JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

The Performance Evaluation of Deep Learning Classifier to Recognize Devanagari Handwritten Characters and Numerical

[1]Anuj Bhardwaj, [2]Prof. (Dr.) Ravendra Singh
[1]Research scholar, Dept. of CS & IT, M.J.P. Rohilkhand University, Bareilly (U.P.), India
[2]Professor, Dept. of CS & IT, M.J.P. Rohilkhand University, Bareilly (U.P.), India
[1]anuj2k3@gmail.com, [2]rsiet2002@gmail.com

Received 2022 March 15; **Revised** 2022 April 20; **Accepted** 2022 May 10.

Abstract
A text classification is a well formed process using various measurable properties and computerized logical procedure to fetch a pattern from different classes.Since classification is important for the pattern recognition process, there are some issues with well-formed classification in this process, which is one of the important issues for proper development and improvement of productive data examinations. On behalf of the versatility of learning and the ability to deal with complex calculations, classifiers are consistently best suited for design patter recognition issues. The aim of this paper is to present a result based comparative study of different classifiers and the optimal recognition of results computation through the Devanagari Handwritten characters and numerical values. Different classifiers were used and evaluated in this investigation including k-Nearest Neighbor (k-NN), Support-Vector machine (SVM), Naïve Bayes, Decision Tree, Random Forest, and Convolution Neural Network (CNN). To accomplish the experiment purpose, this paper used an unbiased dataset with including 123 samples that consists of 123 characters and 123 numerical values. Python 3.0 with sciket learn machine learning open-source environment library have been used to evaluate the performance of the classifiers. The performances of the classifiers accessed by considering the different matrices including dataset volume with best split ratio among training, validation, and testing process, accuracy rate, Ture/False acceptance rate, True/False rejection rate and the area covered under the receiver operating characteristic curve. Similarly the paper shows the correlation of the accuracy of the experiments obtained by applying to chosen the classifier. On behalf of the exploratory results, the infallible classifiers considered in this test have free rewards and must be executed in a hybrid manner to meet the thigh precision rates.In the views of test work, their result compressions and the examination to be performed, it is argued that the Random Forest classifier is performing in a way that the current use of the classifier to recognize the Devanagari Handwritten character and the numerical values with the accuracy rate 87.9% for the considered 123 samples.
Keywords
Devanagari Handwritten Characters, Classification Algorithms, Artificial Intelligence, Machine Learning, Supervised Learning Techniques, Performance Evaluation, Comparative Study

Background

For the process of analysis the reporting and recognition classification is one of the important advancement. During the last few years, the artificial intelligence enabled approaches are constantly sought and are receiving exceptional views by scientists for measurable approval of the results obtained. This can be attributed to improved accessibility, increasing number of real applications, and openness to open machine learning framework that makes it easier to propose new calculations or to change the current one. In the field of computing vision (CV) and the pattern recognition (PR), various classifies are used for characterization due to the learning versatility and the ability to deal with the complex situations. The choice of which method to use for classification performance evaluation depends on the several characteristics and it is assumed that no single strategy meets all ideas requirements. These infers, for some application, require experts to use more than one grouping process to complete a concrete evaluations. In some unfavorable circumstances when the selection of the classification technique calculated less accurate results, unreliable consideration should be given to the reason for the choice. The accuracy of the recognition technique, model training time to the classification process, additionally depends on the number and the nature of the classes in the dataset for arrangement when one uses the same classification for different material of recognitions or for specific datasets such as Devanagari Handwritten that consist of 49 classes, Gurmukhi script with 56 classes, etc.

Experts in the field of character/numerical values recognition are introducing a plethora of work using various classifiers. In this paper, we have assessed the exposition of individual classifiers for Devanagari characters/numerical values recognition, so an efficient classifier in particular can work with a comparative design for different materials such as Devanagari script. Character and numeral values of the script datasets, that uses as different grouping methods to be specific, k-NN, SVM, Naïve Bayes, Decision Tree (DT), Random Forest (RF), and Convention Never Network (CNN). The aim is to create a framework that can effectively realize the characters and numerical values of Devanagari Handwritten script while promising accurate rates. Classification evaluation measurements are believed to be accurate, formulating training sample sizes, false acceptance rate, false rejection rate, and the area under covered by the receiver operating characteristic (AUROC) curve.

Paper is structured in six different sections. In initial section 1 covers the introductory part of the research work. The forwarded section 2 presents the classification techniques related work and the existed datasets. Ultimate aim of this section is to present the foundation work of the character / numerical values recognitions and the illustration of the various methods used by various analysts for recognition of script values. In further section 3 spotlights on the element extraction stage used to separate the character and properties of the points of recognitions. Highlight extraction is a critical time of an optical character recognition framework. In this section, we offer a brief introduction to the component considered in this research work. In the section 4, authors are covering the classifiers evaluated in this work. The classification step is basically used for class enrollment dependent on the highlights extracted from the tests. Section 4 presents the items and presentation of the classifiers considered in this work for performance the evaluation. In section 5 we present specific evaluation measurements. We have assessed the presence of various classifiers dependent on these exhibition evaluation measurements. Section 6 portraits exploratory work used various classifiers. In this part, we break down the exhibition of classifiers used for work that relies on limitations, for example, recognition accuracy, time taken to build a training model, false acceptance rate, false rejection rate, turning area under receiver operating characteristics (AUROC) curve. In this part, the creators have worked for a long time, presenting the performance dependent on the individual features with the best classifiers evaluated in this work. Finally, in the concluding notes and in the view of future directions of the present examination are introduced in the section 7.

Related Work

The literature suggests that a decent measure of work has been done on the exhibition evaluation of a pair of classes for recognition of the character and numerical values. For recognition of the digits, separate techniques are available for extraction and classification were investigated and observed by (Lee, 1993). Results that guarantee high precision with chain codes include modified features, the gradient-features, and the symmetry features(Srikantan & Srihari, 1994). (Jeong et al., 1999) have introduced relationships of various classifiers for digitized recognizant. For specific markings and digit recognition, (Blue et al., 1994) have investigated some classifiers, and thus, the

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

idea of classifiers has shown that there was no issue in the implementation of probabilistic neural networks (PNNs) including the k-NN rule. (Jain et al., 2000) have introduced a probe dependent on a paired dataset, including a digit based dataset. (Zhu et al., 1999)diverged between character drawings and ordinary pictures involving the use of the Fourier transform. Looking at the decision tree, artificial neural networks, and logistic regression, Kim has introduced the adequacy of these classifiers dependent on the root mean square error (Kim, 2008). In this article, the effects of such characteristics and the extent of the dataset on order strategies are analyzed and the results are represented by the regression technique. Simulated artificial neural networks (ANNs) have been applied to real and reproducible information. These detailed results demonstrated that if the information involved mistakes and that on the off chance that actual assessments of symptoms were not accessible, then the factual technique for relapses at the time could work better and prevail than the ANN strategy Dictates execution. (Huang et al., 2003)have considered Naïve Bayes (NB), Decision Tree (DT), and SVM using everything under the Area under Curve (AUC) standard. In view of applying the indicated strategies to the certified data, researcher observed that AUC measurement is better than accomplishing accuracy in contrast to measurement techniques. Furthermore, it was observed that the execution of the C4.5 is based on the choice tree has a high area under curve (AUC) when contrasted with Naive Bayes and SVM. A champion commitment to the most referenced papers around one by (Dietterich, 1998). To illustrate the logical arrangement of measurable inquiry in AI, he focuses on choosing the calculation from two calculations, which produces more accurate results for a given information classification. (C.-L. Liu et al., 2002) have introduced a presentation evaluation concentrator in which some effective classifiers have been used for manually written digit recognitions. Researches have likewise demonstrated that the purchase of elite of various classes should be used with extraordinary consideration.

(Kumar et al., 2019) has introduced a survey for recognition the character of non-Indic and Indic content. In this survey, author has additional oversight of significant difficulties/issues for character/numerical values recognition. (D. V. Sharma & Lehal, 2009) have clarified a technique for the prevention of post-recognition of manually handwritten and machine-printed Gurmukhi OCR systems. (D. V. Sharma et al., 2009) have proposed a calculation for the removal of the field outline boundary of handwritten filled structures in Gurmukhi material. Sharma and (D. Sharma & Jhajj, 2010) set aside the drafting highlights for the manually written Gurmukhi character recognition. Author has used two classifiers in his work, particularly K-NN and SVM. They can meet the most extreme accuracy for the recognition of 72.5% and 72.0% individually, with k-NN and SVM classifiers. (Kumar et al., 2013a) have introduced a novel component extraction method to offline manually handwritten Gurmukhi character values recognition. Likewise they have introduced proficient component extraction methods dependent on curve highlights for offline Gurmukhi character recognition (Kumar, Sharma, et al., 2014). In the table 1 some of the investigation has been listed in which existing highlights and classifiers have been used for characterization and digit recognition.

Author	**Target Script**	**Considered parameters**	**Application classifier**	**Model accuracy rate (%)**
(Lehal et al., 2001)	Gurmukhi	Zoning, local features and global features	Binary decision tree and NN	97
(Bhowmik et al., 2004)	Bangla	Stroke	MLP	84.3

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

(R. 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
(Sundaram & Ramakrishnan, 2008)	Tamil	2-D PCA global features	ModifedMahalanobis distance measure	83.4
(A. Sharma et al., 2008)	Gurmukhi	Elastic matching	k-Means	87.4
(Jindal et al., 2008)	Gurmukhi	Structural features	SVM	92.5
(Desai, 2010)	Gujarati	Projection profles	Feed forward neural network	81.7
(Shanthi & Duraiswamy, 2010)	Tamil	Pixel density	SVM	82
(D. Sharma & Jhajj, 2010)	Gurmukhi	Zoning	SVM	72
(Rampalli & Ramakrishnan, 2011)	Kannada	Transitions, projection profles	SVM	87.7
(Kumar et al., 2013a)	Gurmukhi	Peak extent based features	SVM	95.6
(Kumar, Sharma, et al., 2014)	Gurmukhi	Hierarchical Features	SVM	91.8

Table 1: Past related studies on recognition of langue scripts

Devanagari Handwritten Script- Dataset Description

For exploratory work considered in this research paper, we have used a balanced dataset available publically on open-source repository environment. The considered dataset Devanagari is part of the Brahmic family of scripts of Nepal, India, Tibet, and South-East Asia.(Fischer, 2004)(Gaur, 1992) The script is used to write Nepali, Hindi, Marathi and similar other languages of South and East Asia. The Nepalese writing system adopted from Devanagari script consists of 12 vowels, 36 base forms of consonant, 10 numeral characters and some special characters. Vowel characters are shown in Table 2, consonants characters in Table 3 and numeral characters in Table 4. Moreover, all 36 consonants could be wrapped with the vowels generating 12 other derived forms for each branch of consonant character. One such example for “ta (tabala)” and “pa” is shown in Table 5.

Devanagari Character	अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	ऑ	ऒ
UNICODE	905	906	907	908	909	090A	090F	910	913	914	911	912

Table 2: Devanagari vowels with UNICODE

Devanagari Character	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
UNICODE	915	916	917	918	919	091A	091B	091C	091D	091E	091F
Devanagari Character	ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ
UNICODE	920	921	922	923	924	925	926	927	928	092A	092B
Devanagari Character	ब	भ	म	य	र	ल	व	श	ष	स	ह
UNICODE	092C	092D	092E	092F	930	932	935	936	937	938	939

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

CHARACTER	क्ष	त्र	ज्ञ	These three consonants have no specific UNICODE

Table 3: Devanagari consonants with UNICODE

०	१	२	३	४	५	६	७	८	९
966	967	968	969	096A	096B	096C	096D	096E	096F

Table 4: Devanagari numerals

त	ता	ति	ती	तु	तू	ते	तै	तो	तौ	तं	तः
प	पा	पि	पी	पु	पू	पे	पै	पो	पौ	पं	पः

Table 5: Derived forms of consonant "ta (tabala)" and "pa" when wrapped with vowels.

Devanagari Handwritten Character Dataset is created by collecting the variety of handwritten Devanagari characters from different individuals from diverse fields. Handwritten documents are than scanned and cropped manually for individual characters. Each character sample is 32x32 pixels and the actual character is centered within 28x28 pixels. Padding of 0 valued 2 pixels is done on all four side to make this increment in image size. The images were applied gray-scale conversion. After this the intensity of the images were inverted making the character white on the dark background. To make uniformity in the background for all the images, we suppressed the background to 0 value pixel. Each image is a gray-scale image having background value as 0.

Devanagari Handwritten Character Dataset contains total of 92,000 images with 72,000 images in consonant datasest and 20,000 images in numeral dataset. Handwritten Devanagari consonant character dataset statistics is shown in Table 6 and handwritten Devanagari numeral character dataset statistics is shown in Table 7.

Table 6: Consonant Character Dataset

Devanagari Character (Class)	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
Individual statistics	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000
Devanagari Character (Class)	ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ
Individual statistics	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000
Devanagari Character (Class)	ब	भ	म	य	र	ल	व	श	ष	स	ह
Individual statistics	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000
Devanagari Character (Class)	क्ष	त्र	ज्ञ								
Individual statistics	2,000	2,000	2,000								
Total	**72,000**										

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

Table 7: Numeral Dataset

Devanagari Character (Class)	०	१	२	३	४	५	६	७	८	९
Individual statistics	966	967	968	969	096A	096B	096C	096D	096E	096F
Total	**20,000**									

(Kumar et al., 2013b) have observed that regardless of the highlights, hardly any classifier reliably outperforms if the amount of trials in the training dataset of index expands. In this way, for the test task, the information collection is separated using the training dataset presented in Table 8 and the partitioning systems specific to the test dataset.

Partitioning strategy	**Training data**	**Testing data**
a	50%	50%
b	60%	40%
c	70%	30%
d	80%	20%
e	90%	10%
f	10-Fold cross validation	

Table 8: Dataset splitting strategies

The strengthening strategy f and g introduce standard k-fold cross validation. All in all, k-fold cross-validation is isolated, in the same subset of the total information index for each class. At that point, one subset is taken as test information and the remaining K-1 subset is taken as data for training. By cross-validation, each instance of information generation is additionally anticipated and this effectively levels the perceived test dataset.

Feature extraction process

The recognition system performance is evaluated including the highlight of extraction assumes an important part. The basic logic behind the step of feature extraction is to extracting the essential properties form the digitized character image, which helps in the acceptance of accuracy. In the present work, Nearest Neighborhood Interpol (NNI) strategy has been used from the beginning to convert the digitized images to a size of 32× 32. A feature vector of 105 components has been removed using a different hierarchical process, this feature element vector contains evenly and vertically the top degree features (Kumar et al., 2012), diagonal features(Kumar et al., 2013a), and centroid features (Kumar, Jindal, et al., 2014).

Peak extent based features extraction

With the use of this method, features have been extracted with considering the amount of peak extents, which is fit progressively on the black spotted pixel along with each considered area. The peak extents based feature extraction can be accepted horizontally and vertically. In horizontal feature extraction based peak extents model considered the volume of those fit progressively on the dark pixels on a horizontal plan in each row of a field as the amount of peak extent, although in the vertical feature extraction based peak extent features considered the amount of those spine progressive pixel pins vertically in every segment of the zone. Thus, using this method, the users have achieved 2n features compared to each character.

Centroid pixel based feature extraction technique

The centroid pixel based feature extraction technique is based on the technique the divide the bitmap image into a number of n regions. From that point forward, search the directions of the pixels at the foreground of each region and compute the centroid of these frontal area pixels and store the directions of these closely visible pixels as feature values. As compared to the areas that do not have a pixel at the foreground, take the component with respect to nothing. Using this process, the code developer completed 2n element components for each character image.

Diagonal pixels features extraction technique

In this process, the developer has separated the first reduced image of a character into the number of regions with proportional evaluation. These are featured as pixels of each region move along the diagonals. Each region has 2n - 1 diagonal and ON near-view pixels, which are registered with each diagonal to obtain a solitary sub-features. These 2n - 1 sub-feature esteem fall at the midpoint of the shape of a solitary value and look at the zone as its component. Here, we will be highlighted with each example identified.

Randomization of classifiers used for search operations

Convolutional Neural Network (CNN)

Convolutional Neural Network sometimes known as ConvNet is a unique type of multiple layered neural network architecture that is the most appropriate classifier which is based on the recognition of the patter with consideration deep learning.

In 1990, LeCun and Bengio introduced the idea of CNNs.(LeCun et al., 1990) This deep learning network model includes neurons that have their individual weight and adjustable biases values. Each neuron receives some information, plays a mathematical dot product calculation, and follows it alternately with non-linearity. The entire network communicates a different class score from the raw image pixels in the opposite direction towards the class score on one side and an loss function (e.g. Softmax) on their final (fully connected) layer. CNN is a feed-forward neural network architecture that can extract the topological properties of an image and learn them with optimization for back-propagation algorithm. They may experience design with outrageous variability, (for example, manually written characters). A class graph of the CNN characterization measurements for recognition as outlined in Figure 1.

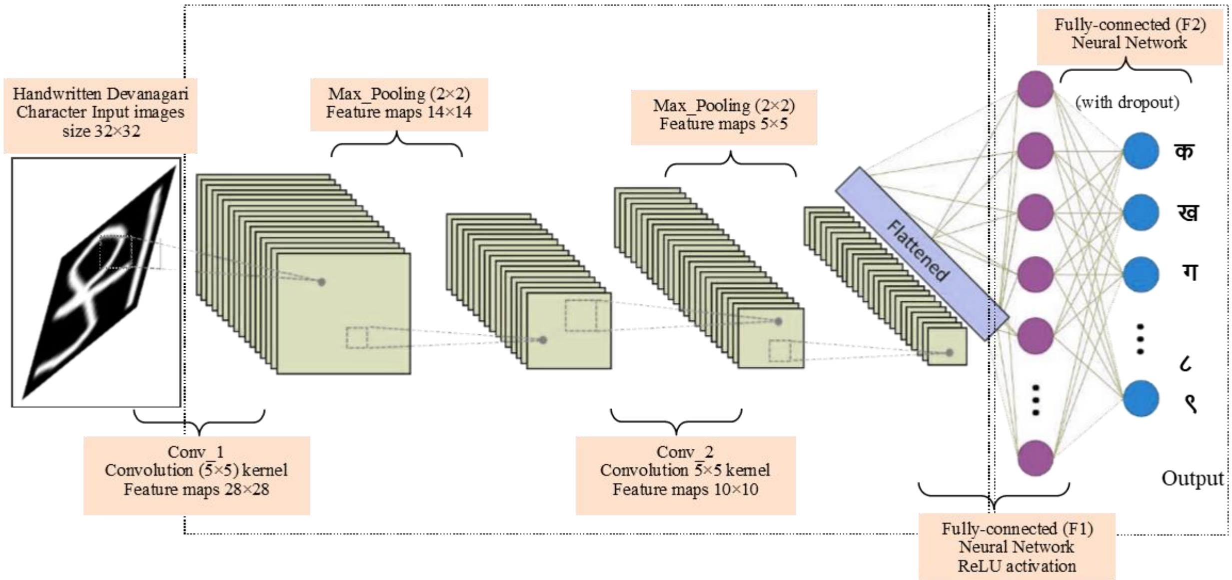


Figure 1: The block diagram of Convolutional Neural Network

CNN model layers Description

The model architecture of CNN uses a sequence of layers and each layer of CNN changes quantities one after the other through different functions. There are three major types of layers to formulate the CNN model architecture, which consist the convolutional layer, the pooling layer, and the

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

fully connected layer. These layers depict as follows;

- Convolutional layer is the center structure block of CNN model which makes the majority of computational work heavy lifting.
- The next layer is the pooling layer which is placed between the successive convolutional layer of CNN model. Its ability is to dynamically reduce the spatial size of the illustration to reduce the number of limitations and calculations in the organization, and over-fit control in a similar way. The pooling layer works independently on every depth slice of the input information and shapes it spatially, using maximum activity.
- In the fully connected layer, neurons have a complete relationship with all actions in the past layer. Activation functions of these layers can be computed using the matrix multiplication formulation followed by using a bias offset.

Some of the structures are accessible, that helps during the working process of CNN's model. These structures as follows;

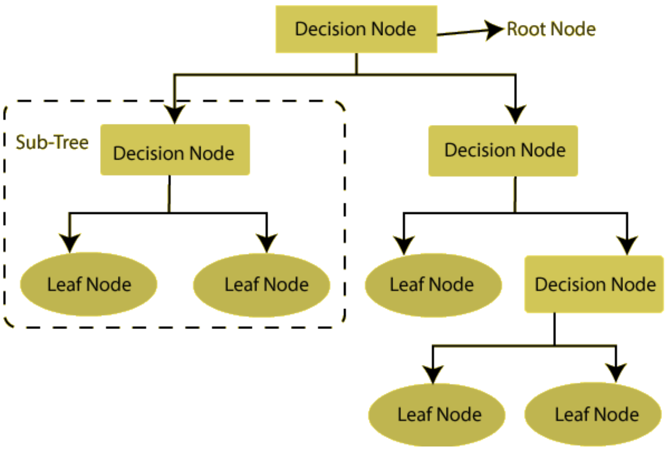
- LeNet structure was first effectively used with the CNNs architecture during the 1990s by LeCun and Bengio and the most popular is the (LeCun et al., 1998) architecture that was used for postal districts, numerals and so on.
- In AlexNet structure Computer Vision is the primary work promoting the Convolutional Network that was Alexnet(Krizhevsky et al., 2017). AlexNet was presented for the ImageNet ILSVRC challenge in 2012 and it was second overall runner-up (top 5 error measurements of 16% with 26% fumble with oddity with the sprinter).
- ZFNet structure ILSVRC 2013 winner was a Convolutional network of Matthew Zeiler and Rob Fergus known as ZFNet(Zeiler & Fergus, 2014). This was an enhancement for AlexNet, particularly by changing the design, hyper-parameters, by increasing the size of the fixture center layers and creating the steps and channel sizes.
- GoogleNet structure ILSVRC 2014 was a convolutional network from (Szegedy et al., 2015) from Google. Its principle commitment was an improvement of a start module that reduced the number of boundaries in the organization (60M to 4M as opposed to AlexNet).
- VGGNet structure ILSVRC was the organization of Sprinter Simonyan and Zisserman in 2014 known as VGGNet(Simonyan & Zisserman, 2014). Its fundamental commitment was in demonstrating that organization proficiency is a fundamental part of great performance.
- ResNet structure is the abbreviation of Residual Network built by (He et al., 2016) was the champion of ILSVRC 2015. Its highlights include an extraordinary skip association and the weighty use of cluster standardization. The ResNet design is likewise missing fully connected layers towards the finish of the organization.

It has been observed that huge amount of discoveries and studies have been introduced into the field of pattern recognition using a convolutional neural network. For example, (Yuan et al., 2012) have implemented CNN to offline manually handwritten English alphabets recognition and use a transformed LeNet-5 CNN model. (C. Liu et al., 2013) proposed a model based on the hybrid technology with a mixture of CNN and Conditional Random Fields (CRF) for the transit model. CNN model is used as a trainable topology-sensitive progressive feature extractor and CRF is formulated to demonstrate dependency between characters. (Anil et al., 2015) have used LeNet-5, CNN for recognition of Malayalam characters using the gradient based learning model and the back-propagation algorithm. Wu et al. (2014) proposed a manually handwritten Chinese character recognition technique that relied on the Relaxation based Convolutional Neural Network (R-CNN) and the Alternately Trained Relaxation Convolutional Neural Network (ATR-CNN) architecture. In the correct research work proposed model have used CNN's LeNet (the first successful use of Convolution Networks) for the considered script characterization with the dropout rate = 0.2, patch size = 3 × 3, width of the pool, and the length of the pool. CNN has secured the third position among the main six machine learning supervised learning algorithms for recognizing the handwritten character and numerical values that considered in the current work covered in the paper.

Decision Tree algorithm

Different descriptions of information are used by preparing and calculation process of the decision tree to computing the well-formed decision. The decision tree has characteristic nodes and each leaf node is addressing an individual class. A decision tree is a type of regulated AI computation where information is continuously separated by specific boundary parameters. The block diagram of the decision tree based characterization for fruit classification orders is presented in Figure 2.

Figure 2: The block diagram of decision tree classification



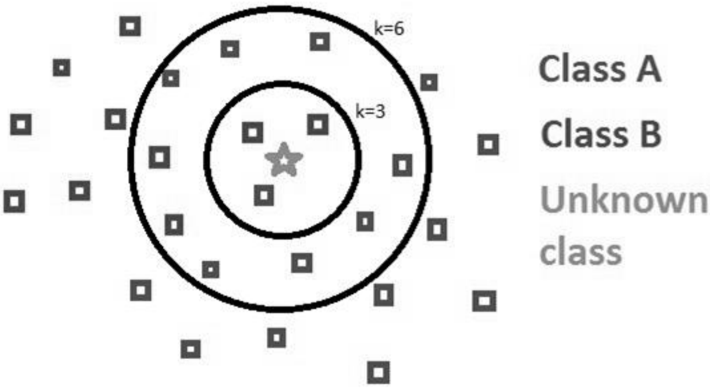
Decision tree based classifier coordinated the progress of test conditions and situations in the structure of a tree. Inside the hierarchy of decision tree, the root and inward hubs have attribute test conditions to separate records with different properties. All terminal nodes are assigned the labels of class, either yes or no. After the development of the decision tree, the group of test records starts at the root nodes and later applies the test conditions to the record and follows the appropriate branch dependent on the result of the experiment. This indicates either another internal nodes at the point to which another test condition applies, or a leaf node. At the point when the leaf node is reached, the class name corresponding to the leaf node is allowed to be recorded. An ideal choice is the important issue in the classification of the tree structure for the decision tree. Various productive algorithms have been made to construct a sensible accurate choice tree in a sensible measure of time. These algorithms typically use a greedy process that grows a decision tree at the progress of locally idealized choices about which quality to use to split the data into a well-defined manner. For example, Hunt's, ID3, C4.5, CART, SPRINT are the algorithms of the solicitation decision tree. The decision tree algorithm is covered in this section to make evidences and related work of examples based on decision that can recognize the character pattern. As an example researcher (Amin & Singh, 1998) have introduced another process for the recognition of hand-written Chinese letters using a machine learning framework i.e. decision-making trees/C4.5. (Sastry et al., 2010) have proposed a framework for identifying and ordering Telugu characters extracted from palm leaves using the structure of decision tree approach. (Ramanan et al., 2015) proposed a novel decision tree approach for recognizing the printed Tamil character using the hybridization of Directed Acyclic Graph (DAG) and the Unbalanced Decision Tree (UDT) classifiers. According to a close examination of the various classification techniques that are introduced in this paper for recognizing the character and numerical value, the decision tree

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452
was ranked fifth best out of six successful in supervised learning algorithms for recognizing the character and numerical values.

k- nearest neighbor (k-NN) classification

k-NN is measured as a sluggish learning algorithm of classification that marks the dataset dependent on their simulation with neighbors. Here k represents the number of items form the dataset that considered for classification. A case is arranged by a major part vote of its neighbors, the case is assigned to the class, which is approximated to the function of the distance normally between its nearest neighbors. In the case that k = 1, at that point the case is considered only to the class of its nearest neighbor. Generally, Euclidean distances are used to find out about the distance between put-away element vectors and the candidate feature vectors in the algorithm of k-nearest neighbor. The block diagram of the K-NN classifier is presented in Figure 3.

Figure 3: A block diagram of the k-NN based classification technique



For the considered attributes in the dataset,

$$A = x_1, x_2, x_3, \dots x_d \qquad (1)$$

In equation (1) d represent the dataset dimension, where we need to predict the value of the corresponding classification group

$$G = y_1, y_2, y_3, \dots, y_n \qquad (2)$$

By the use of adjoining metrics on k items with d dimensions that are characterized by the closeness of affiliation with the end goal that $X \in R^D$, and$Y_P \in G$.

First we select the ideal estimate of k by estimating the information. When all is met, a huge k value is more accurate because it is normally reduced yet there is no assurance. Cross-validation is another approach to fixing a decent k incentive using a free dataset to validate the value of k. (Rathi et al., 2012) proposed a way to deal with the recognition of offline Devanagari Handwritten vowels by methods for the $k - NN$ classifier and meet the rate of recognition 96.1% approximate.(Rashad & Semary, 2014) have designed a system for the printed Arabic character recognition by using $k - NN$, and the Random Forest classifier. (Hazra et al., 2017) have presented an example of recognition by using $k - NN$ to visualize handwritten or printed text. (Elakkiya et al., 2017) have created a system to disconnect manually written Tamil character recognition using $k - NN$. The classification with $k - NN$ is a strategy for organizing the characters and numerical values involved in preparation that includes tests involving preparation. This classifier was ranked fourth among the six distinct algorithm of classification of the recognition for characters and numerical values that covered in this paper with detailed expiation.

Naïve Bayes Classifier

The naïve base (G. H. John & Langley, 2013) classifier is a fundamental technique with very clear semantics addressing a piece of probabilistic information. This classifier is basic or reliable with critical and basic skepticism. It is expected that in the considered class, the quality of the presentation is prohibitively independent. Likewise, the forecast cycle is not affected by any cover or passive attributes. The Naïve Bayes classifier is a

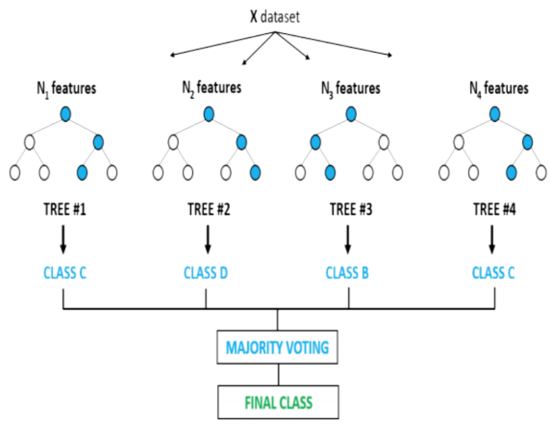
JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

group of probabilistic calculations that exploits the probability hypothesis and Bayes hypothesis to estimate the classification of an instance. This is particularly fit when the dimensionality of the information is high. This classification algorithm is probable, meaning that it detects the probability of every class for the given example, and then derives the classification with the most notable probability. These probabilities can be met using Bayes hypothesis, which reflects the probability of an element, in light of earlier information on terms that can be identified with that highlight. The Innocent Bayes classifier hopes that not all highlights are identified with each other. The presence or the absence of a component does not affect the presence or non-presence of another element. It additionally acknowledges that each element is given equal weight or importance. This strategy ranked sixth among the six classification techniques for the recognition of handwritten characters and numerical values considered in this investigation.

Random forest classifier

The method is known as the Random Forest (RF) classification technique that ensemble with the supervised learning techniques. The over-fitting of the arbitrary decision tree eliminates with the random forest technique. The decision tree classifier is used to sort various sub-instances of the dataset. A metadata assessment that fits the number of tree classifiers preferred for such a scheme is called a random forest. The block diagram of the random forest classifier has appeared in figure 4 that uses a random forest average which helps to over fit perceptual accuracy and control. Random forest is accurately obscured in other currently administered learning algorithm for heterogeneity and run efficiently over large datasets (Breiman, 2001). The random forest classifier forms a bunch of trees of decision from an arbitrarily chosen subset of the preparation set. It totals the votes from different choice trees to choose the last class of test object at that point. On the other hand, random forest can apply weight considerations to consider the after-effects of any tree of decision. A tree with a higher fault rate is given a lower weight value and the other way around. This will create uneven effect of trees with low fault rates. A random forest classifier can have the full number of trees to produce its basic boundaries and select least-partitioned parameters such as tree-related boundaries and therefore tree-organized classifier $\{h\ (x, \Theta k), contains\ a\ classification\ of\ k\ = 1,2,3, \ldots\}$, where Θk are independently, indirectly random forest are planted, and each tree prefers one unit for the last order of input data x. Likewise CART, Random Forest uses a G_{ini} index to decide the last class of each tree. The G_{ini} of node impurity is most helpful for characterization type issues.

Figure 4: A block diagram of random forest



classifier

(Homenda & Lesinski, 2011) have projected an investigation on the adequacy of different classifiers due to highlighted different strategies. Their exploratory results suggest that random forest classifiers provide better results when contrasted with different techniques. (Zahedi &

Eslami, 2012) have investigated the use of random forest classifiers in the field of Persian transit character simulation. (Cordella et al., 2014) have proposed a test investigation of random forest classifier dependence in transcode character recognition, using two real-world datasets, specifically the NIST and PD datasets. (Amrouch et al., 2012) have introduced a system of programmed recognition of Amazigh characters using random forest technology for photos acquired by camera assembled phone. Among the best six calculations considered in this paper is the most appropriate classification technique for characterization and digit recognition. The random forest classifier satisfies the best recognition accuracy in light of the fact that, first; it prefers the productive component to the arrangement. This is at the point that assembles trees dependent on large highlights and favor trees above various trees that rely on features.

Support vector machine (SVM) classifier

The SVM is a machine leaning approach considered under supervised learning algorithm for arranging both linear and non-linear information. It maps authenticated information into large measurements from where it can detect a hyper-plane for the segmentation of information using basic readiness tests called support vector. A block diagram of SVM classifiers appears in the figure 5. The hyper-plane is a "boundary of decision" that distinguishes one class from another (Han et al., 2011). Featuring support vectors and edge-driven classifier, SVM finds the hyper-plane. In this work, the creators have thought of SVM as a linear kernel, specifically the direct SVM, and the RBF kernel with SVM, especially for characterization in RBF-SVM. The kernel threshold for RBF-SVM is assumed to be and $\gamma = 0.01$, and $c = 1$. The irregular state respect is taken as zero in two parts (linear-SVM and RBF-SVM). Linear SVM fulfilled the latter condition and RBF-SVM ranked worst with compared all six supervised learning algorithm for acceptance of Devanagari Handwritten characters and numeral values recognition in this work.

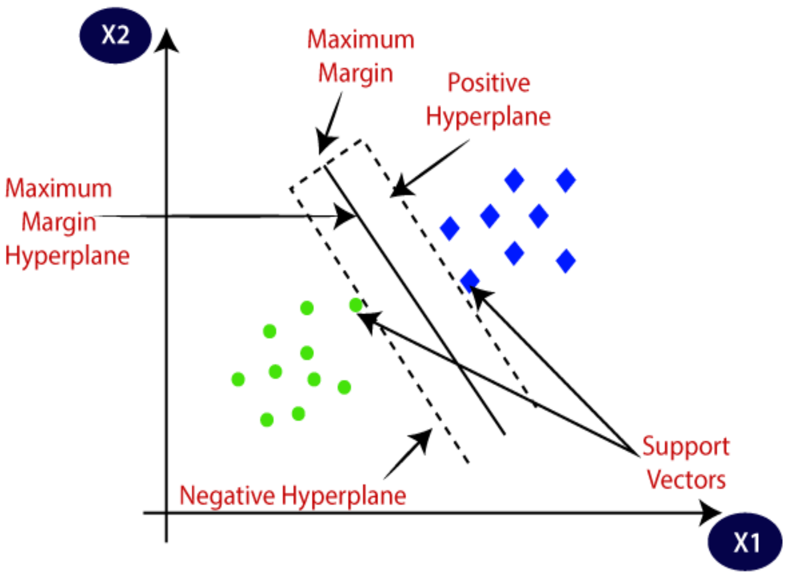


Figure 5: The block diagram of support vector machine (SVM) classifier

The performance metrics of the system

The performance of classifiers about distinct performance measurements such as model training sample size, accuracy of the recognition, false acceptance rate, false rejection rate, and the area-under-receiver operating characteristics (AuROC) curves has been estimated. The false acceptance rate represents the ratio of the probability that the recognition framework will incorrectly experience the test data dataset. The false acceptance rate addresses the range of fake confirmatory numbers

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

by the total number of mixed-up models, as formulated in equation as follows.

$$\text{False Acceptance Rate} = \frac{\textit{Wrongly accepted samples}}{\textit{Total number of wrong samples}}$$

Likewise, false rejection rate is the ratio of the probability that the recognition framework will incorrectly excuse the test data, is formulated in the equation as follows;

$$\text{False Rejection Rate} = \frac{\textit{Wrongly rejected samples}}{\textit{Total number of correct samples}}$$

The shared connection between false acceptance rate and false rejection rate has appeared in figure 6.

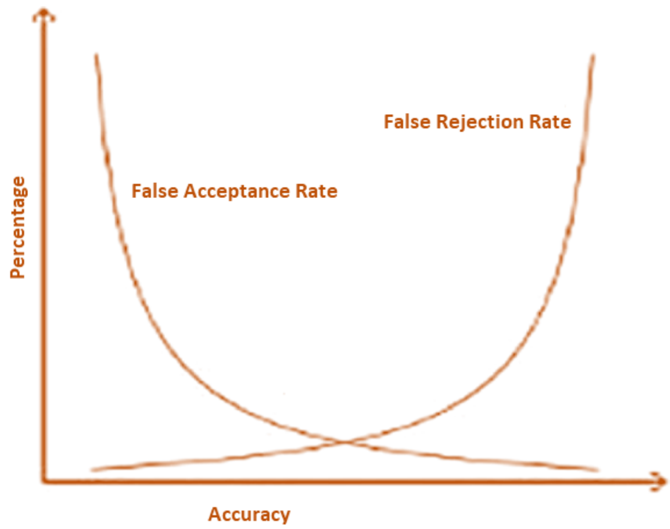


Figure 6: Mutual relationship between False Acceptance Rate and False Rejection Rate

The curve of area under receiver operating characteristics (AuROC) is used in arrangement checking to find which pre-owned models best predict classes. The classifier considered in this task is formulated with a variable number of tests as described in Table 8. We have introduced an exposition metric of these classifiers, looking at the time it takes to assemble the model (Table 9). The accuracy of the recognition obtained using the distinct classification techniques considered in this work that are represented in Table 10.

Classification technique	Data set partitioning strategy					
	a	b	c	d	e	f
CNN	1764.65	1104.34	1117.92	1067.41	1224.91	1124.13
Decision tree	8.48	7.14	7.05	7.13	7.04	7.02
k-NN	0.01	0.01	0.01	0.01	0.01	0.01
SVM	30.93	33.89	34.41	28.46	33.96	30.32
Naïve Bayes	0.28	0.29	0.31	0.29	0.28	0.3
Random forest	30.34	32.24	29.44	32.08	31.56	29.13

Table 9: total time occupied to build to training the model (in second)

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

Outcomes of the experiments

This segment covers the results produced under the experiment, with the distinct model study that are Convolutional Neural Network (CNN), Decision Tree, k-NN, SVM, Naïve Bayes, and Random Forest classifier. A dataset of 92,000 examples has been considered for search results (72,000 images in consonant/character datasets and 20,000 numeral value dataset) for search results. The constructors have used a variable number of tests as a check in the table to formulate six classes. The time taken to prepare the proposed model is presented in Table 9. As shown in Table 9, one can see that k-NN classifier is taking the least time when heterogeneous and different classifiers for model training.

Classification technique	Data set partitioning strategies (in %)					
	a	b	c	d	e	f
CNN	70.98	72.1	73.2	75	75.2	73.7
Decision tree	64.19	65.6	68.3	69.1	70.6	68.8
k-NN	67.95	70.5	71.9	73.7	73.8	74.01
SVM	78.87	80.7	82	81.1	82	81.98
Naïve Bayes	62.71	63.6	64.8	65.1	65.9	64.11
Random forest	84.14	86.11	85.8	87.1	88.12	86.21

Table 10: Accuracy of recognition achieved using the classifiers

Table 10 introduced the accuracy of the recognition that performed that classification with Devanagari Handwritten characters and numerical values. The accuracy of recognition completed with different classifiers is accurately that is illustrated in Figure 7. In Table 10 and Figure 7 it is depicted, that the accuracy of recognition is $87.9\%, 82.5\%, 75.4\%, 74.7\%, 70.7\%, and\ 66.3\%$ with Random Forest, SVM, CNN, K-NN, Decision Tree, and Naïve Bayes respectively have been completed separately with the classifier.

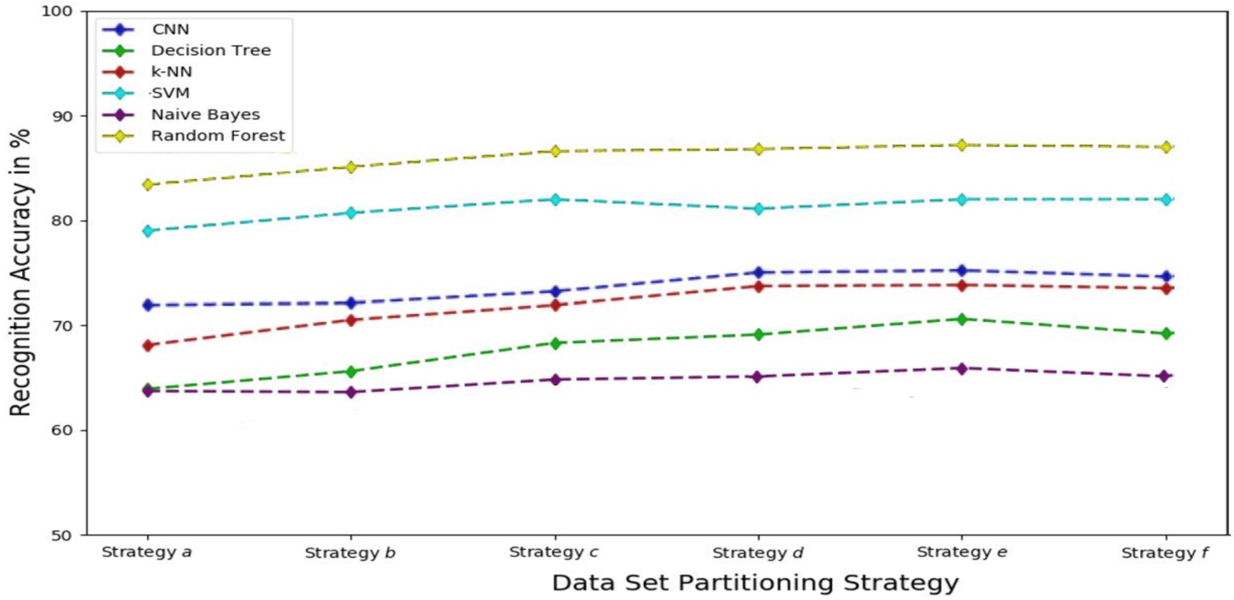


Figure 7: Accuracy of recognition attend using evaluated classification

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

Classification technique	Data set partitioning strategies					
	a(%)	b(%)	c(%)	d(%)	e(%)	f(%)
CNN	0.7	0.7	0.7	0.7	0.6	0.6
Decision tree	0.9	0.8	0.8	0.7	0.7	0.8
k-NN	0.8	0.7	0.7	0.6	0.6	0.7
SVM	0.5	0.5	0.4	0.5	0.4	0.5
Naïve Bayes	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

Table 11: Calculated result of false acceptance rate for various classifiers

Classification technique	Data set partitioning strategies					
	a(%)	b(%)	c(%)	d(%)	e(%)	f(%)
CNN	28.1	27.9	26.8	25	24.9	25.4
Decision tree	36.1	34.4	31.7	30.9	29.4	30.8
k-NN	31.9	29.5	28.1	26.3	26.2	26.5
SVM	21	19.3	18	18.9	18	18
Naïve Bayes	36.3	36.5	36.3	35.4	37.1	35.3
Random forest	16.7	14.9	13.4	13.2	12.8	12.9

Table 12: Calculated results of false rejection rate for various classifiers

Classification technique	Data set partitioning strategies					
	a	b	c	d	e	f
CNN	0.98	0.985	0.988	0.986	0.988	0.987
Decision tree	0.834	0.844	0.858	0.856	0.871	0.861
k-NN	0.844	0.856	0.864	0.874	0.877	0.872
SVM	0.892	0.901	0.908	0.903	0.908	0.908
Naïve Bayes	0.968	0.97	0.971	0.97	0.971	0.969
Random forest	0.993	0.994	0.994	0.994	0.994	0.995

Table 13: Calculation results of area under receiver operating characteristics (AuROC) curve for various classifications

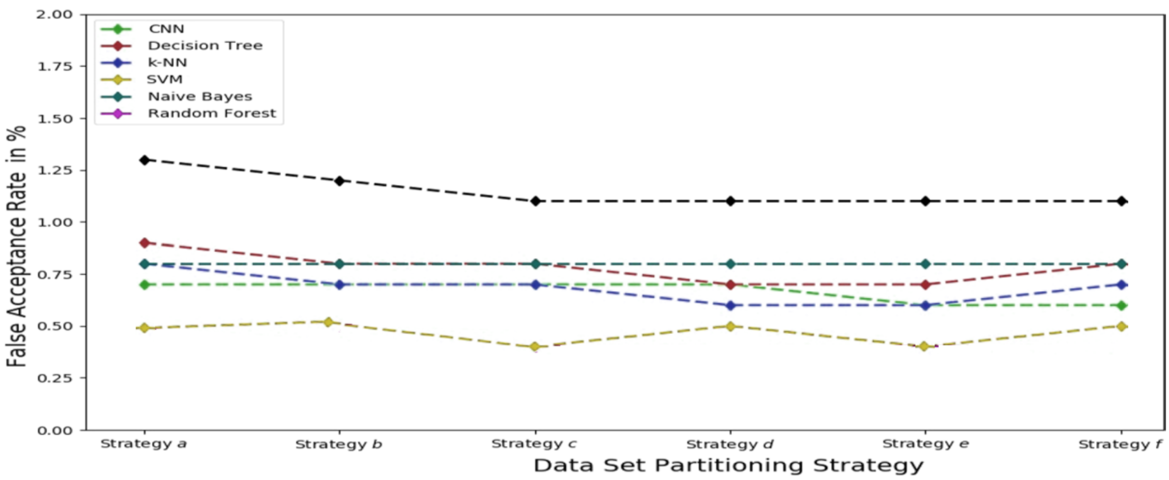


Figure 8: False Acceptance Rate performance analysis with various classifiers

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

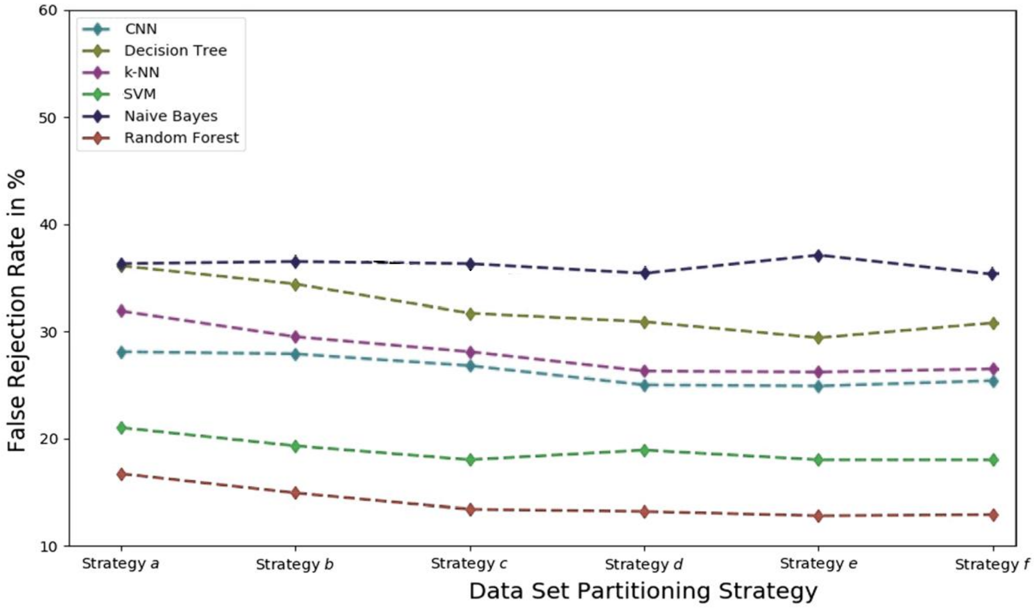


Figure 9: False Rejection Rate performance analysis with various classifiers

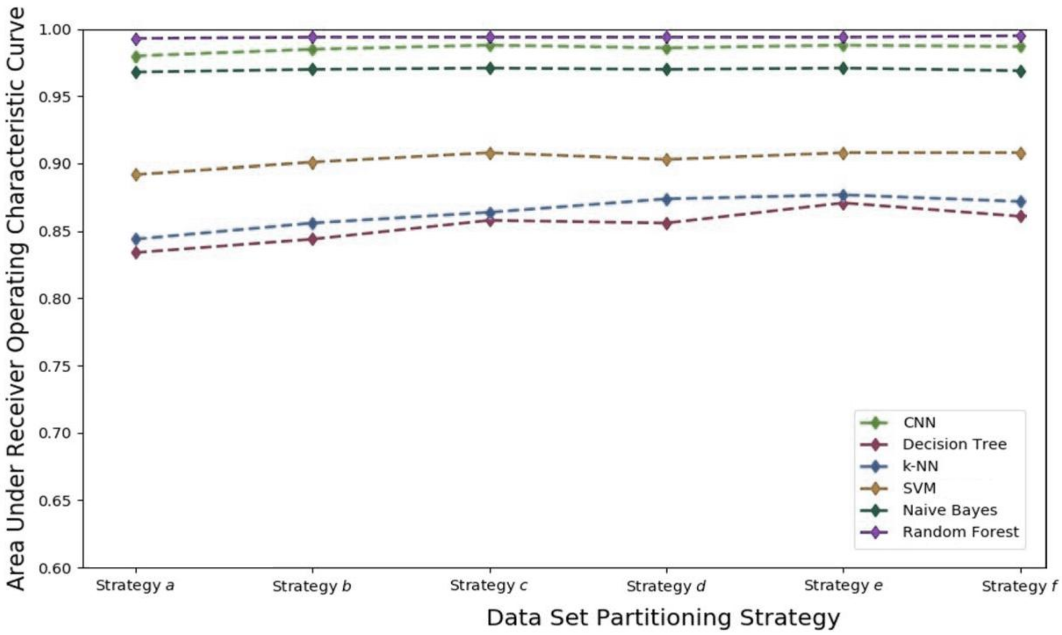


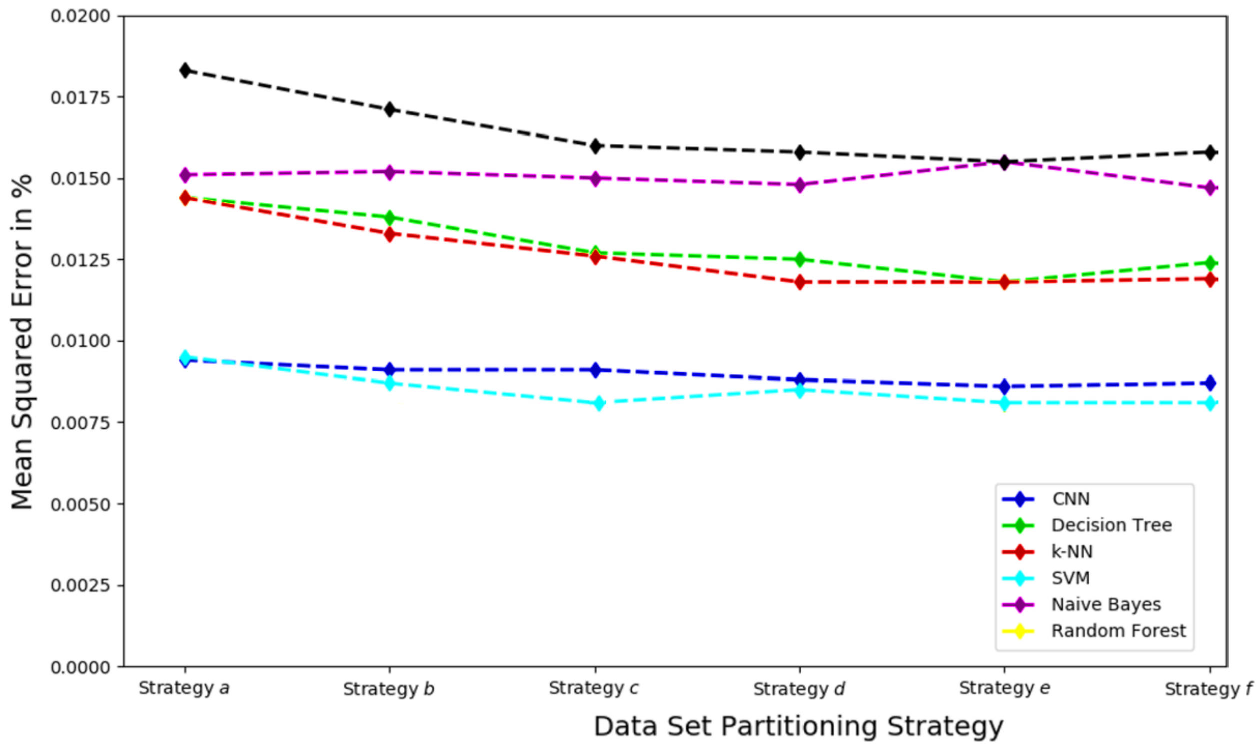
Figure 10: AuROC curve analysis with various classifiers

The false acceptance rate and false rejection rate, and AuROC estimates of the six classifiers considered in this work are illustrated in Tables 11, 12, and 13 and illustrated in figure 8, 9, and 10, respectively.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 252-272
https://publishoa.com
ISSN: 1309-3452

In additionally this paper covers the results of calculation that perhaps the most commonly used loss functions that presents the mean squared error (MSE) for all classifiers considered in this test, which would have calculated the square of the inverse between the actual value and the expected value. The MSE estimates of the six classifiers that considered in this work are individually depicted in Table 14 and illustrated in Figure 11 respectively.

Figure 11: Computed result of Mean Squared error with various classifiers



Classification technique	Data set partitioning strategies					
	a	b	c	d	e	f
CNN	0.0094	0.0091	0.0091	0.0088	0.0086	0.0087
Decision tree	0.0144	0.0138	0.0127	0.0125	0.0118	0.0124
k-NN	0.0144	0.0133	0.0126	0.0118	0.0118	0.0119
SVM	0.0095	0.0087	0.0081	0.0085	0.0081	0.0081
Naïve Bayes	0.0151	0.0152	0.015	0.0148	0.0155	0.0147
Random forest	0.0085	0.008	0.0078	0.0076	0.0076	0.0074

Table 14: Calculation results of mean square error (MSE) for the various classifiers

Resisting the results dependent on the accuracy of recognition, we can see that the accuracy of recognition by the random forest classifier is more accurate than the various classifiers considered in this work. Similarly, the false acceptance rate and false rejection rate, AuROC, and MSE estimates of the Random Forest classifier are additionally depicted as Tables 10, 11, 12, and 13. Random forest classifier applied for individual recognition features with 10-Fold cross-validation strategy is depicted in Table 14. These features are commendable performances for the Devanagari Handwritten character and numerical values recognition.(Sundaram & Ramakrishnan, 2008) These features are additionally valuable for a wide variety of materials, which are basically similar to Devanagari Handwritten scripts. As depicted in Table 15, accuracy of recognition is obtained, with a recognition accuracy of 87.9%, false acceptance rate of 0.4%, and false rejection rate of 12.0%.

Features	Accuracy of Recognition (%)	Training time	False Acceptance Rate (%)	False Rejection Rate (%)	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	0.99	0.0088
Centroid	76.5	24.8	0.6	22.9	0.994	0.0087
Hybrid methodology	87.9	33.2	0.4	12	0.995	0.0076

Table 15: Comparison of performance evaluation based on the individual features with the random forest classifier

Observational Conclusion

To create effective applications under record examination and the recognition process, several directions and alternative options have been highlighted that are used to selecting or extracting the features, and streamlining techniques to improve the accuracy for recognition. Various analysts have proposed for feature extraction or selection procedures and the comparison methods for different materials. The main objective of this paper is to close examine the classifiers for Devanagari Handwritten characters and numerical values recognition. This investigation gives an expected approach towards classification procedures for dataset analysis and recognition in Devanagari Handwritten scripts material. It refers here that by expanding the size of the preparation dataset, the order is accurate and with large improvements. The creators have chosen seven classifiers for character and marking recognition in this work, specifically, Convolutional Neural Network (CNN), Decision Tree, k-NN, SVM, Naïve Bayes, and Random Forest. These classifiers require moderate memory space and computation cost and gives an intelligently high accuracy. In view of the contrast of the results dependent on accuracy of recognition, false acceptance rate, false rejection rate, AuROC and MSE, the authors observed that the Random Forest classifier is performing in a way that Devanagari Handwritten character and numerical recognition- Better than separate classifiers. Analysts may take the new bearing of introducing a novel component extraction and classification strategy giving higher precision rates. One can likewise discover methods of tuning and boosting for order algorithm to ensure that heavy preparation will not occur to fix the set and mess up the higher accuracy of recognition.

References

1. Amin, A., & Singh, S. (1998). Recognition of hand-printed Chinese characters using decision trees/machine learning C4. 5 system. *Pattern Analysis and Applications*, *1*(2), 130–141.
2. Amrouch, M., Es-Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. (2012). Handwritten amazigh character recognition system based on continuous hmms and directional features. *International Journal of Modern Engineering Research*, *2*(2), 436–441.
3. Anil, R., Manjusha, K., Kumar, S. S., & Soman, K. P. (2015). Convolutional neural networks for the recognition of Malayalam characters. *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, 493–500.
4. Bhowmik, T. K., Bhattacharya, U., &

Parui, S. K. (2004). Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. *International Conference on Neural Information Processing*, 814–819.

5. Blue, J. L., Candela, G. T., Grother, P. J., Chellappa, R., & Wilson, C. L. (1994). Evaluation of pattern classifiers for fingerprint and OCR applications. *Pattern Recognition*, *27*(4), 485–501.
6. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.
7. Cordella, L. P., De Stefano, C., Fontanella, F., & Di Freca, A. S. (2014). Random forest for reliable pre-classification of handwritten characters. *2014 22nd International Conference on Pattern Recognition*, 1319–1324.
8. Desai, A. A. (2010). Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recognition*, *43*(7), 2582–2589.
9. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*(7), 1895–1923.
10. Elakkiya, V., Muthumani, I., & Jegajothi, M. (2017). Tamil text recognition using KNN classifier. *Advances in Natural and Applied Sciences*, *11*(7), 41–46.
11. Fischer, S. R. (2004). *History of Writing (pp. 34-40)*. London: Reaktion Books.
12. Gaur, A. (1992). *A history of writing*. British library London, UK.
13. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
14. Hazra, T. K., Singh, D. P., & Daga, N. (2017). Optical character recognition using KNN on custom image dataset. *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, 110–114.
15. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
16. Homenda, W., & Lesinski, W. (2011). Features selection in character recognition with random forest classifier. *International Conference on Computational Collective Intelligence*, 93–102.
17. Huang, J., Lu, J., & Ling, C. X. (2003). Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. *Third IEEE International Conference on Data Mining*, 553–556.
18. Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37.
19. Jeong, S. W., Kim, S. H., & Cho, W. H. (1999). Performance comparison of statistical and neural network classifiers in handwritten digits recognition. *Advances in Handwriting Recognition, World Scientific, Singapore*, 406–415.
20. Jindal, M. K., Sharma, R. K., & Lehal, G. S. (2008). Structural features for recognizing degraded printed Gurmukhi script. *Fifth International Conference on Information Technology: New Generations (Itng 2008)*, 668–673.
21. John, G. H., & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers. *ArXiv Preprint ArXiv:1302.4964*.
22. John, R., Raju, G., & Guru, D. S. (2007). 1D wavelet transform of projection profiles for isolated handwritten Malayalam character recognition. *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, *2*, 481–485.
23. Kim, Y. S. (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 Systems with*

Applications, *34*(2), 1227–1234.

24. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.
25. Kumar, M., Jindal, M. K., & Sharma, R. K. (2014). A novel hierarchical technique for offline handwritten Gurmukhi character recognition. *National Academy Science Letters*, *37*(6), 567–572.
26. Kumar, M., Jindal, M. K., Sharma, R. K., & Jindal, S. R. (2019). Character and numeral recognition for non-Indic and Indic scripts: a survey. *Artificial Intelligence Review*, *52*(4), 2235–2261.
27. Kumar, M., Sharma, R. K., & Jindal, M. K. (2013a). A novel feature extraction technique for offline handwritten Gurmukhi character recognition. *IETE Journal of Research*, *59*(6), 687–691.
28. Kumar, M., Sharma, R. K., & Jindal, M. K. (2013b). Size of training set vis-a-vis recognition accuracy of handwritten character recognition system. *Journal of Emerging Technologies in Web Intelligence*, *5*(4), 380–385.
29. Kumar, M., Sharma, R. K., & Jindal, M. K. (2014). Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. *National Academy Science Letters*, *37*(4), 381–391.
30. Kumar, M., Sharma, R. K., & Jindal, M. K. (2012). Offline handwritten gurmukhi character recognition: Study of different feature-classifier combinations. *Proceeding of the Workshop on Document Analysis and Recognition*, 94–99.
31. Lajish, V. L. (2007). Handwritten character recognition using perceptual fuzzy-zoning and class modular neural networks. *2007 Innovations in Information Technologies (IIT)*, 188–192.
32. LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 396–404.
33. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
34. Lee, D.-S. (1993). Handprinted digit recognition: A comparison of algorithms. *Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition*, 153–164.
35. Lehal, G. S., Singh, C., & Lehal, R. (2001). A shape based post processor for Gurmukhi OCR. *Proceedings of Sixth International Conference on Document Analysis and Recognition*, 1105–1109.
36. Liu, C.-L., Sako, H., & Fujisawa, H. (2002). Performance evaluation of pattern classifiers for handwritten character recognition. *International Journal on Document Analysis and Recognition*, *4*(3), 191–204.
37. Liu, C., Liu, J., Yu, F., Huang, Y., & Chen, J. (2013). Handwritten character recognition with sequential convolutional neural network. *2013 International Conference on Machine Learning and Cybernetics*, *1*, 291–296.
38. Raju, G. (2008). Wavelet transform and projection profiles in handwritten character recognition-A performance analysis. *2008 16th International Conference on Advanced Computing and Communications*, 309–314.
39. Ramanan, M., Ramanan, A., & Charles, E. Y. A. (2015). A hybrid decision tree for printed Tamil character recognition using SVMs. *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 176–181.
40. Rampalli, R., & Ramakrishnan, A. G. (2011). Fusion of Complementary Online and O ine Strategies for Recognition of Handwritten Kannada Characters. *Journal*

of Universal Computer Science, *17*(1), 81–93.

41. Rashad, M., & Semary, N. A. (2014). Isolated printed Arabic character recognition using KNN and random forest tree classifiers. *International Conference on Advanced Machine Learning Technologies and Applications*, 11–17.
42. Rathi, R., Pandey, R. K., Jangid, M., & Chaturvedi, V. (2012). Offline handwritten Devanagari vowels recognition using KNN classifier. *International Journal of Computer Applications*, *49*(23).
43. Sastry, P. N., Krishnan, R., & Ram, B. V. S. (2010). Classification and identification of Telugu handwritten characters extracted from palm leaves using decision tree approach. *J. Applied Engn. Sci*, *5*(3), 22–32.
44. Shanthi, N., & Duraiswamy, K. (2010). A novel SVM-based handwritten Tamil character recognition system. *Pattern Analysis and Applications*, *13*(2), 173–180.
45. Sharma, A., Kumar, R., & Sharma, R. K. (2008). Online handwritten Gurmukhi character recognition using elastic matching. *2008 Congress on Image and Signal Processing*, *2*, 391–396.
46. Sharma, D., & Jhajj, P. (2010). Recognition of isolated handwritten characters in Gurmukhi script. *International Journal of Computer Applications*, *4*(8), 9–17.
47. Sharma, D. V., & Lehal, G. S. (2009). Form field frame boundary removal for form processing system in Gurmukhi script. *2009 10th International Conference on Document Analysis and Recognition*, 256–260.
48. Sharma, D. V., Lehal, G. S., & Mehta, S. (2009). Shape encoded post processing of Gurmukhi OCR. *2009 10th International Conference on Document Analysis and Recognition*, 788–792.
49. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.
50. Srikantan, J. F. G., & Srihari, S. (1994). Handprinted character/digit recognition using a multiple feature/resolution philosophy. *Proceedings of International Workshop Frontiers in Handwriting Recognition*, 57–66.
51. Sundaram, S., & Ramakrishnan, A. G. (2008). Two dimensional principal component analysis for online character recognition. *Proceedings of 11 Th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 88–94.
52. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.
53. Yuan, A., Bai, G., Jiao, L., & Liu, Y. (2012). Offline handwritten English character recognition based on convolutional neural network. *2012 10th IAPR International Workshop on Document Analysis Systems*, 125–129.
54. Zahedi, M., & Eslami, S. (2012). Improvement of random forest classifier through localization of persian handwritten OCR. *ACEEE Int. J. Inf. Technol*, *1*(2), 31–36.
55. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision*, 818–833.
56. Zhu, X., Shi, Y., & Wang, S. (1999). A new distinguishing algorithm of connected character image based on Fourier transform. *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, 788–791.