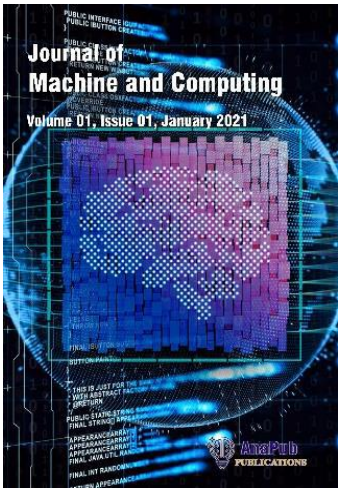
Journal Pre-proof

Firefly Algorithm with Cauchy Crow Search Algorithm based Feature Selection for English Character Recognition



Snehal Prabhakarrao Dongre and Dharmpal D Doye

DOI: 10.53759/7669/jmc202505187

Reference: JMC202505187

Journal: Journal of Machine and Computing.

Received 08 March 2025

Revised from 12 July 2025

Accepted 04 August 2025

Please cite this article as: Snehal Prabhakarrao Dongre and Dharmpal D Doye, "Firefly Algorithm with Cauchy Crow Search Algorithm based Feature Selection for English Character Recognition", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505187.

Firefly Algorithm with Cauchy Crow Search Algorithm based Feature Selection for English Character Recognition

[1,*]Snehal Prabhakarrao Dongre, [2]Dharmpal D. Doye

Department of Computer Science & Engineering, Shri Guru Gobind Singhji Institute of Engineering & Technology, Vishnupuri, Nanded (Maharashtra State) 431606, India

Department of Electronics & Telecommunication Engineering, Shri Guru Gobind Singhji Institute of Engineering & Technology, Vishnupuri, Nanded (Maharashtra State) 431606, India

[1]snehal.dongre@gmail.com, [1]spdongare@sggs.ac.in, [2]dddoye@sggs.ac.in

Corresponding author: Snehal Prabhakarrao Dongre (snehal.dongre@gmail.com)

Abstract

In the present scenario, the identification of English characters from the synthesized, natural, and handwritten images is considered an emerging problem in the researchers' community. The variation in writing style, size of the text, orientation of the text, complex backgrounds, and lower image resolution, along with irrelevant features make character recognition a challenging task. Therefore, a novel automated model of feature selection is implemented in this manuscript for Optical Character Recognition (OCR). In this research, the Firefly Algorithm with Improved Crow Search Algorithm (FAICSA) is proposed for selecting optimal feature vectors for enhancing the classification accuracy. This OCR considers the images acquired from Chars74K and real-time datasets which are pre-processed by implementing binarization and normalization techniques. From the pre-processed real-time images, the English characters are precisely segmented by performing morphological operations known as erosion and dilation. Moreover, the significant features from the images are extracted by using the Local Binary Pattern (LBP), Zernike Moments (ZM), Stroke Width Transform (SWT), and ResNet-18 model. At last, the selected optimal feature vectors from FAICSA are given to the stacked autoencoder model for effective OCR. The proposed FAICSA based OCR is analysed using Matthew's Correlation Coefficient (MCC), sensitivity, Positive Predictive Value (PPV), accuracy, and specificity. The numerical examination states that the FAICSA-stacked autoencoder model attained higher recognition accuracies of 99.64% and 92.06% on the Chars74K and real-time datasets, which are superior values in contrast to those measured for conventional machine-learning classifiers and optimization algorithms.

Keywords: Binarization, Character recognition, Crow search algorithm, Firefly algorithm, Normalization, Stacked Autoencoder.

1. Introduction

In recent decades, huge amounts of images have been generated due to the rapid growth of portable devices like mobile phones, cameras, etc. [1-2]. The text extraction from natural images provides semantic information for several real-time applications like language translation [3], image retrieval [4-5], robot and unmanned vehicle navigation [6], guidance for visually impaired people [7-8], etc. Generally, OCR techniques are utilized for recognizing texts from scanned documents, wherein the texts are well formatted and recorded in well-controlled environments [9]. The traditional OCR techniques achieve recognition accuracy in scanned document images, but have limited execution in natural scene images. The performance of traditional OCR techniques is degraded in natural scene images due to the specifications of variations in font sizes, colours, writing styles, complex backgrounds, and aspect ratios [10-12]. Whereas, the characters in the scanned document images have plain/clean backgrounds, structured orientations, sizes, and colours [13-14]. Currently, numerous studies are performed on scanned document images, and only some studies are carried out on handwritten, synthesized, and natural images [15-18].

This research considers meta heuristics based optimization for feature selection. Metaheuristic is essential tool which offer plans to generate effective optimization algorithms which solves the difficult real world issues based on the searching of optimum solution in specific circumstances. This states that particular solution is not definite, but the algorithm provides one of the best solutions [19]. Some of the feature selection approaches are Hybrid Fireflies-Particle Swarm Optimization (HFPSO) [20], Fish-based Position of Marine Predators and Forest Optimisation (FP-MPFO) [21] and Self-Improved Flower Pollination Algorithm (SI-FPA) [22]. The HFPSO, FP-

MPFO and SI-FPA affects the OCR because it lacks in dynamic adaption, required parameter tuning and sensitive to initial conditions respectively. The FA utilized the combination of attraction and randomization, confirming that new solutions are explored in searching process. Simultaneously, the fireflies moved in the direction of the best solution which leads to avoid the premature convergence (that happens in the Crow Search Algorithm (CSA)) and increases the convergence. The incorporation of firefly attraction confirms that the optimization does not faces the local minima, confirming it to break free from suboptimal solutions and explore wide regions in the solution space.

The character recognition is considered as a challenging task because of the changes in the writing style, text size and orientation, lower image resolution, complex backgrounds and most specifically existence of irrelevant features. The motivation of this work is to overcome the above specified shortcomings for an effective OCR. The primary objective of this research manuscript is to propose an effective automated model for English character recognition on handwritten, synthesized, and natural images. In that, an appropriate pre-processing along with effective feature extraction and optimization based feature selection is designed for improving the recognition of English characters.

The major contributions of this study are pointed out below:

- Binarization and normalization techniques are used to enhance the contrast of the acquired images. The major advantages of utilizing binarization and normalization techniques are: (i) an increase in model efficiency and (ii) a decrease in computational load.
- The significant features from the images are extracted by using the LBP, ZM, SWT and ResNet-18. In that, the LBP is chosen to acquire visual characteristics to differentiate the text region from non-text regions, ZM is chosen to acquire the shape and contour information, SWT is chosen to extract the information of stroke width, and ResNet-18 is chosen to obtain hierarchical features.
- The proposed hybrid optimization algorithm: FAICSA chooses optimal discriminative deep vectors. The dimensionality reduction lessens the complexity and computational time of the model. The proposed FAICSA consumes a computational time of 56.09 and 58[illegible] seconds for the Chars74K and real-time datasets, which are minimum when related to the conventional optimization algorithms.

This manuscript is further presented as follows: literature survey of the existing articles related to character recognition is mentioned in Section 2, while the details about undertaken methods, evaluation results, conclusion, and future work of this work are correspondingly described in Sections 3, 4, and 5.

2. Literature survey

The existing researches related to the character recognition are given in this section along with their advantages and limitations.

Zanwar, S.R et al. [20] developed the Novel Naive Propagation (NNP) Classification to perform the character recognition. The NNP was integrated with the Harmonized Independent Component Analysis (HICA) to perform the feature extraction while the Hybrid Fireflies-Particle Swarm Optimization (HFPSO) was used to select the features. Further, the ensemble classifier that combined the naïve bayes propagation classifier with backpropagation and feed forward neural network. The utilization of particle swarm optimization was contributed to the better convergence while the Firefly was used to enhance the exploitation. The developed HFPSO was lacking in dynamic adaption which made it ineffective higher dimensional searching space.

Parashiva murthy, S.P.T. and Rajashekararadhya, S.V [21] presented the Hybrid Deep Learning Network (HDLNet) for the combination of Attention-Based Deep Temporal Convolution Network (ADTCN) and Dense Long-Short Memory (DLSTM) for recognizing the characters. The different features such as morphological, texture and geometric were considered during feature extraction. Subsequently, the features were weighted by using Fish based Position of Marine Predators and Forest Optimisation (FP-MPFO) to perform the classification using HDLNet. The FP-MPFO based feature optimization was used to solve the issue of misclassification and overfitting issues. This FP-MPFO was struggled in balancing the exploration and exploitation that frequently required the parameter tuning.

Devi, S.N. and Fatima, N.S [22] developed the OCR to address the issues of orientation, text and size variations. The texts were segmented by using the Canny Edge Detection and Efficient and Accurate Scene Text (EAST) was used to ensure the text localization. A low and high level features were extracted by Inception V3 while Long Short-Term Memory (LSTM) utilized for extracting the chronological information. The Self-Improved Flower Pollination Algorithm (SI-FPA) was used to choose the features with the Principal Component Analysis (PCA). Additionally, the integration of Recurrent and transformer neural network were utilized for performing the text recognition which increased the network advantages in processing contextual and sequential data. This SI-FPA

was sensitive to the initialized conditions and it is subjected to the lose the important features because of disparaging dimensionality reduction.

Ptucha et al. [23] implemented an effective character recognition framework based on Fully Convolutional Neural Network (FCNN). Unlike the lexicon-based models, the presented FCNN model effectively recognized the infinite symbol blocks and common words such as acronyms, phone numbers, and surnames. A probabilistic character error rate was integrated with the FCNN model for correcting the errant word blocks. In comparison to the existing models, the presented FCNN model obtained superior results in handwritten character recognition. Correspondingly, Anand et al. [24] developed the LeNET model for an effective recognition of handwritten characters. However, the factors like shape variance, broken edges, skewing, and touching characters, degrade the performance of the FCNN and LeNET models in character recognition.

Lee et al. [25] integrated a shallow deep CNN model with an improved Local Binary Pattern (LBP) descriptor for enhancing the performance of character learning. A weighted depth-wise separable fire module and squeeze net architecture were used in the shallow deep CNN model for improving the recognition ability and reducing the parameters of the overall network. Additionally, shuffle-Net architecture was utilized in the deep network for decreasing the computational cost of the network. The integration of several architectures and modules in the shallow deep CNN model increased the time complexity. Similarly, Yang and Yang implemented an improved LBP descriptor for real-time OCR. The use of handcrafted features increased the vector's semantic gap, which potentially degraded the performance of OCR.

3. Methodology

The proposed FAICSA-stacked autoencoder model includes six phases in English character recognition. These are **dataset description:** Chars74K and real-time datasets, **preprocessing:** image normalization and Binarization, **segmentation:** Morphological operation, **feature extraction:** LBP, ZM, SWT and ResNet-18, **feature optimization:** FAICSA, and **character recognition:** stacked autoencoder model. The flow diagram of the FAICSA-stacked autoencoder model is presented in Figure 1.

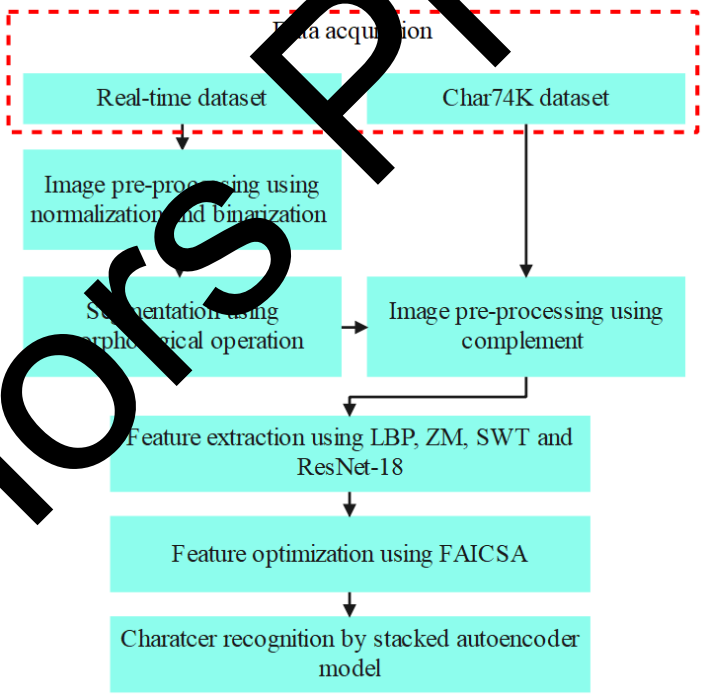


Figure 1. Flow diagram of the FAICSA-stacked autoencoder model

3.1 Dataset description

The proposed model's efficacy is analysed on the Chars74K and real-time datasets. The real-time dataset has English document images and handwritten images. In addition, it has 52 classes belonging to smaller and upper cases, and 5200-segmented English letters utilized for experimental analysis. On the other hand, the online Chars74K dataset has the characters of English and Kannada languages. The Latin Script (excluding accents) and Hindu-Arabic numerals are utilized in English language. The statistics of the Chars74K dataset are given below, and the Chars74K dataset is available at http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/.

- Has 62,992 synthesized character images which are recorded from computer fonts.
- Using a tablet or personal computer, 3,410 handwritten characters are obtained.
- Around 7,705 characters are natural images.
- Has 64 classes (A-Z, a-z, and 0-9). Around 74,000 images are available in the Chars74K dataset. The sample-acquired Chars74K and real-time images are shown in Figures 2 and 3.

Figure 2. Sample-acquired Chars74K image

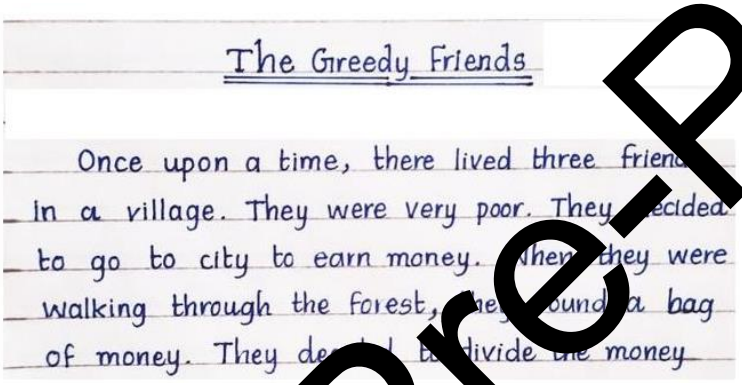


Figure 3. Sample-acquired real-time image

3.2 Pre-processing

In this phase, the acquired images are further denoised by performing binarization, complement and implementing normalization technique . Two different types of pre-processing individually performed for Chars74K images and real-time images due to the nature of the input images. Due to the different characteristics of the data, the pre-processing steps are varied for Chars74K and real-time datasets. For the real time images, the pre-processing is performed by using the following processes of normalization, binarization, segmentation, complementing because it exhibits higher differences in resolution, lighting and background noise. This helps the model acquires appropriate patterns rather than the noise. On the other hand, the Chars74K dataset has handwritten, synthesized and natural images with appropriate formatting, necessitating slight pre-processing i.e., complement. Because, the word or letters in the Chars74K dataset is black (depicts pixel value 0) while the background is in white (depicts pixel value 1), therefore complementing the image makes the word or letters into white that makes appropriate to learn during feature extraction.

3.2.1. Normalization

The image normalization technique is employed for enhancing the contrast of the binary images. The normalization technique converts the n-dimensional gray-scale images $I: \{X \subseteq R^n\} \rightarrow \{Min, \ldots\ldots, Max\}$ into new normalized images $I_N: \{X \subseteq R^n\} \rightarrow \{newMin, \ldots\ldots, newMax\}$. The mathematical formula of the linear normalization technique is presented in equation (1).

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin \quad (1)$$

Where, I_N is denoted as normalized images, $newMax\ and\ newMin$ are represented as new pixel intensity values of normalized images, and Max and Min are indicated as the maximum and minimum pixel intensity values of the binary images I.

3.2.2. Binarization

In the image processing applications, binarization is the process of converting document images into binary-level document images and compliment is used to invert the image colours. The primary objective of binarization is the

separation of letters from the background regions and foreground texts. In this scenario, the Bernsen method is employed for image binarization which uses image contrast $C(i,j)$ for estimating the lowest (I_{min}) and highest pixel intensity values (I_{max}) in the window by utilizing equation (2).

$$C(i,j) = I_{max} - I_{min} \tag{2}$$

Here, the pixel values are divided into background and foreground regions by estimating the local contrast with the threshold value. If the contrast is lesser than the threshold value, the pixels are classified as background regions. Similarly, the pixels are classified as foreground regions when the contrast is higher than the threshold value. Nevertheless, the performance of the Bernsen method is degraded in complex backgrounds.

3.2.3. Segmentation

After normalizing the images, the morphological operations: erosion and dilation are performed on the real-time dataset's images for the precise segmentation of characters. The morphological operator erosion shrinks the image pixels and eliminates unwanted pixels in the boundaries. To perform erosion operation, traversing the element structures is essential and the output pixels of the erosion is computed by utilizing equation (3). Then, the morphological operator, dilation expands the image pixels and adds appropriate pixels in the boundaries. The output pixels of the dilation are estimated by utilizing equation (4) wherein, Fit denotes the retained pixels and Hit represents the eliminated pixels. The sample-segmented characters from the real-time dataset are depicted in Figure 4.

$$Erosion = \{pixel\ (output) = 1 \ \ if\ \{Fit\}\ pixel\ (output) = 0\ \ if\ \{otherwise\}\} \tag{3}$$

$$Dilation = \{pixel\ (output) = 1 \ \ if\ \{Hit\}\ pixel\ (output) = 0\ \ if\ \{otherwise\}\} \tag{4}$$

Figure 4. Sample-segmented characters from the real-time dataset

The steps processed in the morphological segmentation are given as follows:

1. At first, the images is converted into grayscale if it is a color image.
2. Next, the thresholding technique is used for transforming the grayscale image into a binary image. Here, the thresholding is used for separating the foreground (i.e., text) from the background. This thresholding is important for segmentation to simplify the image by decreasing it to two intensity levels for simplifying the further processes.
3. The segmentation is refined and binary image's quality is enhanced by using the morphological operations of erosion and dilation. In that, the small objects are eliminated and boundaries are smoothened using erosion while the dilation is used to fill the gaps and create objects in uniform shape and size.
4. The small objects are eliminated from the binary image used to remove the noise and irrelevant artifacts. This helps minimize the false positive and enhance accuracy in classification.
5. For each object, a unique identifier is assigned by labelling connected components in the binary image which is important for additional processing.
6. The characteristics of each image region i.e., bounding boxes that offer the essential information about the shape, location and size for each segmented object are computed.
An individual character from an original image is extracted to aid for additional processes. The information about bounding box helps obtain the individual characters from the original image. This helps separate the pixels related to each character region.

3.2.4. Complement

The image pixels are categorized into dual pixel collections of white and black. So, the operation of compliment is essential, because in the input data the letters are in black colour, while the background is in white which indicates the colour value of 0 and 1, respectively. Complementing the image improves various aspects of handwriting, being helpful for feature extraction by confirming uniform depiction of characters.

The sample pre-processed images of Chars74K and real-time images are shown in Figures 5 and 6.

Figure 5. Sample pre-processed Chars74K image

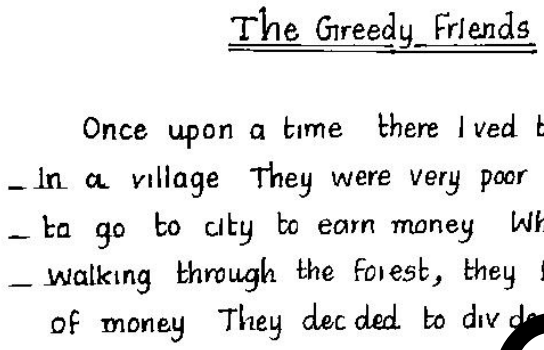


Figure 6. Sample pre-processed real-time image

3.3 Feature extraction

In this phase, four approaches namely, LBP, ZM, SWT and ResNet-18 are used over the pre-processed images for extracting the features. In that, the LBP is used for capturing the visual characteristics appropriate to differentiate the text regions from non-text regions. The contour and shape patterns in text regions are obtained by ZM while the SWT is useful in extracting the stroke width information for an effective differentiation of text from the image. Further, the ResNet-18 is used for learning the rich and hierarchical features from images. The feature extraction is detailed in the below sections.

3.3.1. Local binary pattern

In the LBP approach, the binary number is created for each pixel of pre-processed image that considers the adjacent around every pixel (minimum 3×3). If the neighboring pixel's intensity is higher or equal to 1 according to the central pixel, then the value is fixed as 1 to the respective pixels; otherwise, the value is fixed as 0. Further, the label values are placed together rotationally, creating an 8-bit value using equations (5) and (6).

$$LBP_{R,P} = \sum_{p=0}^{P} \quad s(g_p - g_c) \times 2^P \tag{5}$$

$$s(x) = \{1 \quad x \geq 0 \; 0 \quad x < 0 \tag{6}$$

Where, neighboring radius and the amount of neighborhood pixels of the central pixel are respectively denoted as R and P; the central pixel's brightness intensity is denoted as g_c and neighborhood pixel's brightness intensity is denoted as g_p.

3.3.2. Zernike moment (ZM)

In general, the ZM is developed to overcome the issues of information loss in the Geometric moments. ZM is the moment function utilized to map the images in a group of complex ZM. This denotes the characteristics of image with overlap or redundancy for information among the moments. The discrete form of ZM with order d and repetition m for image size of $N \times N$ is given in equation (7).

$$Z_{d,m} = \frac{d+1}{\lambda_N} \sum_{q=0}^{N-1} \quad \sum_{r=0}^{N-1} \quad f(q,r) R_{d,m}(\rho_{qr}) e^{-jm\theta_{qr}} \tag{7}$$

Where, $0 \leq \rho_{qr} \leq 1$ and normalization factor is denoted as λ_N, real valued one dimensional radial polynomial is denoted as $R_{d,m}$, and image function is denoted as $f(q,r)$. Here, the normalization factor is an amount of pixels positioned in the unit circle by using the mapping transform. Moreover, the transformed distance and phase are denoted as ρ_{qr} and θ_{qr} which are expressed in equations (8) and (9) respectively.

$$\rho_{qr} = \frac{\sqrt{(2q-N+1)^2+(2r-N+1)^2}}{N} \tag{8}$$

$$\theta_{qr} = \left(\frac{N-1-2r}{2q-N+1}\right) \tag{9}$$

3.3.3. Stroke width transform

The stroke width of various image pixels of pre-processed image is discovered using SWT which is widely used for detecting the text features in the images. Generally, SWT is data-dependent and therefore does not require scanning windows or multi-scale calculation. In SWT, each pixel is converted into stroke width when image has color value. Initially, each element is fixed as ∞ in SWT, hence the canny edge detector is used for recovering the strokes. For each pixels of image, a gradient direction (dp) is used in SWT. The dp is computed for pixel p, when the respective pixel's stroke boundary is perpendicular to the direction of the stroke. The aforementioned process is continued until the other pixel q is found. The pixel q's gradient direction is represented as dq. The width $\|p-q\|$ is assigned for various elements of the pixels over the segment $[p,q]$ of SWT resulting image, when the dq direction is opposite to the dp i.e., $dp = -dq \pm \frac{\pi}{6}$. The width value of $\|p-q\|$ is assigned until the existence of minimum value in the pixel p. The ray is discarded when both dp and dq are not in the opposite directions.

3.3.4. ResNet-18

In the segmented images, the discriminative vectors are extracted by implementing ResNet-18 model . It has 18 convolutional layers and learns around 11.7 million parameters from the segmented images. The use of the ResNet-18 model boosts the speed of learning, eliminates redundant data, and improves the model's recognition accuracy.

Further, the features from LBP, ZM, SWT and ResNet-18 are concatenated together to form the overall feature vector. Around 2621 and 2824 deep vectors are extracted from the Chars74K and real-time datasets, which are further given to the optimization algorithm for feature reduction.

3.4 Feature optimization

The extracted vectors are given to FAICSA for dimensionality reduction. In general, handwritten characters show certain variations because of the differenced style of writing, thickness of stroke and noise in the images. From the extracted features, not all features equally contribute to the recognition. Some of the features are irrelevant or noisy, while the remaining features have highly discriminative features. Therefore, the feature selection is used to identify relevant features that help enhance the generalization over unknown data. The inclusion of attractiveness-based movement helps improve the exploitation and exploration capacities in ICSA which is further used to obtain optimum feature subset for an enhanced recognition. In this feature selection phase, the population is generally encoded as binary matrix. Each row depicts a discrete solution i.e., a possible feature subset, and each column depicts whether the respective feature is chosen (1) or not (0). This task is done by considering the threshold-based binarization where the real valued score of feature is validated with the predefined threshold. If the score is higher or equal to the threshold, then the respective feature is selected (1); otherwise, it is not chosen (0).

3.4.1. FA

The FA is considered as one of the many effective bio-inspired metaheuristic-optimization algorithms, which mimics the firefly's flashing behaviour. The FA follows three rules for constructing this algorithm: (i) the fireflies' brightness is considered as the objective function, (ii) fireflies are unisex, and so they are attracted to brighter fireflies, and (iii) the fireflies move randomly if there are no brighter fireflies because the brightness is directly

proportional to attractiveness. Elementary physics states that the distance r is inversely proportional to the intensity of light. The light passes in a medium with light intensity LI and the light absorption coefficient λ varies with a distance r, as mathematically determined in equation (10). As stated in equation (11), the intensity at the source points I_O are combined with equation (10).

$$LI(r) = LI_O e^{-\lambda r} \tag{10}$$

$$LI(r) = LI_O e^{-\lambda r^2} \tag{11}$$

Computation of $e^{-\lambda r^2}$ is harder than that of $1/(1+\lambda r^2)$. So, the intensity is computed utilizing equation (12). The firefly's attractiveness $A(r)$ is mathematically specified in equation (13), where the attractiveness at $r =$ is represented as A_O.

$$LI(r) = \frac{LI_O}{1+\lambda r^2} \tag{12}$$

$$A(r) = \frac{A_O}{1+\lambda r^2} \tag{13}$$

Where, λ denotes light absorption coefficient which defines how rapidly the light intensity decreases along with the distance at the medium. The parameter λ defines the fireflies motion and attractiveness in the optimization. In general, λ is selected empirically or according to the certain features of issue [illegible] as it is chosen between [0.5,1.5] to balance the exploitation and exploration. Lesser λ values leads to slower intensity decay for improving global search while higher value returns the rapid decay to improve local search, therefore the λ is set as 1.

Generally, the firefly located at $y^{i,iter}$ moves towards $(y^{i,iter})'$ when the firefly located at $(y^{i,iter})'$ is brighter when compared to the firefly located at $y^{i,iter}$. The firefly located at $y^{i,iter}$ is updated by utilizing equation (14).

$$y^{i,iter+1} := y^{i,iter} + A_o e^{-\lambda r^2}((y^{i,iter}) - y^{i,iter}) + \alpha\varepsilon \tag{14}$$

Where, iteration is stated as $iter$ and the randomized parameter is represented as α, and the vectors of random numbers are denoted as ε.

3.4.2. ICSA based on Cauchy random number

In a metaheuristic approaches, two distinct criteria such as intensification and diversification are considered for taking optimum results. In conventional CSA, the balance among the diversification and intensification is maintained among two factors such as awareness probability (Ap) and flight length (fl). It depicts that lesser value of flight length ensures local search while higher values ensure global search. Similarly, the lesser value of Ap ensure local search while higher value ensure global search. In general, the value of fl is fixed, before initializing the optimization process and it is same for entire population. In ICSA, the value of fl is dynamically chosen based on the Cauchy random number (E) and it is produced for every test case. Accordingly, the value of fl is interchanged by E that is computed from the inverse mapping of the Cauchy distribution's Cumulative Density Function (CDF) which is shown in equation (15).

$$F(y) = \frac{1}{2} + \frac{1}{\pi} arctan\ arctan\ \left(\frac{y}{t}\right) \tag{15}$$

Where, scaling factor is t, whose value is set as 1 and y is corresponded to an each test case. This operator is used to minimize the solution problems trapped in local minima. Additionally, this used to balance the searching pattern based on the adjustment in the scaling factor. The value of Cauchy random number (E) is acquired by inversing the equation (15). The value of Ap in ICSA corresponded to the lesser threshold which is minimum amount of mutants are destroyed by the test case. This specified whether an amount of mutants destroyed by the test case is sufficient or not for each of new test cases and it is modified for every benchmark. If the crow moved to new location's fitness is higher than the minimum threshold, then it is evaluated with the memorised location's fitness; Otherwise, a test case's random value is chosen. This is used to maintain the balance among the diversification and intensification. Hence, the ICSA improves the capacity of searching by dynamically selecting the fl and fixing the Ap. The mutation is promoted by the Cauchy random number in the ICSA. The Cauchy random number based mutation leads to diverse searching process because of its capacity for generating the outliers and highly important perturbations, that helps in exploring the areas.

ICSA follows the crow's foraging behaviour with Cauchy random numbers. The incorporation of Cauchy random numbers E with the conventional CSA improves the exploration and exploitation ability that helps in obtaining the optimum solutions. By using the concept of mutation testing, the ICSA helps in generating effective test cases.

In ICSA, the test cases t correspond to the population size of the crows. Based on the principles of the CSA, the memory and position are initialized. The memory update is important component in the ICSA, confirming that each crow keeps the best solutions. In the beginning of iterative process, memory of each crow is initialized for storing the initial location in the search space which is denoted in the below equation (16).

$$Memory\ (i) = Position\ (i), \forall i \in Population \tag{16}$$

In every iteration, the location of current crow is evaluated with its memory location based on a fitness function. The memory is updated, when the current location of the crow has the better fitness than their memorized location; Otherwise, there is no update in the memory. This kind of memory update confirms that each crow has the best location's knowledge, confirming an effective exploitation and exploration.

By utilizing the fitness function, the test cases t of the memorized positions are analyzed. A random t is selected and its position $y^{i,iter}$ is computed using equations (17) and (18).

$$y^{i,iter+1} = y^{i,iter} + rand_i \times E \times (m^{j,iter} - y^{i,iter}) rand_j > AP_m^{j,iter} \tag{17}$$

Where,

$$y^{i,iter+1} = a\ random\ position\ otherwise \tag{18}$$

Here, $rand_i$ and $rand_j$ are indicated as the random numbers within the range of zero and one, AP is represented as awareness probability, $m^{j,iter}$ is denoted as a memory of a crow, and an iteration is signified as $iter$.

In the FAICSA, the random position update using equation (18) is replaced by equation (14) that is attractiveness-based movement. The benefits of replacing the random position update are targeted search, improved exploitation, balanced exploration and exploitation and improved convergence. Equation (14) ensures an attractiveness-based update formulated from the FA which allows the crows to move optimal solutions according to the attractiveness of fireflies instead of random motion. Accordingly, the FAICSA obtains more directed searching behaviour, minimizing unwanted randomness and concentrating on appropriate regions in the search space. This kind of searching towards targeted solution is used to enhance the exploitation and to avoid the local optima by balancing the exploration and exploitation. The mitigation of random updates in exploration phase is used to obtain the optimal solutions for enhancing the convergence that is helpful in high-dimensional issues.

After updating the location, the vertical crossover search is used to help some immobile dimension of population (crows) escape from premature convergence, which ensures the FAICSA jumps out of the local optima. It also maximizes the population diversity. In general, vertical crossover is also referred as arithmetic crossover where all the individual in population function two distinct dimensions. Each crow processed in a vertical crossover which only updates one dimension while retaining the remaining dimensions unaffected. This provides an immobile dimensions an opportunity for avoiding the local optima without affecting the remaining optimal dimension. Two distinct dimensions $j1$ and $j2$ are randomly chosen, and the $j1$ dimension of its offspring $y_{vc}^{i,iter+1}$ is generated based on equation (19), while the remaining dimensions are identical as the parent crow.

$$y_{vc}^{i,iter+1} = r_1 \times y^{j1,iter} + (1 - r_1) \times y^{j1,iter} \tag{19}$$

Where, r_1 is a uniformly distributed value among $[0, 1]$. The opposition among the parent crows and offspring is used to retain the optimal crow. This crossover helps the FAICSA to avoid the premature convergence.

The new position's fitness values are superior to the memorized positions, and this procedure is continued until the stopping criteria are met. Hence, the best-memorized positions are considered as the optimum test suites (optimum vectors). The detection score is computed with the help of optimum test suites, and it is mathematically defined in equation (20).

$$Detection\ score = |D(\gamma_d, P_m, T)|/|P_m| \tag{20}$$

Where, the mutated program is represented as P_m, dimensional search space is indicated as D, detection boundary is specified as γ_d, and T is indicated as a set of test cases t. The fitness maximum Relative Error (RE) is computed using equation (21).

$$\tau(P_m(t)) = \{\gamma(P_m, t)\} \tag{21}$$

Where, the maximum RE is denoted as τ. The algorithm of ICSA is determined below;

Algorithm of ICSA

1. Randomly initialize the crow population in the search space.
2. Compute fitness for every crow based on the objective function.
3. Identify the crow with the best fitness as the "leader".
4. Repeat the following steps until a termination criterion is met:
   a. For every crow in the population (excluding the leader):
      - Generate a new candidate solution by modifying the crow's position based on its current position and the leader's position.
      - Apply local search or mutation operators to further explore the solution space.
      - Determine the new crow's position and then, generate a random number and compare it with AP. The crow moves based on equation (17) when the random number is smaller than AP else the crow randomly selects flock crow. Update the crow position based on equation (14).
      - Evaluate the fitness of the new candidate solution.
      - Compare the fitness of the new candidate solution with the crow's current position.
      - If the fitness of the new candidate solution is better, update the crow position to the new solution.

   b. Update the leader by selecting the crow with the best fitness among the population.

   c. Perform population-based exploration and exploitation techniques such as crossover, mutation, or other evolutionary operators based on Cauchy random number.

   d. Evaluate the fitness of the new population.
5. Return the best solution found.

The parameters considered in the FAICSA are stated as follows: number of iterations is 100, maximum generation is 50, randomized parameter α is one, and the light absorption coefficient λ is one. Here, the hyperparameters such as population size and iterations are optimized based on the trial and error approach for both the Chars74K and real-time datasets. The analysis of trial and error approach for FAICSA is shown in Figure 7. Finally, the FAICSA uses the binary encoding to represent the chosen and unchosen features. The threshold based binarization confirms an efficient transformation of real valued updates as binary feature selection.

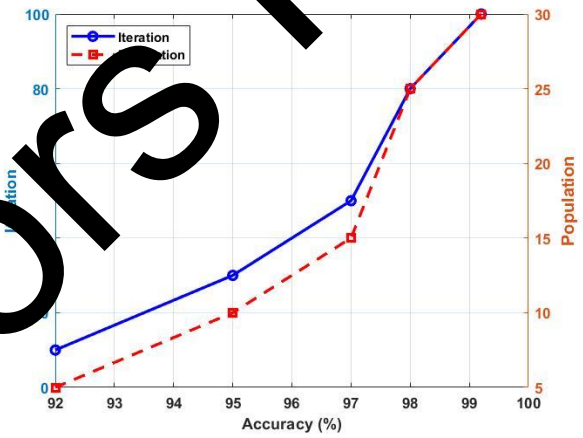


Figure 7. Analysis of trial and error approach for FAICSA

The pseudocode of the FAICSA is stated below:

Pseudocode of the FAICSA

Input: Total population
Output: Best crow position
Initialization
Initialize crow position
Compute crow position
Initialize crow memory

While (iteration<maximum number of iteration)
For all crows,
Select a random crow
Define AP
If (random number>AP)
Update crow position based on equation (17) with mutation based on Cauchy random number
Else
Update crow position based on equation (14)
End if
End for
Check the solution feasibility
Update the crow position based on equation (19) with crossover
Compute Crow's new best position based on fitness value calculated using equation (2
Update the memory
End while
Return: The best solution
Terminate

3.5 Character Recognition

The optimized 1987 and 1875 vectors of Chars74K and real-time datasets are given to the stacked autoencoder model for character recognition. The autoencoder is one of the effective unsupervised learning models, which owns three layers: hidden, output, and input layers. The stacked autoencoder has the capacity of learning hierarchical representations in the input. Therefore, the recognition of characters is enhanced by obtaining complex patterns and relationships among the features. Additionally, the autoencoder includes two training processes: encoding and decoding. Initially, the encoder maps the selected vectors into hidden representations, and the decoder reconstructs the vectors from the hidden representations. The selected deep vectors are considered as $\{x_n\}_{n=1}^{N}$, and the encoding process is mathematically specified in equation (22).

$$h_n = f(W_1 x_n + b_1) \tag{22}$$

Where, $x_n \in R^{m\times 1}$, $\hat{x}_n$ represents the decoder vectors obtained from the output layer, and h_n indicates the hidden encoder vectors which are computed from x_n. Furthermore, b_1 indicates the bias vector, the encoding function is specified as f, and the weight matrix of the encoder is denoted as W_1. Correspondingly, the mathematical formula of the decoding process is indicated in equation (23).

$$\hat{x}_n = g(W_2 h_n + b_2) \tag{23}$$

Where, the bias vector is represented as b_2, the decoding function is indicated as g, and the weight matrix of the decoder is specified as W_2. To reduce the reconstruction error, the parameters in the autoencoder are optimized, and it is mathematically determined in equation (24).

$$\emptyset(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \quad L(x^i, \hat{x}^i) \tag{24}$$

Where, L indicates a loss function $L(x,\hat{x}) = \|x-\hat{x}\|^2$. The important parameters considered in the stacked autoencoder are given as follows: hidden units are 50, hidden layers are 3, sparsity regularization is 1, activation function is ReLU, and maximum epochs are 100. The structure of stacked autoencoder model is graphically mentioned in figure 8, and the steps involved in the stacked autoencoder model are pointed below:

- The 1st autoencoder is trained with the selected deep vectors.
- The learned vectors of the 1st autoencoder are given as the input to the next autoencoder. This process is repeated until the completion of model's training.
- Once the training is completed, a back-propagation algorithm is utilized for minimizing the cost function, so as to achieve fine-tuning. Furthermore, the numerical outcomes of the proposed model are described in section 4.

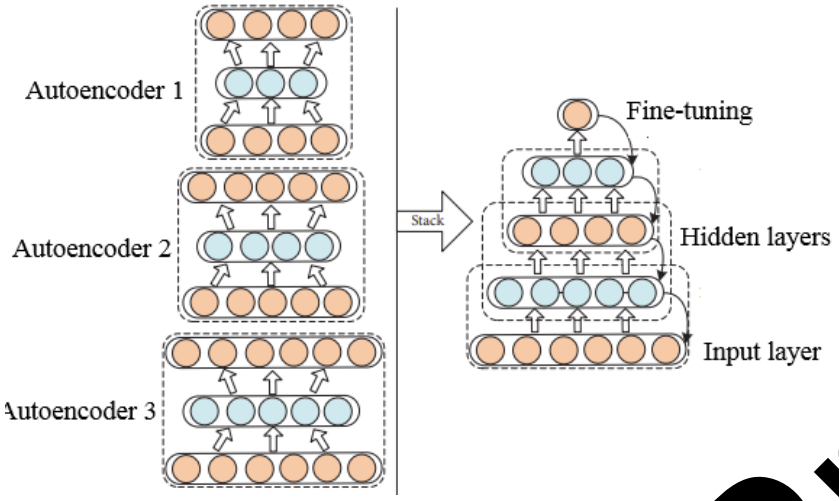


Figure 8. Structure of stacked autoencoder model

4. Results and discussion

The proposed model is implemented by utilizing Matlab R2020b (version 9.9), and is analyzed on a system with a Windows operating system, 64GB random access memory, and Intel i7 12th generation processor. Further, the efficacy of the proposed model is investigated on a benchmark dataset (Chars74K) and a real-time dataset using MCC, sensitivity, PPV, accuracy, and specificity. First, the evaluation metric: MCC computes the correlation between the true classes with the help of the predicted labels. Then, if the stacked autoencoder classifier precisely classifies the positive and negative instances, it gains a higher MCC score. On the other hand, specificity calculates the percentage of actual negative values which are precisely classified. Similarly, sensitivity calculates the percentage of positive values which are precisely classified. The mathematical formulae of MCC, sensitivity, and specificity are defined in equations (25-27).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \times 100 \quad (25)$$

$$Sensitivity = \frac{TP}{FN+TP} \times 100 \quad (26)$$

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (27)$$

In OCR, the PPV is determined as the ratio of true positive values out of total positive values, and accuracy is determined as the percentage of precisely classified instances. The mathematical formulae of the undertaken evaluation metrics: PPV and accuracy, are denoted in equations (28-29). In the confusion matrix, the terms: FP, FN, TP, and TN are defined as false positive values, false negative values, true positive values, and true negative values, respectively.

$$PPV = \frac{TP}{TP+FP} \times 100 \quad (28)$$

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \times 100 \quad (29)$$

4.1. Analysis of Chars74k dataset

Here, the efficacy of the FAICSA-stacked autoencoder model is analyzed on a benchmark dataset named Chars74K. As mentioned in the dataset description, the Chars74K dataset has both English and Kannada characters. In this research manuscript, only English characters are utilized for experimental analysis. The Chars74K dataset has 3,410 handwritten English images, 62,992 synthesized images, and 7,705 natural images. Additionally, the Chars74K dataset has 64 classes comprising upper case, lower case, and digits (A-Z, a-z, and 0-9). The results of dissimilar machine-learning classifiers with and without the utilization of FAICSA are stated in Table 1.

Table 1. Results of classifiers on the Chars74K dataset

With optimized feature vectors using FAICSA					
Classifiers	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**PPV (%)**	**MCC (%)**
CNN	97.98	96.82	96.23	97.68	98.26
DBN	93.56	92.66	92.34	93.23	91.12
Decision tree	92.57	90.89	89.14	90.88	91.44
KNN	93.45	94.14	92.62	90.95	91.19
MSVM	96.24	95.08	94.54	95.66	93.60
Sparse autoencoder	94.16	93.92	90.71	92.14	91.48
Stacked autoencoder	**99.64**	**98.94**	**97.64**	**98.39**	**99.65**
With original extracted feature vectors					
CNN	94.84	92.03	91.89	90.26	90.15
DBN	86.87	85.32	87.23	86.34	[illegible]
Decision tree	86.63	85.76	87.20	84.90	80.95
KNN	89.30	88.84	87.30	90.51	86.39
MSVM	92.38	90.45	91.84	89.66	[illegible]
Sparse autoencoder	91.46	90.58	92.43	[illegible]	89.21
Stacked autoencoder	**95.04**	**96.77**	**94.14**	**[illegible]**	**92.86**

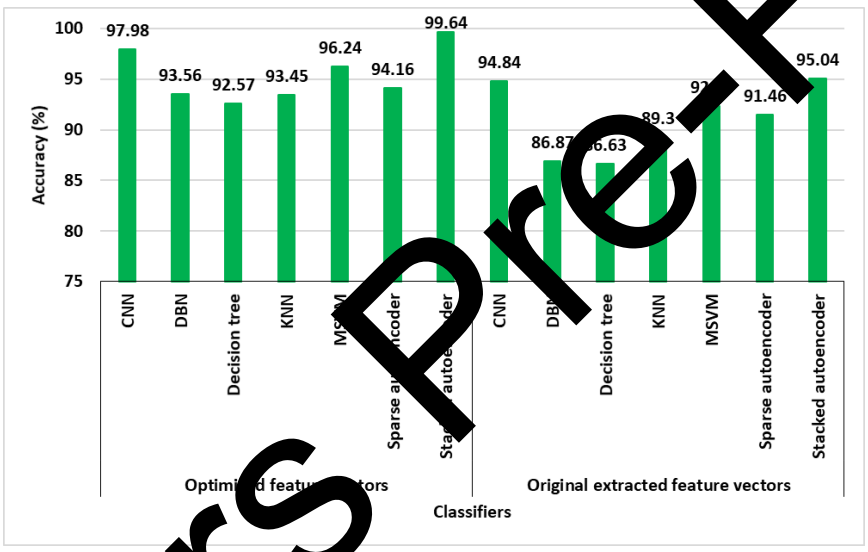


Figure 9. Comparative diagram of different classifiers' results on accuracy for the Chars74K dataset

By viewing Table 1, the stacked autoencoder model with FAICSA is seen to achieve maximum recognition results with MCC of 99.65%, PPV of 98.39%, specificity of 97.64%, the sensitivity of 98.94%, and recognition accuracy of 99.64% on the Chars74K dataset. The achieved results are far better than the other comparative classifiers like MSVM, K-Nearest Neighbor (KNN), decision tree, sparse autoencoder, DBN and conventional CNN. On the other hand, the results of individual classifiers without using optimization algorithms are detailed in Table 1. Related to the comparative classifiers, the stacked autoencoder efficiently tackles the unsupervised learning problems and eliminates the complexity within the Chars74K dataset. The comparative diagram of different classifiers' results on accuracy for the Chars74K dataset is represented in Figure 9.

In addition to this, the results of different optimization algorithms on the Chars74K dataset are specified in Table 2. In this research manuscript, the proposed FAICSA algorithm's efficacy is compared with the conventional optimization algorithms which are, Firefly Algorithm (FA), Salp Swarm Algorithm (SSA), Grey Wolf optimization (GWO) algorithm, WOA, CSA, HFPSO [20], FP-MPFO [21] and SI-FPA [22], while the achieved results are validated in light of MCC, sensitivity, PPV, accuracy, and specificity. These algorithms follow identical bio-inspired paradigms, that are common to the developed FAICSA, hence it is chosen for the evaluation. In this scenario, the selection of optimal feature vectors by the FAICSA efficiently diminishes the computational complexity to linear and computational time to 56.09 seconds, which is limited compared to other optimization algorithms. The comparative optimization algorithms: GWO, WOA, FA, SSA, CSA, HFPSO [20], FP-MPFO [21], SI-FPA [22] and FAPSO consume computational times of 76.02, 68.29, 60.22, 65.52, 73.66, 59.67, 60.96, 61.56 and 62.93 seconds. The graphical comparative diagram of different optimization algorithms' accuracy result on the Chars74K dataset is depicted in Figure 10. The parameters considered for analyzing the comparison are

determined as follows: population size is 30, maximum iteration is 100, alpha is one, beta is one, gamma is one, and theta is 0.97.

Table 2. Results of optimization algorithms on the Chars74K dataset

Optimization algorithms	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	MCC (%)
CSA	95.47	94.87	94.19	94.01	95.03
GWO	90.54	89.51	88.62	87.81	85.96
FA	92.57	90.75	92.00	93.62	88.44
FAPSO	96.06	95.18	95.25	95.11	96.38
FP-MPFO	96.27	96.28	95.39	95.12	96.89
HFPSO	94.83	95.23	92.19	94.13	95.23
SI-FPA	95.99	96.01	93.28	94.08	[illegible]
SSA	94.62	95.30	90.98	93.22	96.33
WOA	91.21	90.81	89.60	87.75	86.65
FAICSA	99.64	98.94	97.64	98.3[illegible]	[illegible]

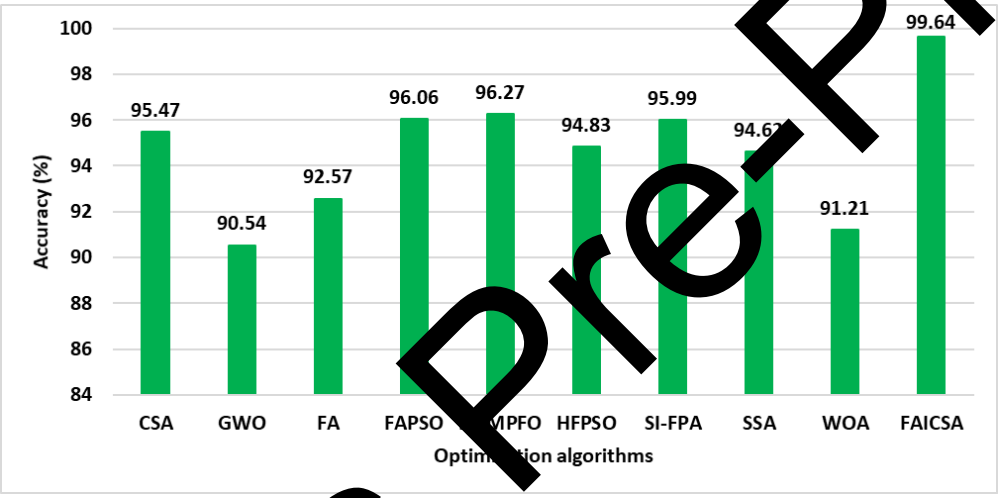


Figure 10. Comparative diagram of different optimization algorithm's accuracy result on the Chars74K dataset

4.2 Analysis by using the real-time dataset

In this scenario, the proposed model's efficacy is analysed on the real-time dataset, which includes 52 classes and 5,200-segmented English characters belonging to upper and lower cases. From the segmented images, 80:20% of images are respectively utilized for model training and testing. The results of classifiers on the real-time dataset are represented in Table 3. Here, the performance of the classifiers is validated with and without the utilization of the optimization algorithm. By viewing Table 3, it is clear that the stacked autoencoder model with FAICSA attains maximum recognition results than the existing machine-learning classifiers. The stacked autoencoder model with FAICSA obtains an MCC of 94.73%, PPV of 90.42%, specificity of 92.78%, a sensitivity of 93.47%, and a recognition accuracy of 92.06% on the real-time dataset. The comparative graph-based diagram of different classifiers' results on accuracy for the real-time dataset is specified in Figure 11.

The stacked autoencoder is one of the effective data compression models utilized for encoding the selected deep vectors into representation of smaller dimensions. Then, the decoder is utilized for reconstructing the input from the encoded format. The stacked autoencoder model performs dimensionality reduction and the vectors of the hidden layers are useful in improving the performance of character recognition.

Table 3. Results of classifiers on the real time dataset

With optimized feature vectors using FAICSA					
Classifiers	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**PPV (%)**	**MCC (%)**
CNN	91.91	92.01	91.33	89.24	92.95
DBN	87.98	88.12	87.56	86.38	87.02
Decision tree	88.32	87.49	85.11	89.98	86.62
KNN	87.06	85.07	84.11	90.49	87.28

MSVM	85.51	84.24	80.28	83.94	85.06
Sparse autoencoder	89.83	88.86	85.28	86.71	87.22
Stacked autoencoder	**92.06**	**93.47**	**92.78**	**90.42**	**94.73**
With original extracted feature vectors					
CNN	85.30	85.00	83.90	84.63	86.45
DBN	86.99	87.34	87.34	85.23	86.87
Decision tree	86.36	85.39	83.72	82.64	86.27
KNN	85.92	83.97	81.23	86.42	84.94
MSVM	82.76	80.26	79.01	83.21	81.42
Sparse autoencoder	84.49	85.74	82.82	83.84	85.39
Stacked autoencoder	**88.38**	**86.70**	**84.85**	**85.36**	**89.16**

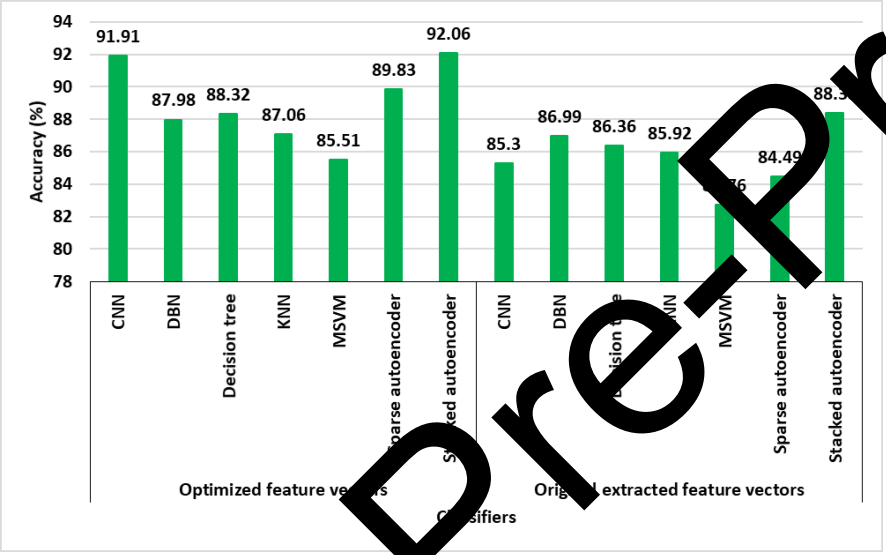


Figure 11. Comparative diagram of different classifier's results on accuracy for the real-time dataset

Table 4. Results of optimization algorithms on the real-time dataset

Optimization algorithms	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	MCC (%)
CSA	90.18	89.76	90.26	89.88	90.63
GWO	80.18	79.11	82.61	78.69	81.47
FA	90.03	90.22	91.58	88.79	89.15
FAPSO	91.24	91.00	90.43	89.87	91.27
FP-MPFO	91.32	92.09	90.87	90.01	91.87
HFPSO	90.89	90.12	89.09	89.12	88.65
SI-FPA	91.01	90.22	90.43	89.76	89.67
SSA	90.73	89.76	88.84	91.90	87.59
WOA	86.55	85.49	83.44	82.70	80.31
FAICSA	**92.06**	**93.47**	**92.78**	**90.42**	**94.73**

The results of dissimilar optimization algorithms on the real-time dataset are stated in Table 4. As discussed in the earlier sections, the selection of optimal feature vectors decreases the complexity and computational time of the model. In the real-time dataset, the FAICSA-stacked autoencoder model consumes minimal computational time of 58.77 seconds, whereby the comparative optimization algorithms: GWO, WOA, FA, SSA, CSA, HFPSO [20], FP-MPFO [21], SI-FPA [22] and FAPSO consume computational times of 78.14, 70.20, 62.28, 66.54, 77.60, 60.54, 62.63, 61.40 and 63.81 seconds, respectively. The proposed FAICSA combines the advantages of two optimization algorithms (FA and CSA), aiding in generating promising candidate solutions. The numerical analysis states that the proposed FAICSA has a faster convergence speed and higher local optimal avoidance ability than the comparative optimization algorithms. The comparative diagram of different optimization algorithm's accuracy result on the real-time dataset is mentioned in Figure 12.

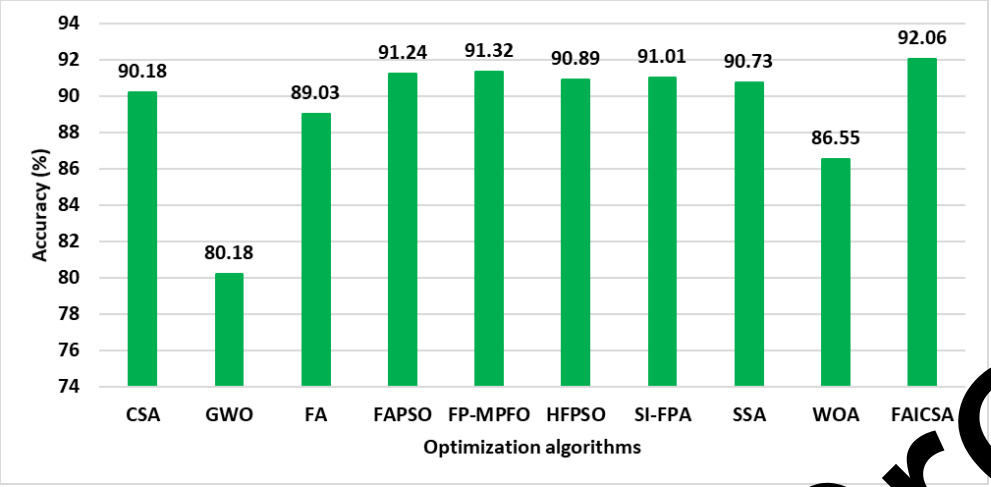


Figure 12. Comparative diagram of different optimization algorithm's accuracy result on the real-time dataset

4.3 Comparative analysis

Here, the FAICSA-stacked autoencoder model's performance is compared with a few existing models, and the results are specified in Table 5. Zanwar, S.R et al. [20] designed the NNP classification for performing the character recognition based on the features selected using HFPSO from HICA. The NNP used along with the HFPSO obtained the accuracy of 97.49%. Lee et al. [25] combined improved LBP descriptor with the shallow deep CNN model for an effectual character recognition. The improved LBP with shallow deep CNN model has 98.44% of accuracy on the benchmark Chars74K dataset. In addition to this, Gaptha et al. integrated enhanced LBP, EHO and LSTM for efficiently recognizing English characters. In this study, the EHO-LSTM model attains 96.66% of accuracy on the Chars74K dataset. Gangadhar and Jagde performed the dimension reduction using MSSAE and it used MSVM to recognize the handwritten characters. In this research, the MSSAE with MSVM obtained 99.18% of accuracy. Madakannu and Selvaraj implemented DIGI-Net for English character recognition. The experiments conducted on the Chars74K dataset state that the DIGI-Net attains 85% of accuracy. As stated in Table 5, the FAICSA-stacked autoencoder model has a recognition accuracy of 99.64%, which is higher than the existing models.

Table 5. Proposed and comparative models' results

Chars74K dataset	
Models	**Recognition accuracy (%)**
NNP-HFPSO [20]	97.49
Improved LBP with shallow deep CNN [25]	98.44
EHO-LSTM	96.66
MSSAE-MSVM	99.18
DIGI-Net	85
FAICSA-stacked autoencoder	**99.64**

4.4. Discussion

In this manuscript, the extraction and selection of optimal feature vectors by ResNet-18 and FAICSA from the images, significantly decreases the overfitting problems and computational time in the stacked autoencoder model. Additionally, the minimal semantic gap between the feature sets increases the recognition accuracy even under the conditions of noise, blur, non-uniform illumination, and complex backgrounds. The efficacy of the proposed model is clearly described in Tables 1, 2, 3, 4, and 5. The key factors that used to enhances the performances of FAICSA are detailed as follows: In FAICSA, the FA's exploration capacity is combined with an enhanced exploitation capacity of ICSA which used to avoid the premature convergence by expanding the searching and increasing local search in appropriate regions. An attractiveness-based movement is leveraged by FA for guiding the candidate solutions in the direction of better solutions during the mitigation of local optima issue via random perturbations. This helps to improve the FAICSA's robustness in a higher dimensional feature space. Therefore, the FAICSA effectively minimizes the dimensionality via discovering highly differentiated features and discarding the irrelevant ones for enhancing the recognition accuracy for both the Chars74K and real-time datasets. The recognition accuracy of Chars74K dataset is 99.64% that is high compared to the state of art classifiers, optimization approaches and comparative models. Specifically, the recognition accuracy of FAICSA-

stacked autoencoder is 99.64% than the existing approaches such as NNP-HFPSO [20], Improved LBP with shallow deep CNN [25], EHO-LSTM , MSSAE-MSVM and DIGI-Net. The incorporation of Cauchy random number and attractiveness-based movement in FAICSA is used to select an appropriate features which helps to enhance the OCR.

On the contrary, the real time dataset has certain issues due to changes in the lighting conditions, background noise and handwriting styles. An appropriate pre-processing techniques such as binarization, normalization, and morphological operations are used for enhancing image quality for a better segmentation and feature extraction. Even with these issues, the developed FAICSA-stacked autoencoder efficiently optimized the feature selection process, achieving a significant accuracy of 92.06%. However, the performance of FAICSA is sensitive to the hyper parameter values such as absorption coefficient, maximum iterations and population size. A fine tuning of aforementioned parameters for certain datasets is important in obtaining enhanced performances, that is not generalizable over various problems.

5. Conclusion

An automated model: FAICSA-stacked autoencoder is introduced for recognizing English characters. The primary aim of this manuscript is to develop a hybrid optimization algorithm for selecting optimum feature vectors. This process improves the performance of OCR with lower computational time and memory usage. Extensive experimental evaluations on the Chars74K and real-time datasets confirm that the proposed model superiorly improves the English character recognition performance. The obtained results of the proposed model are better than the results of other feature optimization algorithms namely, GWO, WOA, FAICSA, CSA, HFPSO, FP-MPFO, SI-FPA and FAPSO. It is observed that the FAICSA outperforms the other algorithms on the evaluation metrics of MCC, sensitivity, PPV, accuracy, and specificity. On the other hand, the stacked autoencoder's efficacy is contrasted against the traditional machine-learning classifiers like decision tree, KNN, MSVM, sparse autoencoder, DBN and conventional CNN. In this context, the stacked autoencoder accomplishes a higher recognition rate with limited complexity and computational time. The proposed FAICSA-stacked autoencoder model gains higher recognition accuracies of 99.64% and 92.06% on the Chars74K and real-time datasets, correspondingly. This study possesses certain limitations regarding confusion in cursive English characters, which potentially is accurately recognizable by understanding in context. In the future, this research work may be extended by automating the process of segmentation, feature extraction, and optimization by developing an effective unsupervised deep learning model. The developed FAICSA has provided improved performances in this research, hence the future research can be explored in the incorporation of newer optimization approaches such as FP-MPFO and SI-FPA.

Author Contributions:

Snehal Prabhakarrao Dongre: Visualization; Conceptualization; Formal Analysis; Resources; Project Administration; Investigation.

Dharmpal D. Doye: Methodology; Supervision; Data Curation; Manuscript - Review & Editing; Validation; Manuscript Original Draft.

All authors have read and approved the final manuscript

Declarations

Funding: This research received no external funding.

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethics Approval: I/We declare that the work submitted for publication is original, previously unpublished in English or any other language(s), and not under consideration for publication elsewhere.

Consent to participate / Informed Consent: Not Applicable.

Consent for publication: I certify that all the authors have approved the paper for release and are in agreement with its content.

Data Availability: The datasets generated during and/or analysed during the current study are available in the Chars74K dataset repository:

- Chars74K dataset link = http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

References

[1] N.D. Cilia, C.D. Stefano, F. Fontanella, A.S. di Freca, A ranking-based feature selection approach for handwritten character recognition, Pattern Recognit. Lett. 121 (2019) 77–86. https://doi.org/10.1016/j.patrec.2018.04.007

[2] A.K. Sampath, N.Gomathi, Handwritten optical character recognition by hybrid neural network training algorithm, The Imaging Science Journal. 67 (2019) 359–374. https://doi.org/10.1080/13682199.2019.1661591

[3] R. Ahmed, M. Gogate, A. Tahir, K. Dashtipour, B. Al-tamimi, A. Hawalah, M.A. El-Affendi, A. Hussain, Novel Deep Convolutional Neural Network-Based Contextual Recognition of Arabic Handwritten Scripts, Entropy. 23 (2021) 340. https://doi.org/10.3390/e23030340

[4] R.S. Alkhawaldeh, Arabic (Indian) digit handwritten recognition using recurrent transfer deep architecture, Soft Comput. 25 (2021) 3131–3141. https://doi.org/10.1007/s00500-020-05368-8

[5] S. Misra, R.H. Laskar, Integrated features and GMM based hand detector applied to character recognition system under practical conditions, Multimedia Tools Appl. 78 (2019) 34927–34961. https://doi.org/10.1007/s11042-019-08105-y

[6] P. Sahare, S.B. Dhok, Robust character segmentation and recognition schemes for multilingual Indian document images, IETE Tech. Rev. 36 (2019) 209–222. https://doi.org/10.1080/02564602.2018.1450649

[7] N.R. Soora, E.U.R. Mohammed, S.W. Mohammed, A Novel Geometrical Scale and Rotation Independent Feature Extraction Technique for Multi-lingual Character Recognition, International Journal of Advanced Computer Science and Applications. 11 (2020) 231–239. https://dx.doi.org/10.14569/IJACSA.2020.0111130.

[8] N. Modhej, A. Bastanfard, M. Teshnehlab, S. Raiesdana, Pattern separation network based on the hippocampus activity for handwritten recognition, IEEE Access. 8 (2020) 212803–212817. https://doi.org/10.1109/ACCESS.2020.3040298.

[9] M.H.H. Nashif, M.B.A. Miah, A. Hasib, A.C. Moulik, M.S. Islam, M. Zakareya, A. Ullah, M.A. Rahman, M. Al Hasan, Handwritten numeric and alphabetic character recognition and signature verification using neural network, J. Inf. Secur. Appl. 9 (2018) 209–224. https://doi.org/10.4236/jis.2018.93015

[10] S.P. Deore, A. Pravin, Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset, Sādhanā. 45 (2020) 243. https://doi.org/10.1007/s12046-020-01484–1

[11] L. Xu, Y. Wang, R. Li, X. Yang, X. Li, A hybrid character recognition approach using fuzzy logic and stroke Bayesian program learning with Naïve Bayes in industrial environments, IEEE Access. 8 (2020) 124767–124782. https://doi.org/10.1109/ACCESS.2020.3007487

[12] M. Jangid, S. Srivastava, Handwritten Devanagari Character Recognition Using Layer-Wise Training of Deep Convolutional Neural Networks and Adaptive Gradient Methods, Journal of Imaging. 4 (2018) 41. https://doi.org/10.3390/jimaging4020041

[13] S.R. Narang, M.K. Jindal, M. Kumar, Devanagari ancient character recognition using DCT features with adaptive boosting and bootstrap aggregating, Soft Comput. 23 (2019) 13603–13614. https://doi.org/10.1007/s00500-019-03897-5

[14] N.P. Sutramiani, N. Suciati, D. Siahaan, MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network, ICT Express. 7 (2021) 521–529. https://doi.org/10.1016/j.icte.2021.04.005

[15] S. Chandure, V. Inamdar, Handwritten MODI character recognition using transfer learning with discriminant feature analysis, IETE J. Res. 69 2584-2594 (2023). https://doi.org/10.1080/03772063.2021.1902867

[16] K. Nongmeikapam, W.K. Kumar, O. N. Meetei, T. Tuithung, Increasing the effectiveness of handwritten Manipuri Meetei-Mayek character recognition using multiple-HOG-feature descriptors, Sādhanā. 44 (2019) 104. https://doi.org/10.1007/s12046-019-1086-0

[17] S. Inunganbi, P. Choudhary, K.M. Singh, Local texture descriptors and projection histogram based handwritten Meitei Mayek character recognition, Multimedia Tools Appl. 79 (2020) 2813–2836. https://doi.org/10.1007/s11042-019-08482-4

[18] F. Naiemi, V. Ghods, H. Khalesi, An efficient character recognition method using enhanced HOG for spa image detection, Soft Comput. 23 (2019) 11759–11774. https://doi.org/10.1007/s00500-018-03428-z

[19] E.H. Houssein, D. Oliva, E. Celik, M.M. Emam, R.M. Ghoniem, Boosted sooty tern optimization algorith for global optimization and feature selection, Expert Syst. Appl. 213 (2023) 119015. https://doi.org/10.1016/j.eswa.2022.119015

[20] S.R. Zanwar, Y.H. Bhosale, D.L. Bhuyar, Z. Ahmed, U.B. Shinde, S. Narote, English handwritten character recognition based on ensembled machine learning, J. Inst. Eng. India Ser. B. 104 (2023) 1053-1067. http://dx.doi.org/10.1007/s40031-023-00917-9

[21] S.P.T. Parashivamurthy, S.V. Rajashekararadhya, HDLNet: design and development of hybrid deep learning network for optimally recognising the handwritten Kannada characters, Aust. J. Electr. Electron. Eng. 21 (2024) 268-288. https://doi.org/10.1080/1448837X.2024.2316493

[22] S.N. Devi, N.S. Fatima, 2024. Handwritten optical character recognition using TransRNN trained with self improved flower pollination algorithm (SI-FPA), Multimedia Tools Appl. https://doi.org/10.1007/s11042-024-19758-9.

[23] R. Ptucha, F.P. Such, S. Pillai, F. Brockler, V. Singh, P. Hutkowski, Intelligent character recognition using fully convolutional neural networks, Pattern recognit. 88 (2019) 604–613. https://doi.org/10.1016/j.patcog.2018.12.017

[24] R. Anand, T. Shanthi, R.S. Sabeenian, S. Veni, Real time noisy dataset implementation of optical character identification using CNN, International Journal of Intelligent Enterprise. 7 (2020) 67–80. https://doi.org/10.1504/IJIE.2020.104646

[25] S.-H. Lee, W.-F. Yu, C.-S. Yang, ILBP-DNet: based on improved local binary pattern shallow deep convolutional neural network for character recognition, IET Image Proc. 16 (2022) 669–680. https://doi.org/10.1049/ipr2.12376