Evolutionary Intelligence (2022) 15:1499–1516
https://doi.org/10.1007/s12065-021-00597-8

SPECIAL ISSUE

# Automatic handwritten character recognition of Devanagari language: a hybrid training algorithm for neural network

**Prashant Madhukar Yawalkar[1] · Madan Uttamrao Kharat[1]**

Received: 23 August 2019 / Revised: 19 February 2021 / Accepted: 20 March 2021 / Published online: 12 April 2021


## Abstract

In the field of image processing and Artificial Intelligence the character recognition and handwritten recognition have been emerging as one of the active and challenging research areas. In recent years, there has been a notable development in the research associated with character recognition in handwritten Devanagari documents. To improve the recognition performance, this paper tactics to develop new handwritten character recognition model using an improved machine learning approach. The proposed character recognition model includes four stages like pre-processing, Segmentation, Feature Extraction, and Classification. Initially, the scanned handwritten document for Devanagari language is subjected to pre-processing, which includes stages like, RGB to gray, thresholding, complement of image, morphological operations, linearization and noise removal using Median Filter. Then, the characters of the pre-processed image are segmented using k-means clustering that is a popular method for cluster analysis. Further, the features like, Kirsch Directional Edge, Freeman chain code and neighborhood distance weight using Delaunay Triangulation are extracted from the segmented characters. Subsequently, the classification of characters is done using Neural Network (NN) where the new training algorithm is used, in which the weights are optimized using a hybrid optimization by combining Lion Optimization Algorithm and Grey Wolf Optimization (GWO). Here, the update procedure of GWO is based on LA algorithm and hence, the proposed algorithm is named as Lion Updated GWO (LU-GWO). To the next of the implementation, a valuable comparative analysis confirms the improved performance of the proposed LU-GWO-NN model over conventional methods in classifying the consonants, numerals, and vowels from different characters.

**Keywords** Devanagari language · Character recognition · Pre-processing · Feature extraction · Neural network classification · Optimization

## Abbreviations

| | |
|---|---|
| CR | Character recognition |
| MF | Median filter |
| KDE | Kirsch directional edge |
| FCC | Freeman chain code |
| NN | Neural network |
| LOA | Lion optimization algorithm |
| GWO | Grey wolf optimization |
| AI | Artificial intelligence |
| DCNN | Deep convolution NN |
| HACR | Handwritten Arabic character recognition |
| CNN | Convolutional NN |
| 3D | Three dimensional |
| FV | Fisher vector |
| GMM | Gaussian mixture models |
| IAHCCR | In-air handwritten Chinese CR |
| SNN | Spiking NN |
| LIF | Leaky integrate and fire |
| HMM | Hidden Markov model |
| HAS | Harmony search algorithms |
| RGB | Red green blue |
| FPR | False positive rate |
| FNR | False negative rate |
| NPV | Net present value |
| FDR | False discovery rate |
| MCC | Matthew's correlation coefficient |
| LM-NN | Levenberg Marquardt-neural network |
| GWO-NN | Grey wolf optimization-neural network |
| FF-NN | FireFly-neural network |

✉ Prashant Madhukar Yawalkar
yawalkarprashantm@gmail.com;
prashant25yawalkar@gmail.com

Madan Uttamrao Kharat
mukharat@rediffmail.com

[1] MET's Institute of Engineering, BKC, Nashik, India

GD-NN Gradient descent-neural network
LA-NN Lion algorithm-neural network

## 1 Introduction

CR [1–3] is regarded as one of the significant technologies in the present world. It is exploited in diverse fields namely, AI, pattern, and computer vision [4, 5]. Handwritten CR [6–8] is partitioned into two types i.e. offline handwritten CR and online handwritten CR. In fact, CR [9, 10] systems address wide research and multifaceted area. Introducing a constructive system necessitates a remarkable dataset to be examined that leads to the consumption of extended time. The most important process, which influences the recognition success rate might exist in the schemes exploited in CR [11, 12] like pre-processing, extracted features and recognition. The shortcomings of establishing a proficient system are not limited to its accuracy and identification rate, but, also its identification time particularly in online identification [13–15]. As a result, it is much general that identification systems might compromise the rate of success to accomplish a satisfactory recognition time.

Devanagari is one of the renowned characters, which includes eleven vowels and thirty-three consonants. They are known as fundamental characters. Vowels could be mentioned as independent letters, and when they are mentioned in this way they are recognized as modifiers and the characters thus created are known as conjuncts [16–18]. At certain times, two or additional consonants can merge and obtain novel shapes. These novel shape clusters are said to be composite characters. These kind of fundamental characters and modifiers are available not only in Devanagari; it is also available in other characters. Devanagari is moreover used for writing Nepali, Sanskrit, and Marathi [19–21].

In current years efforts were made to build up text CR systems in almost the entire scripts and languages of the world. The efforts were done by a research report in 1977 on handwritten Devanagari characters, and the outcomes were obtained with a limited rate of success [22–25]. Devanagari efforts are printed by combining the characters or by integrating characters to include compound characters and moreover by applying "matras" in diverse forms, this makes the recognition or interpretation very complicated. Hence, more contemplation has to be focused regarding the handwritten CR in the Devanagari language [26]. Nowadays, Evolutionary Algorithm such as LOA [27] and GWO model [28] has been successfully applied to find the answer to numerous problems from pattern recognition area. But these algorithms are easy to fall into local optimum, slow convergence and achieve less accuracy. To overcome the convergence speed and to improve the accuracy of the system the hybrid algorithm LU-GWO model is employed. In addition this hybrid algorithm is used to select the optimal weights.

This paper contributes a novel handwritten model using intelligent approaches. Here, the scanned handwritten document for Devanagari language is provided for pre-processing that involves stages such as RGB to gray, thresholding, and complement of an image, morphological operations, linearization and noise removal using MF. Further, the characters of the image are segmented by deploying k-means clustering, and the features namely, KDE, FCC and neighborhood distance weight based on the Delaunay function are extracted from the segmented image. By using NN, the classification is performed, and the weights of NN are optimized by employing a new hybrid optimization algorithm termed as LU-GWO. Finally, the proposed LU-GWO model is distinguished over other traditional models, and the outcomes are examined in terms of varied measures like accuracy, sensitivity, specificity, precision, FPR, FNR, NPV, FDR, F1-score and MCC. The paper is organized as follows. Section II explains the related works that are analyzed in this topic. Section III describes the overall architecture of Devanagari CR, and Section IV portrays the pre-processing, segmentation and feature extraction stages. Moreover, Section V describes the hybrid training algorithm-based NN for Devanagari CR. Section V describes the results, and Section VI concludes the paper.

## 2 Literature review

### 2.1 Related works

Chaouki et al. [1] have analyzed the applicability of DCNN by means of transfer learning approaches implemented based on a novel extended edition of the presented database in recent times. Finally, the outcomes have demonstrated acceptable identification accuracies, and it was found to perform better than the entire other well-known conventional techniques in the field of HACR. Therefore, the effectiveness of the adopted scheme over the conventional techniques was confirmed.

Dazhen et al. [2] have established an approach for identifying Chinese Character CAPTCHAs. Here, an approach based on CNN was adopted to discover radicals, strokes and character aspects of Chinese characters, and demonstrate that the implemented scheme was better than other conventional models. In addition, the association between accuracy, the iterations, and count of training samples were formulated that was exploited to approximate the performance of the established model. Initially, this model significantly develops the identification accuracy of CAPTCHAs with background, revolution and distortion noise. Finally, the adopted model was assessed, and promising investigational results were attained when distinguished with the state-of-the-art schemes.

Shi et al. [3] have introduced a new FV model attained from GMM on CR, and this model includes spatial data as well. Here, a wide-ranging assessment of FV together with a linear classifier that involves both the scene character and handwritten CR. In addition, two Chinese scene CR datasets were moreover gathered to estimate the appropriateness of FV to symbolize Chinese characters. Finally, it was demonstrated that supplementary spatial information was extremely constructive for CR, particularly for Chinese ones that have many multifaceted configurations.

Qu et al. [4] have exploited a CNN model into IAHCCR, and novel bend directional feature maps were also presented. Followed by that, the amalgamation of diverse maps were combined with CNN and attained improved identification performance when distinguished with erstwhile techniques adopted for IAHCCR. Further, for enhancing the identification rate, a novel data augmentation technique devoted to IAHCCR was proposed. The implemented data augmentation technique merges local distortion with global conversion and efficiently increases the training dataset. At last, the investigational outcomes revealed that the implemented techniques could significantly enhance the recognition rate for IAHCCR.

Bawane et al. [5] have portrayed the analysis regarding recognition and classification of objects and several handwritten characters by exploiting one of the well-known models of SNN. In addition, two-level network schemes for CR and LIF approach for object identification was also exploited. In object identification, following the image preprocessing functions, the image was subjected to numerous models for extracting the features. These feature vectors were subsequently provided to a classifier for categorizing an object. Finally, the adopted LIF scheme was distinguished with conventional approaches, and better outcomes were attained.

Rina et al. [6] have introduced a novel HMM and HSA for online Kurdish CR. The Markov representation was incorporated as a transitional group classifier rather than other character classifiers as in the majority of preceding works. The arrangement has attained better successful recognition, and this model has demonstrated an extreme development in identification rate when distinguished with identical systems, which exploits HMM as its major classifier. At last, it was established that the introduced technique enhances the overall correctness of diverse data sets.

Mhiri et al. [7] have established a methodology based on DCNNs that does not necessitate the precise character segmentation, and it could discover an appropriate demonstration for handwritten data in a programmed approach. The established model exploits a CNN to study the mapping of word descriptions to a robust demonstration, known as a pyramid of character sequences. The proposed model could transfer information across words including similar sub-sequences by modeling the allocation of characters implicitly in the data. Here, the experimental outcomes pointed out that the adopted system performs better than other conventional models by a noteworthy identification.

Sheng et al. [8] have established a novel deep learning technique for scanned sunspot portrait handwritten CR. The CNN was an approach of deep learning that was really thriving in the training of multi-layer network configuration. In addition, the benefits of the adopted technique demonstrated the sunspot drawings, and it also attains the sunspot counts and sunspot regions. The experimental results demonstrated that the established technique attains a better identification rate. Furthermore, the adopted representation was examined, and significant conclusions authenticate the improved performance of the established technique when distinguished with conventional schemes.

Thakur et al. [29] have proposed the recognition of printed Hindi characters in Devanagari script. The aim of this paper was towards the recognition of the individual consonant and vowel which can be later reached out to perceive complex inferred words. In this undertaking, the fundamental accentuation was given towards the recognition of the individual consonant and vowel. This approach provides a 97.4% recognition rate as compared to 94.5% for existing techniques, which indicates that the proposed approach was compared to the techniques used in the existing method.

Shalini et al. [30] have presented an efficient Devanagari character classification model using SVM for printed and handwritten mono-lingual Hindi, Sanskrit and Marathi documents, which first preprocesses the image, segments it through projection profiles, removes shirorekha, extracts features, and then classifies the shirorekha-less characters into pre-defined character categories. The experiments performed on the proposed system obtained average classification accuracies of 99.54% and 98.35% for printed and handwritten images respectively and showed better performance than other techniques.

## 2.2 Review

Table 1 shows the methods, challenges, and features of traditional schemes depending on the handwritten CR models. At first, the DCNN algorithm was introduced in [1], which offers better accuracy, and it reshapes the AI. However, it needs a huge count of constraints. DNN algorithm was exploited in [2] that minimize distortion, and it also offers increased identification accuracy, but it requires contemplation on computational time. In addition, FV approach was deployed in [3] that address the unseen categories and accuracy, and it also offers increased potential ability. Anyhow, it was complex to gather adequate training samples. Likewise, CNN scheme was exploited in [4], which eliminates the distortion and it also offers enhanced rate of recognition.

**Table 1** Features and challenges of handwritten character recognition models using various techniques

| Author [citation] | Adopted methodology | Features | Challenges |
|---|---|---|---|
| Chaouki et al. [1] | DCNN | Improved accuracy<br>Reshapes the AI | Requires huge count of constraints |
| Dazhen et al. [2] | DNN | Better identification accuracy<br>Reduces distortion | Needs contemplation on computational time |
| Shi et al. [3] | FV | Addresses the unseen categories<br>Increased potential ability | Complex to gather adequate training samples |
| Qu et al. [4] | CNN | Minimizes the distortion<br>Enhances the rate of recognition | Weight has to be balanced in a precise way |
| Bawane et al. [5] | LIF model | Reduced noise | Requires contemplation on real time applications |
| Rina et al. [6] | HMM | Reduces the matching score<br>Better succession rate | Matching term raises with raise in mismatching directions |
| Mhiri et al. [7] | CNN | Efficient recognition of characters<br>Increased computational efficiency | Increase in error for more count of words |
| Sheng et al. [8] | CNN | High identification accuracy<br>Highly robust | Requires more research on sunspot functions |
| Thakur et al. [42] | NN | Achieves higher accuracy | Requires more time for processing |
| Shalini et al. [43] | SVM | Better segmentation | Needs improvement in multi-fonts and italics text |

However, the weight has to be balanced in a precise way. Also, LIF model was employed in [5], which minimizes the noise and offers better segmentation; however it needs consideration on real time applications. HMM, algorithm was exploited in [6] that offer better succession rate, and it reduces the matching score, anyhow, matching term rises with raise in mismatching directions. CNN was implemented in [7], which offers efficient CR, and it also offers better computational efficiency, but an increase in error for more count of words. CNN algorithm was suggested in [8] that provides better recognition accuracy and is highly robust. NN algorithm However, it requires more research on sunspot functions. NN was introduced in [29] achieves better accuracy but it requires large processing time. At last, SVM framework was used in [30] achieves a better segmentation but it needs improvement in multifont and italics text. There, these limitations have to be considered for improving the performance of CR models effectively in the current research work.

## 3 Overall architecture of Devanagari character recognition

The diagrammatic representation of the proposed handwritten CR model is shown in Fig. 1. In this research work, a novel handwritten CR model is implemented, which consists of four stages: (i) pre-processing, (ii) segmentation, (iii) feature extraction, and (iv) classification. Here, the handwritten document is manually created, and it includes consonants, numerals, and vowels. Initially, the scanned handwritten

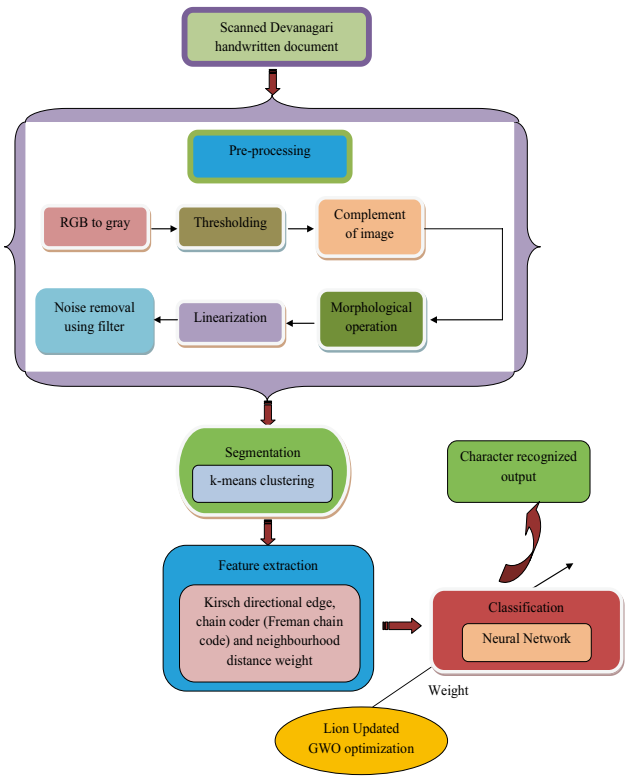


**Fig. 1** Diagrammatic representation of the proposed handwritten CR Model

document for Devanagari language is subjected to pre-processing, which comprises phases namely, RGB to gray, thresholding, the complement of an image, morphological

operations (opening), linearization and noise removal using MF. Then, the characters of the pre-processed image are segmented by means of k-means clustering. Further, the features such as KDE, FCC and neighborhood distance weight based on the Delaunay function are extracted from the segmented characters. Subsequently, the classification process is done using NN where the new training algorithm is used, in which the weights are optimized using a hybrid optimization by combining LOA and GWO. The classification output is in the form of labels, which categorize whether the input testing image belongs to the consonants, numerals or vowels. Thus, the CR is designed efficiently by the proposed framework for Devanagari's handwritten CR document.

## 4 Pre-processing, segmentation and feature extraction stages

### 4.1 Pre-processing

The input image, $I$ is subjected to pre-processing that includes the below steps.

*RGB to Gray scale conversion* [31] A gray-scale image comprises of varied glooms of gray colour. A RGB image could be transformed into a gray scale image by conserving its image. Equation (1) indicates the function that transforms the true colour image, $I$ to gray scale image $I_g$.

$$I_g = RGB\ Gray(I) \tag{1}$$

The RGB image is the amalgamation of red, blue and green colours, which are subjected to normalization with $R + G + B = 1$. This offers the neutral white colour. The gray scale image is obtained from RGB image by merging 30%, 59% and 11% of red, green and red colours. This generates the information of the image related to brightness. The value 0 and 255 indicate black and white colours, respectively, and the level will be either white or black values. Thus the RGB to gray scale converted image is represented by $I_C$.

*Thresholding* The converted image $I_C$ is then subjected to the thresholding process [32], which is one of the significant methods in image segmentation. It could be revealed by Eq. (2), in which *Thre* denotes threshold value. The coordinates of the threshold value point are given by $u, v$ and $p(u, v), f(u, v)$ signifies gray level image pixel points [33].

$$Thre = Thre\left[u, v, p(u, v), f(u, v)\right] \tag{2}$$

Threshold image $I_{Thre}$ is given by Eq. (3).

$$I_{Thre} = \begin{cases} 1 \ if & f(u, v) > 1 \\ 0 \ if & f(u, v) \leq 1 \end{cases} \tag{3}$$

Thus the thresholded image is represented by $I_{Thre}$.

*Complement of image* The image $I_{Thre}$ is then subjected to a complementary process [34]. The image complement block evaluates the complement of an intensity or binary image. The block substitute's pixel values equivalent to zero with one and pixel values equivalent to one with zero for binary images. The block deducts every pixel value from the highest value which could be indicated by the input data kind and provides the variation for an intensity image. Thus the attained complement of the image is indicated by $I_{Com}$.

*Morphological operation* The complementary image $I_{Com}$ is then given to morphological operation [35]. In image processing, morphology indicates the portrayal of the structure and shape of the entities in an image. The opening of an image is a combined function of dilation and erosion. The opening of an image $A$ by structuring element $B$ is defined as given by Eq. (4).

$$A \circ B = (A \Theta B) \oplus B \tag{4}$$

Equation (4) offers the association among opening and dilation and erosion. It portrays that the opening process is the erosion of an image by $B$ and the resultant is dilated with the similar $B$. The edges of the opened image are the points in $B$, which attains the extreme points of the boundary of $A$ as $B$ is 'rolled' around within this boundary. The opening process smoothes the object and clearly outlines the narrow bridges and moreover removes negligible extensions available in the object.

The image resulting from the morphological operation is indicated by $I_{Mor}$.

*Linearization* The image attained from the morphological operation, $I_{Mor}$ is further given to the linearization process. The linearization is "the process that converts non-linear image data into linear image data, so it is supposed to regress the non-linear tone curve that has been applied to the linear image data". It is a characteristic task in analyzing the quality of an image, since the entire images which follow a definite colour space (such as, RGB) include a non-linear tone response curve. The image resulting from linearization is indicated by $I_{\text{Lin}}$.

*Noise removal using median filter (MF)* The image obtained from the linearization process $I_{\text{Lin}}$ is then subjected to MF. The MF [36] is a statistical process based on nonlinear signal processing. Since the MF is 'nonlinear, its arithmetical examination is comparatively multifaceted for the image with arbitrary noise. The noise variation of the MF for an image with zero mean noise under normal distribution is given by Eq. (5). In Eq. (5), $\sigma_i^2$ denotes input noise power, $m$ indicates the mask size of MF and $h(\overline{m})$ denotes noise density function.

$$\sigma_{med}^2 = \frac{1}{4mh^2(\overline{m})} \approx \frac{\sigma_i^2}{m + \frac{\pi}{2} - 1} \frac{\pi}{2} \tag{5}$$

Thus the noise removed image using MF is indicated by $I_{MF}$.

### 4.2 K-means clustering

The pre-processed image, thus obtained by the above procedure is subjected to k-means clustering for the segmentation process. Clustering [36] is a scheme that split a group of data into a precise number of sets. A renowned technique among various clustering models is identified as k-means clustering, which divides a group of data to $k$ number of data. It categorizes a specified group of data into $k$ number of disjoint cluster. K-means approach comprises of two partitioned stages. In the initial stage, it evaluates the $k$ centroid, and in the subsequent phase, it acquires every point to the cluster that has an adjacent centroid from the relevant data point. There are various techniques to describe the distance of the closest centroid, and a widely exploited scheme is Euclidean distance. Next, to the grouping process, it evaluates the novel centroid of every cluster and depending on that centroid; a novel Euclidean distance is formulated among every centre and all data points. Then, it allocates the points in the cluster that has the least Euclidean distance. The centroid is the point to which the summation of distances from the entire objects that particular cluster is reduced. Thus, K-means is an iterative approach that reduces the total distances from every data point to its cluster centroid for entire clusters $Cl_{nl}$, in which $nl$ indicates the total number of assigned clusters.

Consider $X_i = \{x_1, x_2, \ldots, x_{xn}\}$ be the set of data points and $C_j = \{C_1, C_2, \ldots, C_{nC}\}$ be the set of centroids.

1. Select the $C_j$ cluster centre (centroid) arbitrarily.
2. Calculate the Euclidean distance, $G$ among every cluster centre and data point as shown by Eq. (6).

$$G = \sum_{i=1}^{xn} \sum_{j=1}^{nC} \left( \left\| X_i - C_j \right\| \right)^2 \tag{6}$$

3. Assign the data point to the cluster centre, which has the least distance.
4. Re-evaluate the novel cluster centre as per Eq. (7), where $xn$ represents the number of data points $n$th cluster.

$$C_j = \left( \frac{1}{xn} \right) \sum_{X_i \in Cl_{nl}} X_i \tag{7}$$

5. Re-evaluate the distance among every new cluster centre and data points.
6. If the data point was not reallocated, stop the function, or else, repeat from step 3.

Thus the clustered image using k-means clustering is represented by $I_{cl}$.

### 4.3 Feature extraction

The clustered image $I_{cl}$ is subjected to a feature extraction process, for which, KDE, FCC and neighborhood distance are used.

*Kirsch directional edge (KDE)* [37] KDE exactly detects the four directional edges as the entire eight adjacent edges are taken in to account in this operation. The directional gradient implemented by Kirsch is specified by Eq. (8). In Eq. (8), $H(r, t)$ indicates the pixel gradient at $(r, t)$.

$$H(r,t) = \max_{k=0}^{7} \left\{ 5J_k - 3V_k \right\} \tag{8}$$

where

$$\begin{aligned} J_k &= a_k + a_{k+1} + a_{k+2} \\ V_k &= a_{k+3} + a_{k+4} + a_{k+5} + a_{k+6} + a_{k+7} \end{aligned} \tag{9}$$

The four directional features are computed as given by Eq. (10).

$$\begin{aligned} H(r,t)_h &= \max\left( \left|5J_0 - 3V_0\right|, \left|5J_4 - 3V_4\right| \right) \\ H(r,t)_v &= \max\left( \left|5J_2 - 3V_2\right|, \left|5J_6 - 3V_6\right| \right) \\ H(r,t)_{rd} &= \max\left( \left|5J_1 - 3V_1\right|, \left|5J_5 - 3V_7\right| \right) \\ H(r,t)_{ld} &= \max\left( \left|5J_3 - 3V_3\right|, \left|5J_7 - 3V_7\right| \right) \end{aligned} \tag{10}$$

In Eq. (10), $H(r,t)_v$, $H(r,t)_h$, $H(r,t)_{ld}$ and $H(r,t)_{rd}$ points out the directional edge features for vertical, horizontal, left and right diagonal directions, correspondingly. The features extracted by FDE are represented by $F_{kde}$.

*Freeman chain code (FCC)* [38] Chain codes are a "shape descriptor" classified into contour dependent scheme, especially the structural scheme. It portrays the shape by a set of line segments with a predetermined set of feasible orientations [8]. Here, general chain codes are 8 or 4 directional. Chain code is "a vector of integer numbers where each number represents one of the possible directions as shown by Fig. 2.

Assume that the sequence of $n$ integer coordinate points portrays a closed curve $c$, $c = \{g_i = (u_i, v_i), i = 1, 2, \ldots, n\}$,

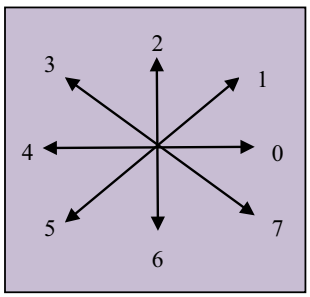


**Fig. 2** Representation of freeman chain code (8 directional)

in which $g_{i+1}$ indicates a neighbour of $b_i$ (modulo $n$). The FCC of $c$ comprises of $n$ vectors, $\vec{h} = g_1 - 1g_1$ and each one of it is indicated by a integer, $p = 0, \ldots, 7$ as revealed in Fig. 1 and here, $\frac{1}{4\pi p}$ indicates the angle among vector and x-axis. Chain of $c$ is portrayed as $\left\{\vec{h},\, i = 1, \ldots, n\right\}$ and $\left\{\vec{h}, = \overline{h}_{i\pm n}\right\}$. The entire integers are modulo $n$. Thus the features extracted by FDE are represented by $F_{fcc}$.

### 4.4 Neighborhood distance based on Delaunay triangulation function

The major fact behind deploying the Delaunay diagram [39] is to evade the need of portraying a distance threshold for finding the neighborhoods, and accordingly, there is not any requirement to continue the procedure of determining neighborhoods for numerous user-defined constraints and to discover neighborhood in a dynamic manner. The pseudocode of the Delaunay function is given by Algorithm 1. The features thus extracted using neighborhood distance based on the Delaunay triangulation function are represented by $F_{df}$.

**Algorithm 1:** Delaunay function

```
Obtain the count of points from user in O.
For i-> 1 to O
   Draw the triangle among every point
   and i.
   Verify for intersection among lines
   Take the entire intersections in
   intersect []
   For j-> 1 to (size of intersect []).
      Remove the respective triangles.
   End of Loop
End of loop
```

Thus the features extracted by neighborhood distance based on Delaunay triangulation function is represented by $F_{nd}$.

## 5 Hybrid training algorithm-based neural network for Devanagari character recognition

### 5.1 Neural network

The features extracted from KDE, FCC and neighborhood distance weight are denoted by $F_{kde}$, $F_{fcc}$, $F_{df}$ and it is combined, which is then applied to NN for classifying the Devanagari characters. Neural networks are artificial systems that were inspired by biological neural networks. Accordingly, the combined feature is represented as per Eqs. (11) and (12). NN [40] is a classifier that intakes the features (combined) as the input ($F_{Fe}$) and it is given by Eq. (11), in which *nu* specifies the number of features.

$$F_{Fe} = F_{kde} + F_{fcc} + F_{df} \tag{11}$$

$$F_{Fe} = \left\{F_{Fe1}, F_{Fe2}, \ldots, F_{Fe\,nu}\right\} \tag{12}$$

The architecture of NN comprises of three layers: Input layer, Hidden layer, and Output layer, in which each layer contains a set of neurons. The output of NN starts with the determination of hidden layer output, and then the overall network output. The output of hidden layers is indicated by Eq. (13), where $i$ denotes the hidden neuron, $W_{(Bi)}^{(H)}$ refers to the bias weight to $i$th hidden neuron, $n_i$ specifies the number of input neurons, $nf$ denotes the activation function, $W_{(Bi)}^{(H)}$ indicates the bias weight to $i$th hidden neuron, $W_{(ji)}^{(H)}$ indicates the weight from $j$th input neuron to $i$th hidden neuron, and $F_{Fe}$ refers to the input features. The overall output of the network is represented in Eq. (15), where $W_{(Bm)}^{(G)}$ refers to the bias weight to $m$th output neuron, $n_h$ indicates the number of output neurons, and $W_{(im)}^{(G)}$ denotes the weight from $i$th hidden neuron to $m$th output neuron. Moreover, the error function refers to the absolute difference between the actual and predicted output as given in Eq. (15), which has to be minimized. In Eq. (15), $G_m$ denotes the actual output, and $\hat{G}_m$ denotes the predicted output as given in Eq. (14).

$$e^{(H)} = nf\left(W_{(Bi)}^{(H)} + \sum_{j=1}^{n_i} W_{(ji)}^{(H)} F_{Fe}\right) \tag{13}$$

$$\hat{G}_m = nf\left(W_{(Bm)}^{(G)} + \sum_{i=1}^{n_h} W_{(im)}^{(G)} e_i^{(H)}\right) \tag{14}$$

$$E^* = \underset{\left\{W_{(Bi)}^{(H)}, W_{(ji)}^{(H)}, W_{(Bm)}^{(G)}, W_{(im)}^{(G)}\right\}}{\arg\min} \sum_{m=1}^{n_G} \left|G_m - \hat{G}_m\right| \tag{15}$$

For minimizing the error, the weight $W_l = \left\{W_{(Bi)}^{(H)}, W_{(Bm)}^{(G)}, W_{(ji)}^{(H)}, W_{(im)}^{(G)}\right\}$ has to be tuned or optimized properly, which is done by the proposed LU-GWO algorithm.

## 5.2 Solution encoding and objective function

In the proposed handwritten CR, the major objective function is to reduce the error among the predicted and actual classified output as given in Eq. (16). As mentioned earlier, the weights of NN, $W_l$ ranging from $W_1$ to $W_N$, $l = 1, 2, \ldots, N$ is given as input for solution encoding as specified by Fig. 3, where $N$ indicates the number of weights allotted for NN.

$$Objective\ Function = Min(E^*) \tag{16}$$

## 5.3 Lion optimization algorithm

The LA [27, 41–43] was established by Raja Kumar and is exploited for large-scale and standard bilinear systems. The progress of this LA model is continued with four imperative steps i.e., mating, pride generation, territorial takeover and territorial defence. Here, the solution vector is specified by $Z$ and is indicated by $Z = [z_1, z_2, \ldots, z_n]$.

### 5.3.1 Pride generation

The initialization of pride is made with territorial lion $Z^{mal}$ and lioness $Z^{fem}$ and a nomadic lion $Z^{nd}$.

The vector element of $Z^{mal}$, $Z^{fem}$ and $Z^{nd}$ is represented as $z_{len}^{mal}$, $z_{len}^{fem}$ and $z_{len}^{nd}$, which are considered as the random integers that lies in the maximum and minimum limits while $n > 1$, in which $len = 1, 2, \ldots, Len$. Here, the length of lion is indicated as $Len$ and is specified by Eq. (17), in which, $n$ and $y$ are integers.

$$Len = \begin{cases} n; & n > 1 \\ y; & otherwise \end{cases} \tag{17}$$

when $n = 1$, the constraints given in Eqs. (18) and (19) have to be satisfied, and $R(z_{len})$ is given by Eq. (20).

$$R(z_{len}) \in \left(z_{len}^{\min}, z_{len}^{\max}\right) \tag{18}$$

$$y\%2 = 0 \tag{19}$$

$$R(z_{len}) = \sum_{len=1}^{Len} z_{len} 2^{\left(\frac{Len}{2} - len\right)} \tag{20}$$

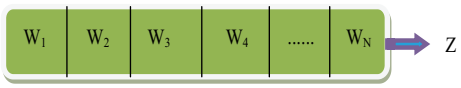


**Fig. 3** Solution encoding (weight optimization)

### 5.3.2 Fertility evaluation

The local optima or global optima is attained by the laggards when $Z^{fem}$ and $Z^{mal}$ gets saturated by the fitness. Here, the laggard is regarded as $Z^{mal}$ and the laggardness rate is specified as $La_r$ and is maximized by 1, when $f(Z^{mal})$ is higher than $f^r$ that indicates reference fitness. After crossover, the fertility of $Z^{fem}$ is confirmed by sterility rate $St_r$ and is maximized by 1. When $St_r$ becomes higher than the tolerance $St_r^{\max}$, then the update for $Z^{fem}$ is done as per Eq. (21). The mating process is done, when the updated female $Z^{fem+}$ is considered as $Z^{fem}$, owing to its enhancement. In Eqs. (21) and (22), $z_{len}^{fem+}$ and $z_d^{fem}$ are considered as $len^{th}$ and $d^{th}$ vector element of $Z^{fem+}$ correspondingly and $\nabla_d$ is given by Eq. (23).

$$z_{len}^{fem+} = \begin{cases} z_d^{fem+} & if\ len = d \\ z_{len}^{fem} & otherwise \end{cases} \tag{21}$$

$$z_d^{fem+} = \min\left[z_d^{\max}, \max\left(z_d^{\min}, \nabla_d\right)\right] \tag{22}$$

$$\nabla_d = \left[z_d^{fem} + (0.1r_2 - 0.05)\left(z_d^{mal} - r_1 z_d^{fem}\right)\right] \tag{23}$$

Arbitrary integer $d$ is produced in the limits between [1, $Len$], the female update operation is indicated by $\nabla$ and the arbitrary integer is specified as $r_1$ and $r_2$ that are produced in the limits between [0, 1].

### 5.3.3 Mating

Crossover and mutation are the two major steps in this process. By this, the cubs are generated by $Z^{mal}$ and $Z^{fem}$. Thus, four cubs are produced by a lioness pregnancy and thus four cubs are generated by the crossover process.

### 5.3.4 Lion operators

The sequence of the territorial defence is specified as survival fight, producing nomad coalition and then nomad and pride coalition updates. If Eqs. (24)–(26) are satisfied, then $Z^{e-nd}$ will be chosen.

$$f(Z^{e-nd}) < f(Z^{mal}) \tag{24}$$

$$f(Z^{e-nd}) < f(Z^{mal\_cub}) \tag{25}$$

$$f(Z^{e-nd}) < f(Z^{fem\_cub}) \tag{26}$$

The update of pride takes place only after the defeat of $Z^{mal}$, whereas the updation of the nomad coalition takes place only after the defeat of $Z^{nd}$. Territorial takeover forces

the model to update $Z^{mal}$ and $Z^{fem}$ if $Z^{mal\_cub}$ and $Z^{fem\_cub}$ are matured, i.e. while the age of cubs goes beyond the highest age for cub maturity, $A^{\max}$.

#### 5.3.5 Termination

The implementation of the algorithm will terminate only if any of the two conditions as given in Eqs. (27) and (28) is satisfied. In Eq. (27), the count of generation is specified as $t$ which is fixed as zero initially and increased by 1 while the territorial takeover is performed.

$$t > t_{\max} \tag{27}$$

$$f(Z^{mal}) - f(Z^{opt}) \leq er_{th} \tag{28}$$

The error threshold and maximum count of generations is specified by $er_{th}$ and $t_{\max}$ correspondingly.

### 5.4 Grey wolf optimization

The mechanism of GWO [28, 44] portrays the grey wolves' hierarchy of leadership and hunting features. There are three groups of grey wolves, i.e. $\alpha$, $\beta$, $\delta$ that are exploited for carrying out the leadership hierarchy. Piercing, encircling, and attacking are the three chief actions in hunting for improving optimization. The wolves, $\alpha$, $\beta$ and $\delta$ are considered as the foremost wolves that perform hunting. Here, $\alpha$ is assigned as the leader which formulates a resolution relating to hunting, time to awake, sleeping site, etc. $\beta$ and $\delta$ takes 2nd and 3rd level that aids $\alpha$ in taking choices. The last wolf $\delta$ is engaged only for eating. The encircling characteristics of wolves are designed by means of Eqs. (29) and (30), in which $U$ and $Y$ addresses the coefficient vectors, $Z_p$ indicates prey's position vectors, $t$ signifies the present iteration and $Z$ addresses the grey wolves' position vectors. The configuration for $U$ and $Y$ is specified by Eqs. (31) and (32), and $a$ is a factor that is reduced gradually from 2 to 0 for the whole iterations and $r_1$ and $r_2$ signifies the arbitrary vectors, which were distributed constantly among [0, 1].

$$D = \left| Y.Z_p(t) - Z(t) \right| \tag{29}$$

$$Z(t+1) = Z_p(t) - U.D \tag{30}$$

$$U = 2a.r_1 - a \tag{31}$$

$$Y = 2r_2 \tag{32}$$

The numerical principle for relating the hunting nature of the wolves is accessible in Eqs. (33)–(39), in which the final modified position of wolves is given by Eq. (39) that offers the updated $Z$.

$$D_\alpha = \left| Y_1 - Z_\alpha - Z \right| \tag{33}$$

$$D_\beta = \left| Y_2 - Z_\beta - Z \right| \tag{34}$$

$$D_\delta = \left| Y_3 - Z_\delta - Z \right| \tag{35}$$

$$Z_1 = Z_\alpha - U_1.\left(D_\alpha\right) \tag{36}$$

$$Z_2 = Z_\beta - U_2.\left(D_\beta\right) \tag{37}$$

$$Z_3 = Z_\delta - U_3.\left(D_\delta\right) \tag{38}$$

$$Z(t+1) = \frac{Z_1 + Z_2 + Z_3}{3} \tag{39}$$

### 5.5 Proposed LU-GWO algorithm

Rather than the amazing facts of the conventional GWO algorithm, it includes certain drawbacks such as low precision, bad searching mechanism and a slower rate of convergence. In the conventional GWO approach, positions of $\alpha$, $\beta$, $\delta$ are updated based on Eqs. (36)–(38). To improve the performance of conventional GWO, the proposed logic prevails over the abovementioned challenges. Here, the variables $D_\alpha$, $D_\beta$ and $D_\delta$ of GWO is updated based on the update formula of the female lion (Eq. (22)) as shown in Eqs. (40)–(42), where $U$ indicates the coefficient vectors as shown in Eq. (31). The pseudocode for the proposed LU-GWO scheme is given by algorithm 2.

$$Z_1 = \min\left[Z_d^{\max}, \max\left(Z_d^{\min}, \left[Z_d^{fem} + (0.1r_2 - 0.05)(Z_\alpha - U.D_\alpha)\right]\right)\right] \tag{40}$$

$$Z_2 = \min\left[Z_d^{\max}, \max\left(Z_d^{\min}, \left[Z_d^{fem} + (0.1r_2 - 0.05)(Z_\beta - U.D_\beta)\right]\right)\right] \tag{41}$$

$$Z_3 = \min\left[Z_d^{\max}, \max\left(Z_d^{\min}, \left[Z_d^{fem} + (0.1r_2 - 0.05)(Z_\delta - U.D_\delta)\right]\right)\right] \tag{42}$$

**Algorithm 2: Proposed LU-GWO Algorithm**

Initialize the grey wolves' population $Z_p$ ( $p = 1,2,3..n$ )
Initialize $a$, $U$ and $Y$
Compute the fitness of all search agents
Assign $Z_\alpha$ as the best search agent
Assign $Z_\beta$ as the second best search agent
Assign $Z_\delta$ as the third best search agent
While $(t < t_{\max})$, $t_{\max}$ denotes the maximum iteration
  For each wolf
    Determine $Z_1$ as per Eq. (40)
    Determine $Z_2$ as per Eq. (41)
    Determine $Z_3$ as per Eq. (42)
    Update the position by means of Eq. (39)
  End for
  Update $a$, $U$ and $Y$
  Compute the fitness for whole search agents
  Update $Z_\alpha$, $Z_\beta$ and $Z_\delta$
  $t = t + 1$
End while
Return $Z_\alpha$

## 6 Results and discussions

### 6.1 Simulation procedure

The proposed Devanagari CR model has been implemented in MATLAB 2018a, and the significant results were obtained. The database used is manually prepared Devanagari database, which comprises of 36 consonants, 10 numerals, and 12 vowels. Each folder has more than 200 images. The dataset consists of three test cases: The 1st test case consists of 7415 images, 2nd test case consists of 2880 images and 3rd test consists of 2652 images. After the implementation, the proposed LU-GWO-NN was compared with conventional algorithms namely, LM-NN [45], GD-NN [46], FF-NN [47], GWO-NN [28] and LA-NN [27] algorithms. Moreover, the analysis was done for three test cases with respect to the performance measures such as accuracy, sensitivity, specificity, precision, FPR, FNR, NPV, FDR, F1-score and MCC, and enhanced outcomes were attained. In our proposed method the weights are optimally selected using the hybrid optimization by combining Lion Optimization Algorithm (LOA) and Grey Wolf Optimization (GWO). This optimal solution achieves better results than the traditional methods. In the proposed method, hybrid training algorithm-based Neural Network evaluated the features more efficiently than other models.

The sample original images with pre-processed and segmented images for consonants, numerals, and vowels are shown in Fig. 4. Formulas for calculating Accuracy, Sensitivity, Specificity, Precision, FPR, $F_1$ Score, MCC, FNR, NPV, FDR is given in Eqs. (43)–(52).

Accuracy: The accuracy is described as the degree of closeness of an estimated value with respect to its original value.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{43}$$

Sensitivity: It refers to the fraction of positives that are recognized by the classifier correctly.

$$Sensitivity = \frac{TP}{TP + FN} \tag{44}$$

Specificity: It refers to the ratio of negatives identified using the classifier.

$$Specificity = \frac{TN}{FP + FP} \tag{45}$$

where TP, TN, FP and FN signify true positive, true negative, false positive and false negative achieved during the classification.

Precision: Precision quantifies the number of positive class predictions that belong to the positive class.

$$Precision = \frac{TP}{TP + FP} \tag{46}$$

F1-score: It provides a single score that balances both the concerns of precision and recall in one number. Thus the F score is defined as the weighted harmonic mean of the precision and recall.

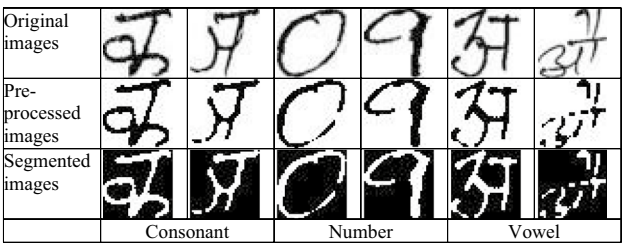


**Fig. 4** Sample images of the proposed CR model of Devanagari language for **a** consonants, **b** numerals, **c** vowels

$$F_{score} = 2 \times \frac{Precision \times Sensitivity}{(Precision + Sensitivity)} \tag{47}$$

NPV: It is the likelihood that a negative test result reflects the absence of disease

$$NPV = \frac{TN}{TN + FN} \tag{48}$$

FPR: It is defined as the number of incorrect positive results that occurred among all negative samples available during the test.

$$FPR = \frac{FP}{FP + TN} \tag{49}$$

FNR: It is the proportion of positives that yield negative test outcomes i.e., the conditional probability of negative test results given that the condition being looked for is present.

$$FNR = \frac{FN}{TP + FN} \tag{50}$$

FDR: It is the proportion of individuals with a positive test result for which the true condition is negative.

$$FDR = \frac{FP}{FP + TP} \tag{51}$$

MCC: This approach is recommended for binary data when class numbers are unequal and is preferred to simpler approaches, such as the average of the sensitivity and specificity, as it preserves information about all four components of the contingency matrix in an unbiased way.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{52}$$

### 6.2 Performance analysis

The performance analysis of the developed LU-GWO-NN for handwritten CR model with respect to three test cases is given by Figs. 5, 6 and 7. From Fig. 5a, for 1st test case, the proposed LU-GWO-NN model in terms of accuracy at 42nd population size is 0.15% better than LM-NN, 0.15% better than GD-NN, 0.26% better than FF-NN, 0.2% better than GWO-NN and 0.41% better than LA-NN techniques. From Fig. 5c, the specificity of the adopted LU-GWO-NN method at 85th population size is 0.24% superior to LM-NN, 0.19% superior to GD-NN, 0.19% superior to FF-NN, 0.19% superior to GWO-NN and 0.14% superior to LA-NN techniques. The precision that is attained from Fig. 5d, which is 12% improved than LM-NN, 8.6% improved than GD-NN, 8.6% enhanced than FF-NN, 10% superior to GWO-NN and 9% better than LA-NN schemes at 85th population size. Also, from Fig. 5e, the FPR of the adopted LU-GWO-NN model at 85th population size is found to be 99.96% superior to LM-NN, 99.5% superior to GD-NN, 99.5% superior to FF-NN, 99.5% superior to GWO-NN and 99.41% superior to LA-NN algorithms. In addition, from Fig. 5g, the NPV of suggested LU-GWO-NN model at 85th population size is 0.24% superior to LM-NN, 0.19% superior to GD-NN, 0.19% superior to FF-NN, 0.19% superior to GWO-NN and 0.14% superior to LA-NN algorithms. In Fig. 5h, the FDR of presented LU-GWO-NN model at 85th population size is found to be 11% superior to LM-NN, 10% superior to GD-NN, 10% superior to FF-NN, 10% superior to GWO-NN and 8% superior to LA-NN approaches. Also, from Fig. 5i, the F1-score of the adopted LU-GWO-NN scheme at 85th population size is 3.66% superior to LM-NN, 3.7% improved than GD-NN, 3.7% better than FF-NN, 3.7% superior to GWO-NN and 2.44% better than LA-NN models. Finally, from Fig. 5j, the offered LU-GWO-NN scheme in terms of MCC at 45th population size is 5% superior to LM-NN, 17.5% superior to GD-NN, 17.5% superior to FF-NN, 17.5% superior to GWO-NN and 22.5% superior to LA-NN algorithms. In addition, the performance analysis of the adopted LU-GWO-NN for handwritten CR with respect to the second test case and the third test case is demonstrated by Figs. 6 and 7 correspondingly. Thus the effectiveness of the proposed LU-GWO-NN algorithm for a better handwritten CR model has been proved by the attained results.

### 6.3 Convergence analysis

The convergence analyses of the traditional and proposed method for three test cases are given in Fig. 8. From Fig. 8c at the 10th iteration, the proposed LU-GWO-NN has convergence in the range of 0.009 and the traditional methods like LM-NN, GWO-NN, LA-NN, GD-NN, FF-NN, PSO-NN and WOA-NN converge in the range of 0.012, 0.023, 0.02, 0.028, 0.022, 0.014 and 0.024. Moreover, the proposed method is seemed to achieve better performance with minimum convergences value for all iterations than conventional methods.

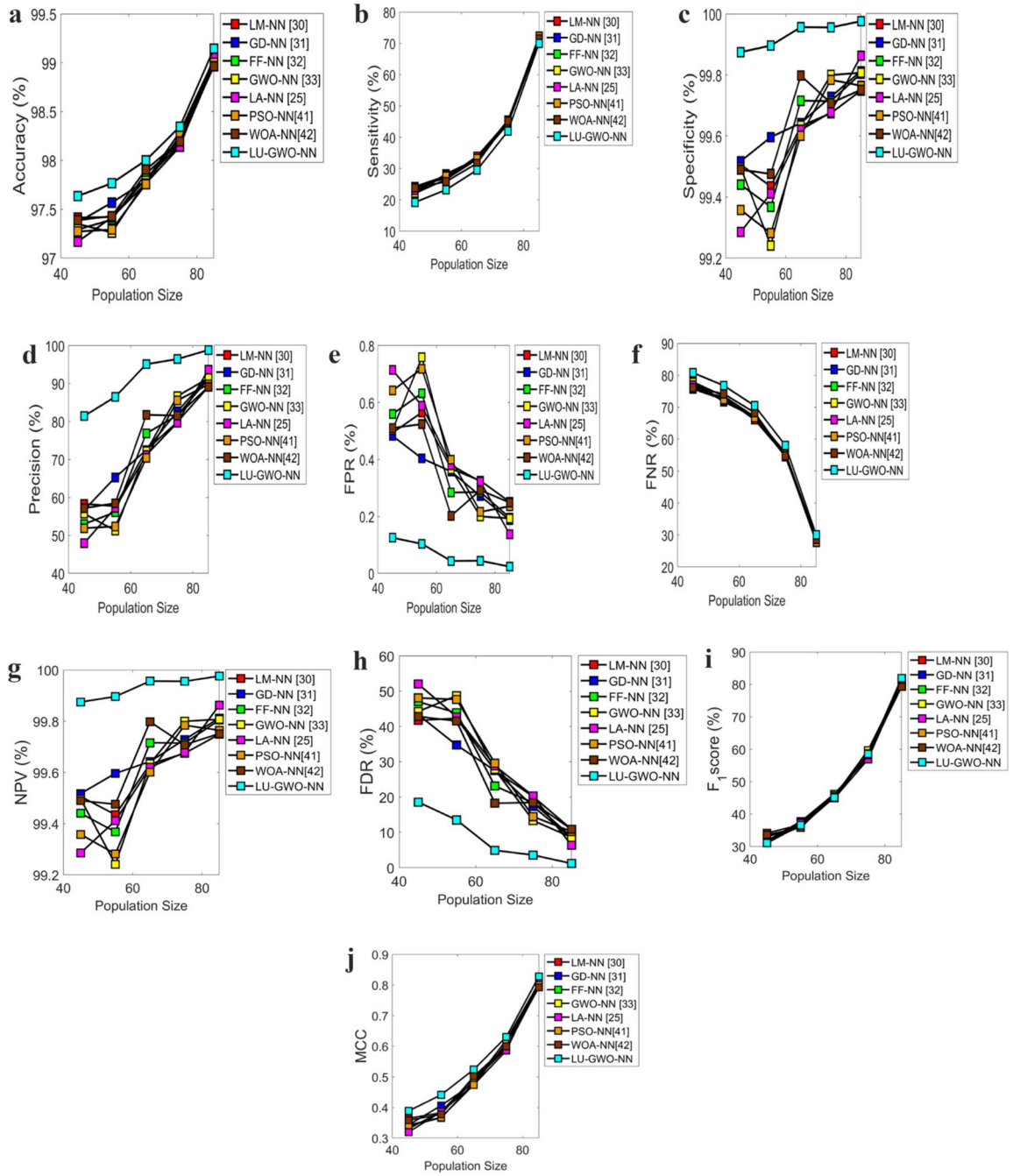


**Fig. 5** Performance analysis of the proposed and conventional handwritten character recognition models for 1st test case in terms of **a** accuracy, **b** sensitivity, **c** specificity, **d** precision, **e** FPR, **f** FNR, **g** NPV, **h** FDR, **i** F1-score, **j** MCC

### 6.4 Statistical analysis

Table 2 depicts the Statistical analysis of the proposed and conventional handwritten character recognition models for three test cases in terms of P-Test and T-Test.

### 6.5 Overall performance analysis

The overall performance analysis of the adopted LU-GWO-NN algorithm for the enhanced handwritten CR model is given in Tables 3, 4 and 5 for three test cases. From Table 3, on considering the 1st test case, the suggested model is

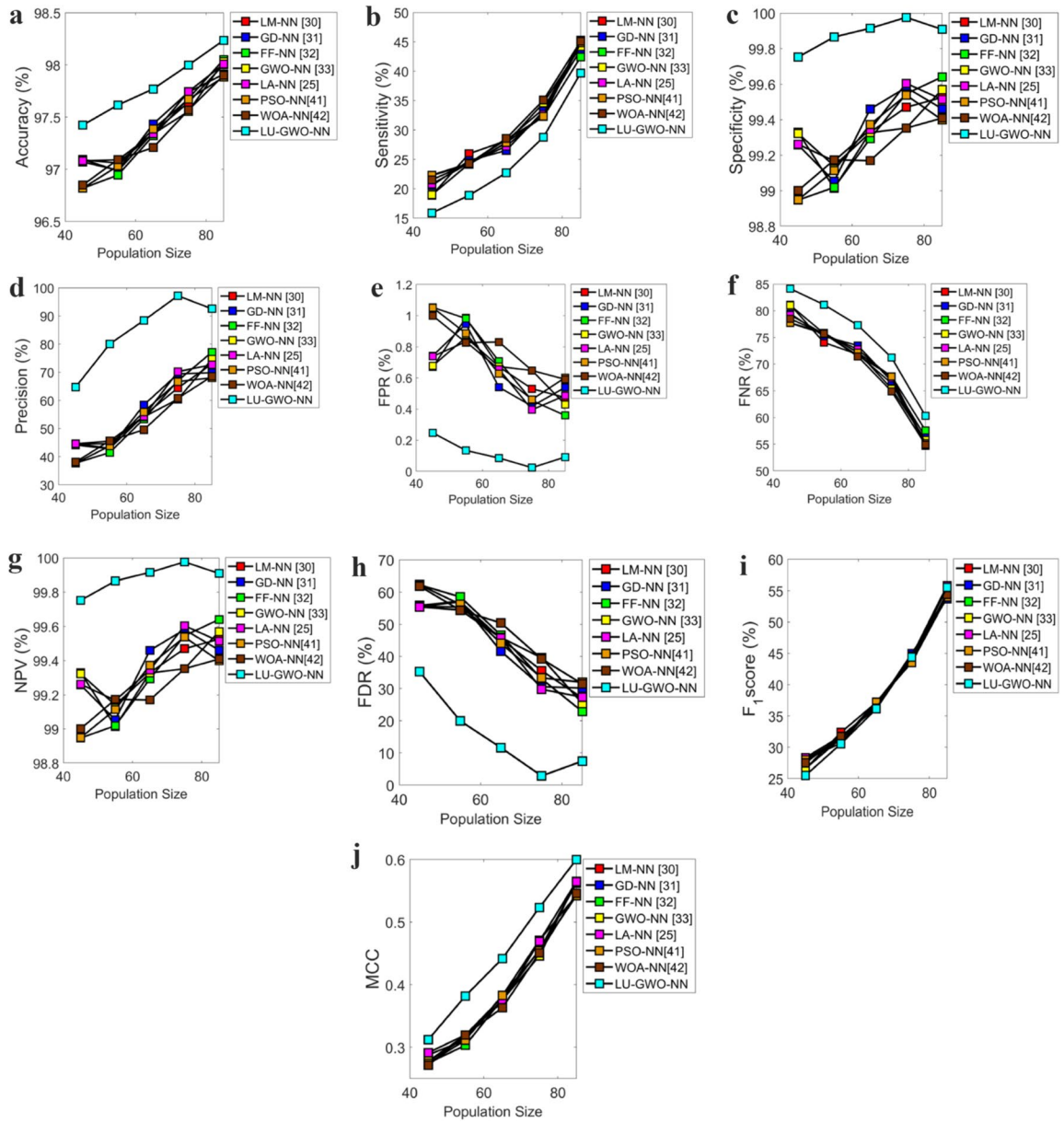


**Fig. 6** Performance analysis of the proposed and conventional handwritten character recognition models for 2nd test case in terms of **a** accuracy, **b** sensitivity, **c** specificity, **d** precision, **e** FPR, **f** FNR, **g** NPV, **h** FDR, **i** F1-score, **j** MCC

0.189% better than LM-NN, 0.13% better than GD-NN, 0.172% superior to FF-NN, 5.49% enhanced than GWO-NN, 5.34% better than PSO-NN and WOA-NN and 0.21% better than LA-NN approaches in terms of accuracy. The specificity of the adopted LU-GWO-NN scheme is 0.28% superior to LM-NN, 0.23% superior to GD-NN, 0.24% superior to FF-NN, 0.16% superior to GWO-NN and 0.28% superior to LA-NN techniques. The FPR of the presented scheme is 86.36% superior to LM-NN, 83.64% superior to GD-NN, 84.46% superior to FF-NN, 77.8% superior to GWO-NN and 86.22% better than LA-NN, PSO-NN and WOA-NN models. In addition, the NPV of the proposed LU-GWO-NN scheme is 0.28% superior to LM-NN, 0.23% superior to GD-NN, 0.24% superior to FF-NN, 0.16% superior to GWO-NN and 0.29% superior to LA-NN algorithms. Moreover, the MCC of the offered LU-GWO-NN scheme is 6.42% better than LM-NN, 4.3% superior to GD-NN, 6.14% superior to FF-NN, 1.56% superior to GWO-NN and 7.62% superior to PSO-NN, WOA-NN LA-NN approaches. Therefore, the enhancement of the proposed LU-GWO-NN scheme has

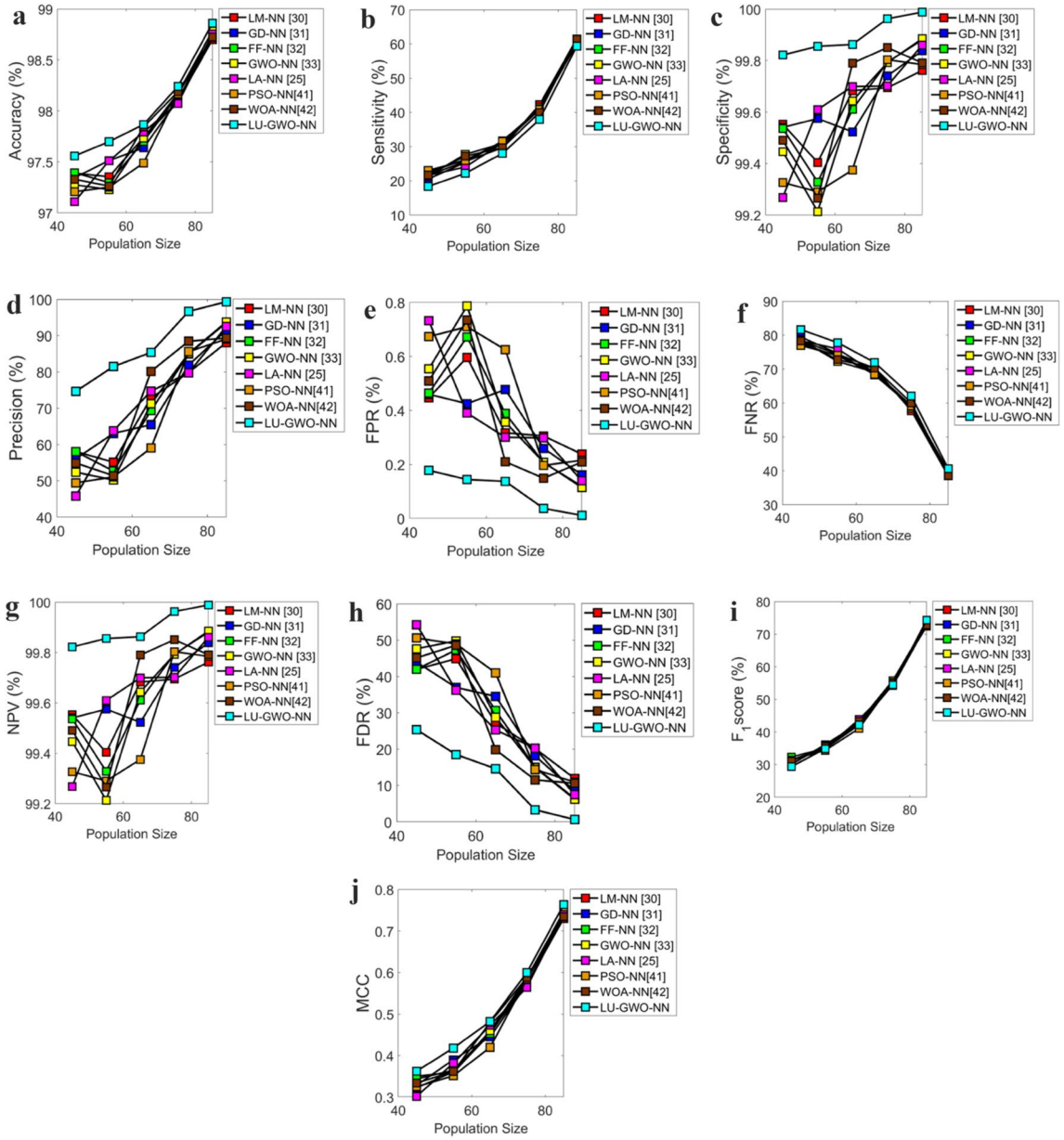


**Fig. 7** Performance analysis of the proposed and conventional handwritten character recognition models for 3rd test case in terms of **a** accuracy, **b** sensitivity, **c** specificity, **d** precision, **e** FPR, **f** FNR, **g** NPV, **h** FDR, **i** F1-score, **j** MCC

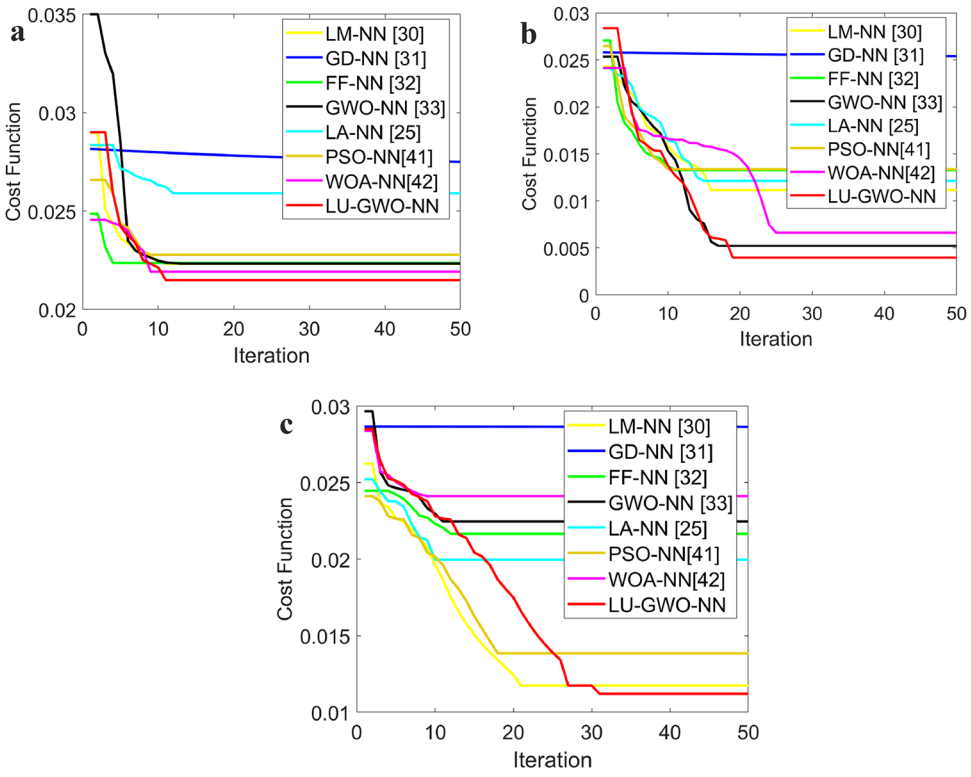


**Fig. 8** Convergence analysis **a** test case 1, **b** test case 2, **c** test case 3

been validated successfully in terms of the overall performance analysis.

## 7 Conclusion

This paper has presented the handwritten CR using a proposed intelligent approach called LU-GWO-NN model. The pre-processing of scanned document was done by a series of steps like RGB to gray, thresholding, and complement of image, morphological operations, linearization and noise removal using MF. Further, k-means clustering was used for segmenting the particular character. Further, the features namely, KDE, FCC and neighborhood distance weight based on Delaunay function were extracted from the segmented image. The process of classification was done by deploying the NN model, in which the weights were optimized by deploying a hybrid optimization known as LU-GWO. Finally, the implemented LU-GWO-NN was distinguished over other conventional schemes, and the outcomes were examined in terms of varied measures. From the analysis, the suggested model was 0.189% better than LM-NN, 0.13% better than GD-NN, 0.172% superior to FF-NN, 5.49% enhanced than GWO-NN and 0.21% better than PSO-NN,

**Table 2** Statistical analysis of the proposed and conventional handwritten character recognition models for three test cases in terms of *P*-Test and *T*-Test

| Methods | *P*-Test | *T*-Test |
|---|---|---|
| Test case 1 | | |
| LM-NN | 0.0096197 | 4.6558 |
| GD-NN | 0.011473 | 4.4243 |
| FF-NN | 0.010205 | 4.5773 |
| GWO-NN | 0.010404 | 4.5518 |
| LA-NN | 0.0097903 | 4.6323 |
| PSO-NN | 0.015856 | 4.0206 |
| WOA-NN | 0.0096296 | 4.6544 |
| LU-GWO-NN | 0.010753 | 4.5085 |
| Test case 2 | | |
| LM-NN | 0.0021701 | 7.0189 |
| GD-NN | 0.0018042 | 7.3722 |
| FF-NN | 0.001091 | 8.4165 |
| GWO-NN | 0.0023698 | 6.8557 |
| LA-NN | 0.0021474 | 7.0386 |
| PSO-NN | 0.0039533 | 5.9705 |
| WOA-NN | 0.001664 | 7.5319 |
| LU-GWO-NN | 0.0018405 | 7.3333 |
| Test case 3 | | |
| LM-NN | 0.0074407 | 5.0094 |
| GD-NN | 0.0071601 | 5.0641 |
| FF-NN | 0.0060531 | 5.3083 |
| GWO-NN | 0.0059165 | 5.3422 |
| LA-NN | 0.0070453 | 5.0872 |
| PSO-NN | 0.010624 | 4.5242 |
| WOA-NN | 0.0061537 | 5.2839 |
| LU-GWO-NN | 0.0068036 | 5.1374 |

WOA-NN and LA-NN approaches in terms of accuracy. Therefore, the development of the presented LU-GWO-NN scheme was verified by the simulation analysis, and it was substantiated that the offered model could offer better outcomes when compared with the conventional schemes in recognizing certain characters. In Future, the work will be further extended to recognize the word or entire document. In this research, individual character image dataset only given as the input. In future research, the words or sentence based images will be used for character recognition which will improve the robustness of the proposed framework.

**Table 3** Overall performance analysis of the adopted and conventional handwritten character recognition models for test case 1

| Measures | LM-NN [30] | GD-NN [31] | FF-NN [32] | GWO -NN [26] | PSO-NN [48] | WOA-NN [49] | LA-NN [25] | LU-GWO-NN |
|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.9816 | 0.9821 | 0.9817 | 0.9829 | 0.98253 | 0.98194 | 0.98139 | 0.98346 |
| Sensitivity | 0.4516 | 0.4533 | 0.4438 | 0.455 | 0.44667 | 0.45222 | 0.44278 | 0.42 |
| Specificity | 0.99675 | 0.9972 | 0.99714 | 0.998 | 0.99784 | 0.99708 | 0.99678 | 0.99956 |
| Precision | 0.79862 | 0.82675 | 0.81614 | 0.8666 | 0.85532 | 0.81563 | 0.797 | 0.96429 |
| FPR | 0.00325 | 0.00271 | 0.0028 | 0.002 | 0.0021587 | 0.0029206 | 0.003222 | 0.000444 |
| FNR | 0.54833 | 0.54667 | 0.55611 | 0.545 | 0.55333 | 0.54778 | 0.55722 | 0.58 |
| NPV | 0.99675 | 0.99729 | 0.99714 | 0.998 | 0.99784 | 0.99708 | 0.99678 | 0.99956 |
| FDR | 0.20138 | 0.17325 | 0.18386 | 0.13333 | 0.14468 | 0.18437 | 0.203 | 0.035714 |
| F1-score | 0.577 | 0.58558 | 0.57503 | 0.59672 | 0.58686 | 0.58184 | 0.56929 | 0.58514 |
| MCC | 0.5926 | 0.60465 | 0.59416 | 0.621 | 0.61095 | 0.5996 | 0.58602 | 0.63066 |

**Table 4** Overall performance analysis of the adopted and conventional handwritten character recognition models for test case 2

| Measures | LM-NN [30] | GD-NN [31] | FF-NN [32] | GWO -NN [26] | PSO-NN [48] | WOA-NN [49] | LA-NN [25] | LU-GWO-NN |
|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.9764 | 0.97739 | 0.97677 | 0.97554 | 0.97671 | 0.97568 | 0.97744 | 0.97999 |
| Sensitivity | 0.33612 | 0.33232 | 0.32281 | 0.34601 | 0.32319 | 0.35095 | 0.32624 | 0.28783 |
| Specificity | 0.9947 | 0.99582 | 0.99546 | 0.99353 | 0.99538 | 0.99353 | 0.99605 | 0.99976 |
| Precision | 0.64431 | 0.6942 | 0.67009 | 0.60425 | 0.66667 | 0.60764 | 0.70213 | 0.97176 |
| FPR | 0.005302 | 0.004183 | 0.004541 | 0.006475 | 0.0046171 | 0.0064747 | 0.003954 | 0.000239 |
| FNR | 0.66388 | 0.66768 | 0.67719 | 0.65399 | 0.67681 | 0.64905 | 0.67376 | 0.71217 |
| NPV | 0.9947 | 0.99582 | 0.99546 | 0.99353 | 0.99538 | 0.99353 | 0.99605 | 0.99976 |
| FDR | 0.35569 | 0.3058 | 0.32991 | 0.39575 | 0.33333 | 0.39236 | 0.29787 | 0.028241 |
| F1-score | 0.44178 | 0.44947 | 0.43572 | 0.44004 | 0.43534 | 0.44493 | 0.44548 | 0.44412 |
| MCC | 0.45493 | 0.47077 | 0.45519 | 0.44598 | 0.45422 | 0.45056 | 0.46922 | 0.5232 |

**Table 5** Overall performance analysis of the adopted and conventional handwritten character recognition models for test case 3

| Measures | LM-NN [30] | GD-NN [31] | FF-NN [32] | GWO -NN [26] | PSO-NN [48] | WOA-NN [49] | LA-NN [25] | LU-GWO-NN |
|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.98099 | 0.98109 | 0.98155 | 0.98173 | 0.98166 | 0.98189 | 0.98073 | 0.9824 |
| Sensitivity | 0.42269 | 0.41014 | 0.40813 | 0.41516 | 0.40863 | 0.4001 | 0.41064 | 0.37952 |
| Specificity | 0.99694 | 0.9974 | 0.99793 | 0.99792 | 0.99803 | 0.99851 | 0.99702 | 0.99963 |
| Precision | 0.7981 | 0.81864 | 0.84953 | 0.85082 | 0.85594 | 0.88457 | 0.79727 | 0.96675 |
| FPR | 0.003055 | 0.002596 | 0.002065 | 0.00208 | 0.001965 | 0.0014917 | 0.002983 | 0.000373 |
| FNR | 0.57731 | 0.58986 | 0.59187 | 0.58484 | 0.59137 | 0.5999 | 0.58936 | 0.62048 |
| NPV | 0.99694 | 0.9974 | 0.99793 | 0.99792 | 0.99803 | 0.99851 | 0.99702 | 0.99963 |
| FDR | 0.2019 | 0.18136 | 0.15047 | 0.14918 | 0.14406 | 0.11543 | 0.20273 | 0.033248 |
| F1-score | 0.55267 | 0.54649 | 0.55137 | 0.55803 | 0.55318 | 0.55099 | 0.54208 | 0.54506 |
| MCC | 0.57279 | 0.57172 | 0.58155 | 0.58707 | 0.58422 | 0.58811 | 0.56413 | 0.59994 |

## References


1. Boufenar C, Kerboua A, Batouche M (2018) Investigation on deep learning for off-line handwritten Arabic character recognition. Cogn Syst Res 50:180–195
2. Lin D, Lin F, Lv Y, Cai F, Cao D (2018) Chinese Character CAPTCHA Recognition and performance estimation via deep neural network. Neurocomputing 288:11–19
3. Shi C, Wang Y, Jia F, He K, Xiao B (2017) Fisher vector for scene character recognition: a comprehensive evaluation. Pattern Recognit 72:1–14
4. Qu X, Wang W, Lu K, Zhou J (2018) Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network. Pattern Recognit Lett 111:9–15
5. Bawane P, Gadariye S, Chaturvedi S, Khurshid AA (2018) Object and character recognition using spiking neural network. Mater Today Proc 5(1):360–366s
6. Zarro RD, Anwer MA (2017) Recognition-based online Kurdish character recognition using hidden Markov model and harmony search. Eng Sci Technol Int J 20(2):783–794
7. Mhiri M, Desrosiers C, Cheriet M (2018) Convolutional pyramid of bidirectional character sequences for the recognition of handwritten words. Pattern Recognit Lett 111:87–93
8. Zheng S, Zeng X, Lin G, Zhao C, Xiong L (2016) Sunspot drawings handwritten character recognition method based on deep learning. New Astron 45:54–59
9. Cilia ND, De Stefano C, Fontanella F, di Freca AS (2018) A ranking-based feature selection approach for handwritten character recognition. Pattern Recognit Lett 121:77–86
10. Qu X, Wang W, Lu K, Zhou J (2018) In-air handwritten Chinese character recognition with locality-sensitive sparse representation toward optimized prototype classifier. Pattern Recognit 78:267–276
11. Wei X, Lu S, Lu Y (2018) Compact MQDF classifiers using sparse coding for handwritten Chinese character recognition. Pattern Recognit 76:679–690
12. Pramanik R, Bag S (2018) Shape decomposition-based handwritten compound character recognition for Bangla OCR. J Vis Commun Image Represent 50:123–134
13. Sohal JS (2016) Improvement of artificial neural network based character recognition system, using SciLab. Optik 127(22):10510–10518
14. Wang Y, Shi C, Wang C, Xiao B, Qi C (2017) Multi-order co-occurrence activations encoded with Fisher Vector for scene character recognition. Pattern Recognition Letters 97:69–76
15. Phangtriastu MR, Harefa J, Tanoto DF (2017) Comparison between neural network and support vector machine in optical character recognition. Procedia Comput Sci 116:351–357

16. Chang Y, Su F, Tzeng S, Ko H, Yang C (2014) The contribution of rapid automatized naming to Chinese character recognition. Learn Individ Differ 34:43–50
17. Tian S, Bhattacharya U, Lu S, Su B, Tan CL (2016) Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. Pattern Recognit 51:125–134
18. Xiao X, Jin L, Yang Y, Yang W, Chang T (2017) Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. Pattern Recognit 72:72–81
19. Yao C, Cheng G (2016) Approximative Bayes optimality linear discriminant analysis for Chinese handwriting character recognition. Neurocomputing 207:346–353
20. Rahman A, Verma B (2013) Effect of ensemble classifier composition on offline cursive character recognition. Inf Process Manage 49(4):852–864
21. Singla SK, Yadav RK (2014) Optical character recognition based speech synthesis system using LabVIEW. J Appl Res Technol 12(5):919–926
22. Bostik O, Klecka J (2018) Recognition of CAPTCHA characters by supervised machine learning algorithms. IFAC-PapersOnLine 51(6):208–213
23. Guruprasad P, Majumdar J (2016) Multimodal recognition framework: an accurate and powerful Nandinagari handwritten character recognition model. Procedia Comput Sci 89:836–844
24. Perrini F, Lombardo L, Arreghini A, Medori S, Siciliani G (2016) Caries prevention during orthodontic treatment: In-vivo assessment of high-fluoride varnish to prevent white spot lesions. Am J Orthod Dentofac Orthop 149(2):238–243
25. Lombardo L, Carinci F, Martini M, Gemmati D, Nardone M, Siciliani G (2016) Quantitive evaluation of dentin sialoprotein (DSP) using microbeads-A potential early marker of root resorption. ORAL Implantol 9(3):132
26. Zhou M-K, Zhang X-Y, Yin F, Liu C-L (2016) Discriminative quadratic feature learning for handwritten Chinese character recognition. Pattern Recognit 49:7–18
27. Rajakumar BR (2012) The Lion's algorithm: a new nature-inspired search algorithm. Procedia Technol 6:126–135
28. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
29. Thakur A, Kaur A (2019) Devanagari handwritten character recognition using neural network. Int J Sci Technol Res 8(10)
30. Puri S, Singh SP (2019) An efficient Devanagari character classification in printed and handwritten documents using SVM. Procedia Comput Sci 152:111–121
31. Khaire P, Kumar P, Imran J (2018) Combining CNN streams of RGB-D and skeletal data for human activity recognition. Pattern Recognit Lett 115:107–116
32. Li X, Wang P, Xu X-J, Xiao G (2019) Universal behavior of the linear threshold model on weighted networks. J Parallel Distrib Comput 123:223–229
33. Kumbhar PG, Holambe SN (2015) A review of image thresholding techniques. Int J Adv Res Comput Sci Softw Eng 5(6)
34. Sharp TH, Faas FGA, Koster AJ, Gros P (2017) Imaging complement by phase-plate cryo-electron tomography from initiation to pore formation. J Struct Biol 197(2):155–162
35. Yuan C, Li Y (2015) Switching median and morphological filter for impulse noise removal from digital images. Optik 126(18):1598–1601
36. Dhanachandra N, Manglem K, Chanu YJ (2015) Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Comput Sci 54:764–771
37. Shekar BH, Uma KP (2015) Kirsch directional derivatives based shot boundary detection: an efficient and accurate method. Procedia Comput Sci 58:565–571
38. Žalik B, Mongus D, Lukač N, Žalik KR (2018) Efficient chain code compression with interpolative coding. Inf Sci 439–440:39–49
39. Normand N, Strand R, Evenou P, Arlicot A (2013) Minimal-delay distance transform for neighborhood-sequence distances in 2D and 3D. Comput Vis Image Underst 117(4):409–417
40. Mohan Y, Chee SS, Xin DKP, Foong LP (2016) Artificial neural network for classification of depressive and normal in EEG. In: 2016 IEEE EMBS conference on biomedical engineering and sciences (IECBES)
41. Rajakumar BR (2014) Lion algorithm for standard and large scale bilinear system identification: a global optimization based on Lion's social behavior. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 2116–2123
42. Boothalingam R (2018) Optimization using lion algorithm: a biological inspiration from lion's social behavior. Evol Intell 11(1):31–52
43. Brammya DTA, Deepa TA (2019) Job Scheduling in cloud environment using lion algorithm. J Netw Commun Syst 2(1):1–14
44. Roy RG (2019) Rescheduling based congestion management method using hybrid grey wolf optimization: grasshopper optimization algorithm in power system. J Comput Mech Power Syst Control 2(1):9–18
45. Fan J, Zeng J (2013) A Levenberg–Marquardt algorithm with correction for singular system of nonlinear equations. Appl Math Comput 219(17):9438–9446
46. Kobayashi M (2017) Gradient descent learning for quaternionic Hopfield neural networks. Neurocomputing 260:174–179
47. Wang H, Wang W, Zhou X, Sun H, Cui Z (2017) Firefly algorithm with neighborhood attraction. Inf Sci 382–383:374–387
48. Zanwar SR, Vaidya NS, Bhuyar DL, Narote SP (2020) Feature extraction methods for handwritten character recognition. Int J Adv Sci Technol 29(8s):5154–5167
49. Sahlol AT, Abd-Elaziz M, Al-Qaness MAA, Kim S (2020) Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set. IEEE Access 8:23011–23021