.0085	0.0080	0.0078	0.0076	0.0076	0.0074	0.0076
RBF-SVM	0.0183	0.0171	0.0160	0.0158	0.0155	0.0158	0.0155

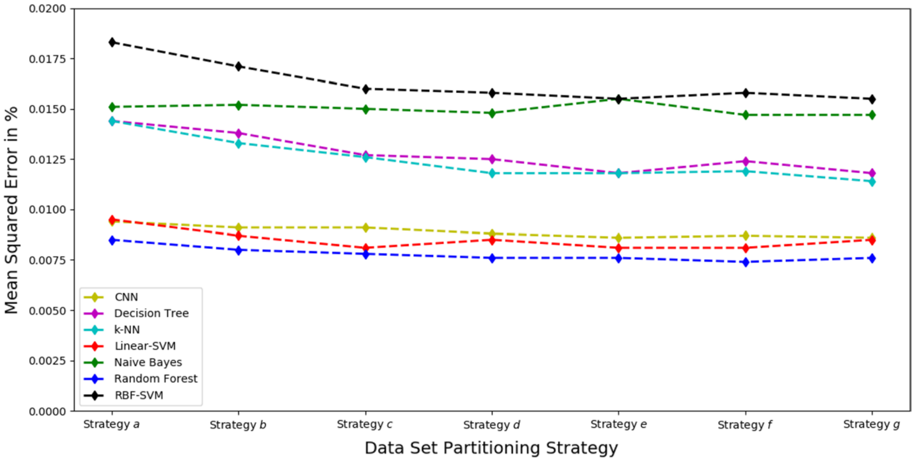

Fig. 11 Mean squared error for the classifier

Table 9 Performance based on individual features and random forest classifier

Features	Recognition accuracy (%)	Training time	FAR (%)	FRR (%)	AUROC	MSE
Horizontally peak extent	85.7	22.2	0.6	13.7	0.977	0.0082
Vertically peak extent	84.9	22.8	0.5	14.6	0.955	0.0081
Diagonal	79.8	26.2	0.7	19.5	0.990	0.0088
Centroid	76.5	24.8	0.6	22.9	0.994	0.0087
Hybrid methodology	87.9	33.2	0.4	12.0	0.995	0.0076

different scripts. In this paper, the authors have focused on the comparative analysis of the classifiers for offline handwritten Gurmukhi character and numeral recognition. This study provides an abstract view for potential readers towards the classification techniques for document analysis and recognition in Gurmukhi script. It is worth mentioning here that by increasing the size of the training dataset, the classification accuracy is generally improved. Authors have selected seven classifiers, namely, Convolution Neural Network, decision tree, *k*-NN, Linear-SVM, Naïve Bayes, RBF-SVM and Random Forest for the character and numeral recognition in this work. These classifiers required moderate memory space and computation cost and provided reasonably high accuracy. After comparing the results based on recognition accuracies, FAR, FRR and AUROC, MSE, authors observed that the Random Forest classifier is performing better than other classifiers for offline handwritten Gurmukhi character and numeral recognition. Researchers can take the new direction of introducing a novel feature extraction and classification method giving higher accuracy rates. One can also look for the tuning and optimizing techniques for the classification algorithms to make sure that the large training set will not cause over fitting problem and achieve higher recognition accuracy.

Table 10 Confusion matrix of random forest classifier with tenfold cross validation

Total characters/numerals		Recognized samples	Confused with characters/numerals								
Gurmukhi characters	ੳ (200)	ੳ (185)	ਅ (2)	ਖ (2)	ਚ (1)	ਢ (2)	ਫ (3)	ਰ (2)	0 (1)	8 (2)		
	ਅ (200)	ਅ (184)	ਸ (2)	ਖ (3)	ਗ (4)	ਙ (3)	ਮ (2)	ਕ (1)	9 (1)			
	ੲ (200)	ੲ (169)	ਙ (4)	ਚ (2)	ਟ (6)	ਠ (5)	ਦ (4)	ਣ (5)	8 (5)			
	ਸ (200)	ਸ (171)	ਗ (3)	ਨ (4)	ਪ (5)	ਮ (3)	ਯ (4)	ਲ (1)	ੜ (2)	4 (4)	8 (3)	
	ਹ (200)	ਹ (177)	ਙ (6)	ਬ (4)	ਰ (4)	0 (2)	2 (3)	3 (1)	6 (3)			
	ਕ (200)	ਕ (174)	ਢ (3)	ਜ (2)	ਨ (4)	ਰ (6)	ਞ (4)	2 (2)	6 (3)	8 (2)		
	ਖ (200)	ਖ (170)	ੳ (2)	ਅ (4)	ਸ (3)	ਕ (2)	ਘ (5)	ਙ (4)	ਪ (6)	ਥ (4)		
	ਗ (200)	ਗ (181)	ਸ (5)	ਮ (4)	ਕ (3)	ਢ (2)	ਠ (2)	ਡ (3)				
	ਘ (200)	ਘ (183)	ਅ (4)	ਸ (2)	ਖ (2)	ਙ (1)	ਜ (2)	ਧ (1)	ਪ (3)	4 (2)		
	ਙ (200)	ਙ (176)	ਅ (4)	ੲ (2)	ਸ (2)	ਕ (1)	ਖ (3)	ਚ (3)	ਜ (2)	ਝ (4)	ਡ (3)	
	ਚ (200)	ਚ (172)	ੳ (2)	ੲ (2)	ਕ (2)	ਙ (3)	ਜ (5)	ਠ (4)	ਡ (3)	ਢ (4)	ਥ (3)	
	ਛ (200)	ਛ (174)	ੳ (3)	ਙ (3)	ਚ (2)	ਠ (1)	ਡ (4)	ਢ (3)	ਫ (3)	5 (2)	6 (3)	8 (2)
	ਜ (200)	ਜ (175)	ਕ (2)	ਧ (4)	ਨ (3)	ਮ (5)	ਲ (2)	ਥ (5)	ੜ (4)			
	ਝ (200)	ਝ (176)	ੳ (4)	ਅ (2)	ੲ (3)	ਕ (2)	ਖ (2)	ਙ (4)	ਢ (2)	2 (3)	3 (2)	
	ਞ (200)	ਞ (174)	ਸ (2)	ਖ (2)	ਗ (3)	ਢ (4)	ਞ (7)	ੜ (4)	3 (4)			
	ਟ (200)	ਟ (180)	ੲ (2)	ਠ (3)	ਡ (3)	ਦ (2)	ਰ (4)	ਨ (2)	5 (2)	6 (1)	8 (1)	
	ਠ (200)	ਠ (173)	ਹ (6)	ਡ (2)	ਨ (3)	ਫ (4)	ਰ (4)	3 (1)	5 (2)	6 (2)	8 (3)	
	ਡ (200)	ਡ (179)	ਝ (2)	ਠ (2)	ਚ (3)	ਢ (2)	ਦ (4)	ਫ (3)	ਯ (2)	ਰ (3)		
	ਢ (200)	ਢ (173)	ੳ (2)	ਙ (3)	ਚ (6)	ਦ (5)	ਫ (3)	ਭ (1)	ਯ (1)	ਰ (3)	5 (2)	6 (1)
	ਣ (200)	ਣ (166)	ੲ (5)	ਙ (4)	ਢ (6)	ਟ (3)	ਠ (3)	ਡ (4)	ਫ (2)	ਭ (5)	2 (1)	5 (1)
	ਤ (200)	ਤ (165)	ਠ (2)	ਡ (7)	ਫ (4)	ਭ (8)	ਰ (3)	3 (2)	5 (4)	8 (3)	9 (2)	
	ਥ (200)	ਥ (159)	ੳ (4)	ੲ (6)	ਖ (8)	ਪ (7)	ਙ (2)	ਚ (4)	ਟ (3)	ਠ (2)	ਧ (5)	

Table 10 (continued)

Total characters/numerals		Recognized samples	Confused with characters/numerals									
	ਦ (200)	ਦ (176)	ਏ (4)	ਕ (3)	ਘ (2)	ਚ (3)	ਟ (2)	ਡ (3)	ਢ (2)	2 (1)	8 (3)	
	ਧ (200)	ਧ (163)	ਸ (5)	ਕ (4)	ਗ (4)	ਘ (3)	ਥ (5)	ਪ (6)	ਬ (3)	ਯ (3)	2 (2)	4 (2)
	ਨ (200)	ਨ (177)	ਸ (3)	ਕ (2)	ਫ (4)	ਥ (3)	ਪ (2)	ਮ (1)	ਲ (5)	4 (1)	6 (2)	
	ਪ (200)	ਪ (173)	ਏ (4)	ਕ (3)	ਙ (1)	ਠ (1)	ਧ (7)	ਨ (2)	ਥ (6)	4 (1)	8 (2)	
	ਫ (200)	ਫ (175)	ੳ (3)	ਏ (4)	ਗ (3)	ਚ (5)	ਧ (4)	ਪ (3)	ਠ (1)	ਨ (2)		
	ਬ (200)	ਬ (174)	ੳ (3)	ਅ (2)	ਏ (3)	ਖ (5)	ਘ (3)	ਧ (4)	ਥ (6)			
	ਭ (200)	ਭ (182)	ੳ (2)	ਅ (2)	ਟ (3)	ਠ (2)	ਤ (4)	3 (2)	8 (2)	9 (1)		
	ਮ (200)	ਮ (173)	ਅ (3)	ਸ (7)	ਖ (3)	ਗ (6)	ਨ (2)	ਰ (3)	4 (3)			
	ਯ (200)	ਯ (179)	ਅ (3)	ਸ (3)	ਘ (4)	ਥ (2)	ਧ (4)	ਪ (2)	ਮ (3)			
	ਰ (200)	ਰ (166)	ੳ (1)	ਹ (9)	ਚ (8)	ਝ (4)	ਡ (3)	ਥ (2)	2 (3)	3 (2)	6 (2)	
	ਲ (200)	ਲ (167)	ਏ (7)	ਸ (4)	ਗ (3)	ਙ (6)	ਢ (2)	ਜ (5)	2 (1)	3 (2)	5 (3)	
	ਵ (200)	ਵ (148)	ਏ (9)	ਵ (12)	ਙ (3)	ਢ (4)	ਝ (6)	ਟ (5)	ਭ (6)	ਰ (3)	5 (2)	6 (2)
	ੜ (200)	ੜ (148)	ੳ (9)	ਕ (2)	ਖ (2)	ਗ (1)	ਙ (5)	ਚ (6)	ਝ (2)	ਡ (8)	ਭ (8)	3 (9)
Numerals	0 (600)	0 (591)	ੳ (3)	2 (1)	5 (1)	8 (2)	9 (2)					
	1 (600)	1 (571)	ਕ (4)	ਟ (3)	5 (4)	6 (7)	7 (5)	9 (6)				
	2 (600)	2 (567)	ਏ (3)	ਕ (4)	ਗ (2)	ਘ (2)	ਙ (5)	ਟ (3)	ਡ (3)	ਥ (4)	ਮ (2)	8 (5)
	3 (600)	3 (555)	ਹ (6)	ਠ (4)	ਡ (8)	ਤ (9)	ਭ (7)	1 (4)	2 (2)	4 (3)	5 (2)	
	4 (600)	4 (569)	ਖ (4)	ਕ (3)	ਧ (6)	ਨ (3)	ਪ (5)	0 (2)	2 (3)	3 (4)	5 (1)	
	5 (600)	5 (567)	ਠ (6)	ਡ (5)	0 (4)	1 (3)	3 (4)	4 (4)	6 (2)	8 (5)		
	6 (600)	6 (582)	ਚ (4)	ਠ (4)	0 (2)	4 (2)	5 (1)	7 (2)	8 (3)			
	7 (600)	7 (578)	ਅ (3)	ਏ (4)	2 (2)	4 (1)	6 (5)	8 (3)	9 (4)			
	8 (600)	8 (567)	ੳ (5)	ਙ (3)	ਚ (4)	1 (2)	2 (3)	3 (5)	6 (4)	7 (3)	9 (4)	
	9 (600)	9 (569)	ਜ (4)	1 (5)	2 (3)	3 (5)	4 (4)	5 (3)	7 (4)	8 (3)		

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest in this work.

References

Amin A, Singh S (1998) Recognition of hand-printed Chinese characters using decision trees/machine learning C4.5 system. Pattern Anal Appl 1(2):130–141

Anil R, Manjusha K, Kumar SS, Soman KP (2015) Convolutional neural networks for the recognition of Malayalam characters. In: Proceedings of the 3rd international conference on frontiers of intelligent computing: theory and applications (FICTA), pp 493–500

Bhowmik TK, Bhattacharya U, Parui SK (2004) Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. In: Proceedings of international conference on neural information processing (ICONIP'04), pp 814–819

Blue JL, Candela GT, Grother PJ, Chellappa R, Wilson CL (1994) Evaluation of pattern classifiers for fingerprint and OCR applications. Pattern Recognit 27(4):485–501

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Cordella LP, Stefano CD, Fontanella F, Freca ASD (2014) Random forest for reliable pre-classification of handwritten characters. In: Proceedings of the 22nd international conference on pattern recognition, pp 1319–1324

Desai AA (2010) Gujarati handwritten numeral optical character reorganization through neural network. Pattern Recognit 43(7):2582–2589

Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10(7):1895–1924

Elakkiya V, Muthumani I, Jegajothi M (2017) Tamil text recognition using KNN classifier. Adv Nat Appl Sci 11(7):41–45

Favata JT, Srikantan G, Srihari SN (1994) Handprinted character/digit recognition using a multiple feature/resolution philosophy. In: Proceedings of 4th international workshop on frontiers of handwriting recognition, pp 57–66

Han J, Kamber M (2001) Data mining concepts and techniques. Morgan Kaufmann Publishers, San Francisco, pp 70–181

Hazra TK, Singh DP, Daga N (2017) Optical character recognition using KNN on custom image dataset. In: Proceedings of the 8th annual conference on industrial automation and electromechanical engineering, pp 110–114

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Homenda W, Lesinski L (2011) Features selection in character recognition with random forest classifier. In: Proceedings of the international conference on computational collective intelligence, pp 93–102

Huang J, Lu J, Ling CX (2003) Comparing Naïve Bayes, decision trees, and SVM with AUC and accuracy. In: Proceedings of the third IEEE international conference on data mining, pp 1–4

Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Jeong SW, Kim SH, Cho WH (1999) Performance comparison of statistical and neural network classifiers in handwritten digits' recognition. In: Lee S-W (ed) Advances in handwriting recognition. World Scientific, Singapore, pp 406–415

Jindal MK, Sharma RK, Lehal GS (2008) Structural features for recognizing degraded printed Gurmukhi script. In: Proceedings of the 5th international conference on information technology: new generations (ITNG), pp 668–673

John GH, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th conference on uncertainty in artificial intelligence, pp 338–345

John R, Raju G, Guru DS (2007) 1D wavelet transform of projection profiles for isolated handwritten Malayalam character recognition. In: Proceedings of international conference on computational intelligence and multimedia applications (ICCIMA), vol 2, pp 481–485

Kim YS (2008) Comparison of the decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size. Expert Syst Appl 34(2):1227–1234

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th international conference on neural information processing, vol 1, pp 1097–1105

Kumar M, Sharma RK, Jindal MK (2012) Offline handwritten Gurmukhi character recognition: study of different features and classifiers combinations. In: Proceedings of international workshop on document analysis and recognition, IIT Bombay, pp 94–99

Kumar M, Sharma RK, Jindal MK (2013a) A novel feature extraction technique for offline handwritten Gurmukhi character recognition. IETE J Res 59(6):687–692

Kumar M, Sharma RK, Jindal MK (2013b) Size of training set vis-a-vis recognition accuracy of handwritten character recognition system. J Emerg Technol Web Intell 5(4):380–384

Kumar M, Sharma RK, Jindal MK (2014a) Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. Natl Acad Sci Lett 37(4):381–391

Kumar M, Jindal MK, Sharma RK (2014b) A novel hierarchical technique for offline handwritten Gurmukhi character recognition. Natl Acad Sci Lett 37(6):567–572

Kumar M, Jindal MK, Sharma RK, Jindal SR (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

Lajish VL (2007) Handwritten character recognition using perceptual fuzzy-zoning and class modular neural networks. In: Proceedings of 4th international conference on innovations in information technology (ICIIT), pp 188–192

LeCun Y, Bengio Y (1990) Handwritten digit recognition with a back-propagation network. In: Proceedings of the advances in neural information processing systems, pp 396–404

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Lee DS, Srihari SN (1993) Handprinted digit recognition: a comparison of algorithms. In: Proceedings of 3rd international workshop on frontiers of handwriting recognition, pp 153–164

Lehal GS, Singh C, Lehal R (2001) A shape based post processor for Gurmukhi OCR. In: Proceedings of the 6th international conference on document analysis and recognition (ICDAR), pp 1105–1109

Liu CL, Sako H, Fujisawa H (2002) Performance evaluation of pattern classifiers for handwritten character recognition. Int J Doc Anal Recognit 4(3):191–204

Liu C, Liu J, Yu F, Huang Y, Chen J (2013) Handwritten character recognition with sequential convolutional neural network. In: Proceedings of the international conference on machine learning and cybernetics, pp 291–296

Rachidi Y, Mahani Z (2017) Handwritten Amazigh character recognition system for image obtained by camera phone. Int J Sci Eng Res 8(3):1319–1324

Raju G (2008) Wavelet transform and projection profiles in handwritten character recognition—a performance analysis. In: Proceedings of international conference on advanced computing and communications, pp 309–314

Ramanan M, Ramanan A, Charles EYA (2015) A hybrid decision tree for printed Tamil character recognition using SVMs. In: Proceedings of the 15th international conference on advances in ICT for emerging regions (ICTer), pp 130–141

Rampalli R, Ramakrishnan AG (2011) Fusion of complementary online and offline strategies for recognition of handwritten Kannada characters. J Univers Comput Sci (JUCS) 17(1):81–93

Rashad M, Semary NA (2014) Isolated printed Arabic character recognition using KNN and random forest tree classifiers. In: Proceedings of the international conference on advanced machine learning technologies and applications, pp 11–17

Rathi R, Pandey RK, Jangid M (2012) Offline handwritten Devanagari vowels recognition using KNN classifier. Int J Comput Appl 49(23):11–16

Sastry PN, Krishnan R, Ram BVS (2010) Classification and identification of Telugu handwritten characters extracted from palm leaves using decision tree approach. ARPN J Eng Appl Sci 5(3):22–32

Shanthi N, Duraiswamy K (2010) A novel SVM based handwritten Tamil character recognition system. Pattern Anal Appl (PAA) 13(2):173–180

Sharma DV, Jhajj P (2010) Recognition of isolated handwritten characters in Gurmukhi script. Int J Comput Appl 4(8):9–17

Sharma DV, Lehal GS (2009) Form field frame boundary removal for form processing system in Gurmukhi script. In: Proceedings of the 10th international conference on document analysis and recognition (ICDAR), pp 256–260

Sharma A, Kumar R, Sharma RK (2008) Online handwritten Gurmukhi character recognition using elastic matching. In: Proceedings of the congress on image and signal processing, pp 391–396

Sharma DV, Lehal GS, Mehta S (2009) Shape encoded post processing of Gurmukhi OCR. In: Proceedings of the 10th international conference on document analysis and recognition (ICDAR), pp 788–792

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proceedings of the international conference on learning representations, pp 1–14

Sundaram S, Ramakrishnan AG (2008) Two dimensional principal component analysis for online character recognition. In: Proceedings of 11th international conference on frontiers in handwriting recognition (ICFHR), pp 88–94

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Computer vision and pattern recognition. arXiv:1409.4842

Wu C, Fan W, He Y, Sun J, Naoi S (2014) Handwritten character recognition by alternately trained relaxation convolutional neural network. In: Proceedings of the 14th international conference on frontiers in handwriting recognition, pp 291–296

Yuan A, Bai G, Jiao L, Liu Y (2012) Offline handwritten English character recognition based on convolutional neural network. In: Proceedings of the 10th IAPR international workshop on document analysis systems, pp 125–129

Zahedi M, Eslami S (2012) Improvement of random forest classifier through localization of Persian handwritten OCR. ACEEE Int J Inf Technol 2(1):13–17

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Proceedings of the European conference on computer vision, pp 818–833

Zhu X, Shi Y, Wang S (1999) A new distinguishing algorithm of connected character images based on Fourier transform. In: Proceedings of 4th international conference on document analysis and recognition, pp 788–791

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

Affiliations

Munish Kumar[1] · M. K. Jindal[2] · R. K. Sharma[3] · Simpel Rani Jindal[4]

M. K. Jindal
manishphd@rediffmail.com

R. K. Sharma
rksharma@thapar.edu

Simpel Rani Jindal
simpel_jindal@rediffmail.com

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

[2] Department of Computer Science and Applications, Panjab University Regional Centre, Muktsar, PB, India

[3] Department of Computer Science and Engineering, Thapar Institute of Engineering & Technology, Patiala, PB, India

[4] Computer Science and Engineering, Yadavindra College of Engineering, Talwandi Sabo, Bathinda, PB, India