SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I7P112

Volume 10 Issue 7, 125-139, July 2023

Original Article

Handwritten Character Recognition of Kannada Scripts using Novel Feature Extraction Techniques and BMCNN Classifier

Supreetha Patel Tiptur Parashivamurthy[1], Sannangi Viswaradhya Rajashekararadhya[2]

[1,2]*Department of Electronics and Communication Engineering, Kalpataru Institute of Technology, Tiptur, Tumkur District, Karnataka State, India.*

[1]*Corresponding Author : supreetha.patel@gmail.com*

Received: 07 May 2023 Revised: 02 July 2023 Accepted: 21 July 2023 Published: 31 July 2023

Abstract *- Handwritten Character Recognition (HCR) is one of the most popular research in recent years. Many HCR systems were developed based on various languages. However, only a few works are based on handwritten Kannada characters. Recognising handwritten Kannada characters is challenging because of the curvy and symmetric nature of the Kannada characters. Although various works were conducted for Kannada HCR, some issues must be solved. Hence, this work proposed BMCNN-based Kannada HCR. In the preprocessing phase, 3M filtering and the CLAHE techniques perform noise reduction and contrast enhancement. Then, the image is resized, angle rotated and mirror-inverted to obtain better accuracy of the input image. Then, the zonal, pattern and gradient features are extracted from the preprocessed image. Next, the significant features are selected by ISSA and then given to the BMCNN classifier to recognise the input Kannada character. To prove the efficiency of the proposed framework, the experimental analysis is conducted in terms of various measures and compared with state-of-art techniques. The results showed that the proposed recognition technique performs better than the existing techniques.*

Keywords *- Brownian Motion-based Convolutional Neural Network (BMCNN), Contrast Limited Adaptive Histogram Equalisation (CLAHE), Handwritten Character Recognition (HCR), Improved Sparrow Search Algorithm (ISSA), Mean Modified Median (3M) filter.*

1. Introduction

Around the globe, even though data generation and storage are majorly done in the electronic medium, there are still many data from the past that are in hand-written form [9, 10]. Hence, a solution is needed to recognise handwritten characters and digits. The character recognition system is the most emerging technology for digitising handwritten documents from any script [5]. Character recognition is achieved through segmentation, feature extraction, and classification using machine learning methods, making tremendous advancements [1]. Two different recognition categories of handwritten techniques exist offline and online recognition [7].

Online recognition automatically converts text written on a specialised digitiser based on pen tip movements. Offline recognition is based on converting text from an image to codes that can be processed by a computer application such as a text editor [3]. However, offline methods are more complex than online recognition methods. Many types of research have been successfully done for handwritten recognition of characters in different languages such as English, Arabic, and Indic languages. In Indic scripts, enormous work has been carried out in Tamil and Bengali scripts, whereas the work on handwritten Kannada numerals/characters recognition is in its infant stage [8].

The Kannada language, also called Kanarese and Kannana, is a member of the Dravidian language family and the official language of Karnataka in southern India. Kannada character set has 51 symbols, of which 16 are vowels and 35 are consonants. There is a possibility of 196000 combinations (35x35x16) [2]. Kannada scripts have a lot of confusing characters because of the complexity of the characters and the curve nature, so recognising those confusing characters is a tedious process [6]. Hence, developing a more dependable approach or more technically 'a system' for recognising handwritten characters of Kannada regional scripts still poses a challenge to researchers [11].

Various Machine Learning (ML) techniques such as Hidden Markov Model (HMM), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Deep Learning (DL) based methods like Convolutional Neural Network

(CNN), Deep Belief Network (DBN), etc., were utilised for handwritten character recognition. Despite several attempts based on ML architectural models and several DL architecture limitations, these systems remained unsolved [4]. Hence, an efficient method for the accurate detection of the Kannada language based on the novel Improved Sparrow Search Algorithm (ISSA) feature extractor and Brownian Motion-based Convolutional Neural Network (BMCNN) classifier is proposed in this paper.

1.1. Problem Identification

Although much research has been conducted on recognising Kannada characters in handwritten documents, several loopholes affecting existing systems' performance are still available. Some of the drawbacks of the existing methods are given below.

- Kannada character set is very vast, containing 17340 possible combinations. Designing a classifier is very complex if each character is considered a separate class.
- There are many characters in the Kannada script which are similar in structure. Hence, efficient feature extraction for this character is a tedious process. Only the image information like shape and structure does not give accurate results due to the structural complexity of the characters.
- Handwritten image contains noise and contrast problem. So, it is necessary to implement an efficient noise reduction technique.
- Angle variation is one of the significant problems in the Handwritten Character Recognition (HCR) system to reduce the recognition rate.
- Most of the Kannada Scripts are in curved shape. So, extracting suitable features by considering this problem is a significant challenge.

Analysing the abovementioned limitations, the proposed method aims to design a better model for recognising handwritten Kannada characters.

The structure of this paper is organised as follows; Section 2 analyses the various prior works related to handwritten character recognition. Section 3 discusses the proposed methodologies for recognising handwritten Kannada characters based on a novel feature extraction technique and BMCNN classifier. Section 4 analyses the performance of the proposed methodologies. Last, of all, section 5 completes the paper with a conclusion.

2. Related Works

Parashuram Bannigidad and Chandrashekar Gudada [12] suggested a historical Kannada handwritten character recognition technique based on the K-Nearest neighbour (KNN) approach. The presented approach used the box bonding segmentation method, and feature extraction was done based on the K-nearest neighbour. Experimental results showed that the historical Kannada handwritten characters of the Vijayanagar dynasty achieved the best recognition on the applied datasets. However, the classifier remained slower and costlier regarding time and memory.

Parashuram Bannigidad and Chandrashekar Gudada [13, 14] developed the Histogram of Oriented Gradient (HOG) feature-based age-type identification and recognition of historical handwritten Kannada documents. The accuracy of the suggested method was calculated based on two classifiers: K-NN classifier and the Support Vector Machine (SVM) classifier. The average accuracy classification for various dynasties based on the K-NN classifier was 92.3%, and the SVM classifier was 96.7%. However, the SVM classifier did not apply to large datasets.

Abhishek S. Rao et al. [15] presented a framework for handwritten Kannada character recognition based on the Deep Learning technique. The Convolutional Neural Network (CNN) based feature extraction was carried out that helped the model on the better classification of Kannada handwritten character recognition. The Char 74K dataset was used for experimental analysis, and found that the accuracy obtained was 95.11%. However, the computational speed of the presented method was low.

Nazal Modhej et al. [16] introduced a novel method for handwritten recognition based on a pattern separation network. Hence, an intelligent network was introduced based on the hippocampus's Dentate Gyrus (DG) activity. The DG was responsible for the recognition of highly overlapped patterns.

The Character Error Rate (CER) of 0.6% was obtained for the Modified National Institute of Standards and Technology (MNIST) dataset—also, the suggested method successfully recognised patterns with high noise. However, the method consumed more time as it used many datasets.

N. Shobha Rani et al. [17] suggested a classification technique for a deformed character set of printed and handwritten Kannada characters. The degraded patterns of character image samples were trained based on Deep CNN (DCNN), also known as Alex net. It was observed that the performance of the Alex net on the classification of printed character samples was reported as 91.3%, and regarding the handwritten text, an accuracy of 92% was recorded. However, the implementation of Alexnet struggled to learn features from image sets.

Based on a modified neural network, S. Kowsalya and P. S. Periasamy [18] presented a new framework for recognising handwritten Tamil characters. The input image was preprocessed, segmented, feature extracted, and given to the optimal artificial neural network for Tamil character recognition. The weights on the neural network were calculated by Elephant Herding Optimization (EHO). The suggested method attained high recognition rate than other methods when experimentally analysed. However, the EHO algorithm does not fit in the local minimum and has fewer control parameters, affecting the performance.

Raymond Ptucha et al. [19, 20] developed a novel offline handwriting recognition technique by implementing a Fully Convolutional Neural Network (FCN). The network measured and then resampled the input stream to a conical representation. The conical representation was processed by a convolutional network for precise character prediction. The presented method recognised common words and infinite symbol blocks such as surnames, phone numbers, and acronyms. However, FCN ignored oversized objects, which led to the loss of details on classification.

Nicole Dalia Cilia et al. [21] suggested a feature selection approach based on ranking for recognising handwritten characters. First, the method considered the univariate measures that produced the feature ranking. Next, a greedy search approach was employed for the feature subset selection that maximised the classification results.

The experimental analysis showed that the presented methodology obtained better classification results by selecting the reduced set of features. However, greedy algorithms failed to find the globally optimal solution because redundant features were not identified.

Mahesh Jangid and Sumit Srivastava [22] presented a framework for recognising unconstrained handwritten Devanagari characters. The method used DCNN that found the best features automatically and classified the features. The recognition accuracy was enhanced to 98% using Network Architecture (NA)-6 and the RMSProp adaptive gradient method. However, determining the number of CNN network layers and the number of neurons was the research challenge on DCNN.

Faisal Mushtaq et al. [23, 24] developed offline handwritten Urdu character recognition using a Deep Neural Network (DNN). The method used CNN architecture to recognise handwritten Urdu characters and numerals. A new handwritten dataset for the Nasta'liq Urdu script was introduced, named the Handwritten Urdu Character dataset (HUCD). The CNN-based method achieved a better recognition rate that outperformed the results of all state-of-the-art systems for the Urdu language. However, the method did not converge to an optimal solution, which affected the classification results.

K. Manjusha et al. [25] introduced a Malayalam handwritten character recognition approach based on integrating feature maps with CNNs. The approach utilised scattering transform-based wavelet filters as the first-layer convolutional filters in CNN architecture. The ScatCNN architecture experimented with the Malayalam handwritten character and verified that the performance was better than the baseline CNN technique. However, the method has limitations on salt-and-pepper noise corrupted images and uneven illumination images.

S. Karthik and K. Srikanta Murthy [26] suggested a framework for recognising handwritten Kannada characters based on the Deep Belief Network (DBN). The method employed DBN architecture with the Distributed Average of Gradients (DAG) feature for the recognition of isolated handwritten characters of Kannada.

The recognition accuracy obtained on the DBN-based method was 97.04%, which was better than the existing methods. The research suggested employing the presented method for Telugu character recognition, but a suspicion aroused whether the recognition rate would differ from the Kannada character recognition rate.

Yu Weng and Chunlei Xia [27] developed a new Deep Learning-Based Handwritten Character Recognition System for Mobile Computing Devices. The presented method applied CNN architecture on the Shui character dataset to effectively recognise handwritten characters. The application of Shui character recognition showed that CNN classified characters effectively and was more suitable for deployment on mobile devices. However, using resources such as memory, storage, power, computation, etc., was not reduced on mobile devices.

Saleh Albahli et al. [28] presented a framework for recognising handwritten characters based on an improved, faster Regional Convolutional Neural Network (RCNN) model. The improved Faster-RCNN was employed with DenseNet-41, which computed the deep features. Then, a regressor and classifier layer was used to localise and classify digits into ten classes. The performance was analysed on the standard MINST database and proved the efficiency of RCNN based model on accurate detection and classification of numerals than other methods. However, the algorithm required passing through a single image many times to extract all the objects.

Vishweshwrayya C. Hallur and R. S. Hegadi [29] suggested an approach for recognising the Kannada numerals based on DCNN. The attribute was excerpted from the segmented image using techniques like Run Length Count Directional Chain code, Discrete Wavelet Transform (DWT), and Curvelet Transform Wrapping. Then, excerpted attributes were passed to the DCNN model for classification. The results showed that DCNN performed well and yielded 96% accuracy. However, the DWT suffered from computational complexity and high time, which caused the method to process slowly.

3. Proposed BMCNN-Based Kannada HCR System

The Kannada language is written using the script which evolved from the 5th-century Kadamba script. Handwritten character recognition of Kannada characters was a complex process because of the shape similarity, non-uniformity of the characters, and various writing styles of different individuals in the characters. Hence, ISSA-based feature selection and BMCNN-based recognition techniques are proposed in this work. This work consists of five phases. The block representation of the proposed methodology is shown in Figure 1.

3.1. Input Data

Initially, the input data is collected. The data consists of handwritten Kannada Characters, which are denoted as follows,

$$H_c = \{k_1, k_2, \ldots\ldots, k_n\} \; or \; k_i, \; i = 1, 2, \ldots., n \qquad (1)$$

Here H_e is the whole dataset and k_n denotes the n number of handwritten Kannada characters.

3.2. Preprocessing

Initially, the input image k_i is preprocessed. Preprocessing is a technique to reduce computational complexity and attain better accuracy. Here, the preprocessing step consists of noise reduction, contrast enhancement, and image altering, which are explained below.

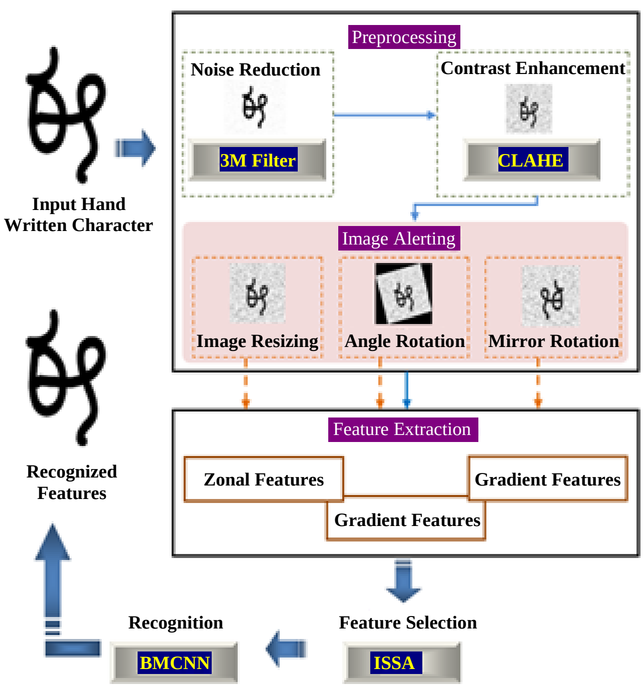


Fig. 1 Proposed handwritten Kannada character recognition framework

3.2.1. Noise Reduction

Initially, the noise in the input image is mitigated and sharpened using the Mean Modified Median (3M) filter. The median filter moves through the image pixel by pixel, replacing the center pixel value with the median value of neighbouring pixels. However, this process is less effective for the removal of noise. Hence, a modification is done by neglecting the central pixel and calculating the mean value of two central pixels in the ordered list of pixels.

$$k_i = \{p_1, p_2, \ldots\ldots, p_n\} \in W \qquad (2)$$

Where, k_i denotes the input image and $\{p_1, p_2, \ldots. p_n\}$ denotes the pixel values in the ordered list, W represents the $W_i x W_j$window.

$$M = \frac{1}{2}\left[C_i + C_j\right], C_i \in W_i, C_j \in W_j \qquad (3)$$

Here M is the mean of the central value of pixels, arranged in numerical order, in the windows of size w x w C_i and C_j denotes the central pixel value of the windows W_i W_j, respectively. The entire pixel value of W_i and W_j is replaced by the value of M. Hence, the obtained image is free of noise. The obtained image is represented as N_f.

3.2.2. Contrast Enhancement

In this session, the noise-reduced image N_f is enhanced by increasing the contrast of the image based on Contrast Limited Adaptive Histogram Equalization (CLAHE). The classical Adaptive Histogram Equalization (AHE) method applies enhancement in a particular region of an image according to the neighbouring pixels. However, a problem occurs due to the amplifying noise in the homogenous regions. Hence, the CLAHE method is developed, a popular block-based processing technique.

Initially, the image is divided into tiles of size $x \times y$. Next, calculate the histogram for each region based on the gray level. Then, calculate the Contrast Limited (CL) histogram of the region based on CL values as follows,

$$R_{avg} = \frac{(R_x \times R_y)}{R_G} \qquad (4)$$

Where, R_{avg} is the average number of pixels, $R_G$is the region containing several gray levels, R_x, R_y is the number of pixels in the dimensions x and y. Then, the actual contrast limit is calculated using the clip limit value as,

$$R_{cl} = R_c \times R_{avg} \qquad (5)$$

Where R_{cl} is the actual clip limit, R_c is the normalised CL within the range of [0, 1], the pixels are clipped if the number of pixels $p_x > R_{cl}$ $R_{\sum c}$ denotes the total clipped pixel. The average number of remaining pixels R_{avgP} is calculated as,

$$R_{avgP} = \frac{R_{\sum c}}{R_G} \tag{6}$$

The redistribution of pixels is performed until the remaining pixels are redistributed. The redistribution of pixels is formulated as,

$$\varsigma = \frac{R_G}{R_{remain}} \tag{7}$$

Here, ς is the Positive integer value and is greater than or equal to 1. R_{remain} is the number of clipped pixels that is remained. The intensity enhancement in each region is performed based on Rayleigh transform to make the image appear more natural. The density of each intensity value (D_i) based on Rayleigh forward is termed as,

$$D_i = D_{\min} + \sqrt{2\lambda^2\left[-\ln(-\psi(i))\right]} \tag{8}$$

Where, the lower bound of the pixel value is $D_{\min}$ and λ is Rayleigh scaling parameter, $\psi(i)$ is the transfer function. A higher λ value will improve the contrast enhancement in the image. The enhanced image thus obtained is denoted as E(i).

3.2.3. Image Altering

The given image is altered by resizing the image, rotating in different angles, and inverting the image by mirror rotation to get accurate classifier results.

Image Resizing

Each input image (E(i)) has different sizes. Hence, image resizing is necessary to increase and decrease the number of pixels. Here, the input image of different widths and lengths is resized to the standard dimensions of 128x128 and 512x512 without changing the actual data.

Angle Rotation of Images

Many characters present in the image (E(i)) may be slightly tilted. Because the characters are taken from handwritten data, the characters differ according to the writing environment and style. Hence, the image is rotated at four angles to effectively recognise characters during the testing phase.

Mirror Rotation

When the 3D handwritten image is converted to a 2D image, the image may be subject to mirror rotation. This problem will affect the recognition of handwritten characters. Hence, the proposed method rotates the enhanced image as mirror rotation.

The output from this segment consists of three (resized, angle rotated, mirror rotated) images, which are represented as,

$$O = \{o_1, o_2, o_3\} \; or \; o_q; \quad q = 1,2,3 \tag{9}$$

Where o_1 denotes the resized image, o_2 represents the angle rotated image, and o_3 denotes the mirror rotated image.

3.3. Feature Extraction

The resized, angle-rotated and mirror-rotated image o_q is given as input for the feature extraction process. The critical three feature classes are used in this proposed methodology. The three classes are zonal, pattern, and gradient features for the identification of the shape of the character.

3.3.1. Zonal Features

In character recognition, zonal feature extraction is done based on the topological information obtained from the pattern in the image. The given image o_q is divided into different zones, and from each zone, features like height and width are extracted using the formula:

$$Z_H = \frac{Row}{3}; \; D_H = \frac{Column}{3} \tag{10}$$

Where the zonal height is depicted as Z_H and D_H represents the zonal width. Then, the mean value of each zone is calculated to obtain zonal features as follows:

$$Z(i) = \sum_{H=1}^{N} \frac{Z_H + D_H}{2} \tag{11}$$

The output image thus obtained is termed as Z(i).

3.3.2. Pattern Features

Next, from the given image o_q, the patterns are extracted. The Local Binary Pattern (LBP) is employed for better performance of pattern extraction. The LBP computes the difference between the center pixel and the corresponding neighbouring pixels. The method of thresholding the difference between the center and neighbouring pixels is introduced to reduce the capturing of unnecessary things in the images. The threshold of a local binary pattern (LBP) is calculated as,

$$LBP_{K,R} = \sum_{p=1}^{S-1} 2^p \times T\left(V_p - V_c\right) \tag{12}$$

$$T(V_p - V_c) = \begin{cases} 1, & If\ (V_p - V_c) \geq 0 \\ 0, & Else \end{cases} \tag{13}$$

Where, K, R denotes the number of Surrounding neighbour pixels (K) within the radius (R) and V_p & V_c denotes the surrounding pixel values and center pixel values, respectively.

T() value denotes the threshold function. Hence, the pattern features are extracted, and the output image is denoted as φ_i .

3.3.3. Gradient Features

Here, the gradient features from the character image o_q are given as input of the gradient feature extractor, which is based on the Histogram of Orientated Gradients (HOG) feature descriptor.

Here, the gradients are obtained by combining the angle and magnitude obtained from the image. The angle of gradients on the x-axis $\left(X_g\right)$ and y-axis $\left(Y_g\right)$ is calculated as follows:

$$\theta = Sin^{-1}\left(\frac{\left(\frac{Y_g}{X_g}\right)}{\sqrt{1+\left(\frac{Y_g}{X_g}\right)^2}}\right) \tag{14}$$

Where θ denotes the gradient angle. The given equation finds the magnitude of the gradient:

$$|G| = \left(\left(X_g\right)^2 + \left(Y_g\right)^2\right)^{1/2} \tag{15}$$

Finally, the extracted feature is represented as:

$$F_e = \left\{f_1, f_2, \ldots\ldots, f_m\right\}\ or\ \ f_\eta,\ \ \eta = 1,2,3,\ldots,n \tag{16}$$

Where F_e denotes the feature set and f_m denotes the m number of extracted features.

3.4. *Feature Selection*

This sectionselects optimal features from the extracted features to mitigate the processing time for recognising handwritten characters. The Improved Sparrow Search Algorithm (ISSA) is implemented to select essential features in the proposed method.

The SSA algorithm is developed based on the foraging and anti-predation behaviour of the sparrows. There are two types of sparrows; the discoverers are responsible for foraging directions for the sparrow population, and the scrounger obtains food from the producers by joining the population or stealing.

The scouter sparrows identify the danger and alert the sparrow population to perform anti-predation behaviour. Hence, the most suitable place for the population is found after several iterations of the location of food search by the discoverers and scroungers. However, the SSA algorithm has problems in efficient recognition due to low convergence speed. Hence, the proposed work employed a logarithmic function in the updating process to solve this problem.

3.4.1. Initialisation

Initialise all parameters, such as population size (τ) (i.e., here the extracted features f_η are considered as the initial population in which each feature is considered as a sparrow), numbers of discoverers, scroungers, Scouters $\left(D_r, Y_r, N_s\right)$, and also the dimension of the search of food (ϑ), and maximum iterations $\left(It_{\max}\right)$. The initial location of the sparrow is shown as:

$$L_j = \left\{l_{j1}, l_{j2}, \ldots., l_{jd}\right\},\ \ j = \left(1,2,\ldots.,\tau\right) \tag{17}$$

Where L_j represents the position of the j^{th} sparrow and l_{jd} denotes the position of the j^{th} sparrow in dimension d . The fitness of each sparrow is calculated as $f\left\lfloor F_e\left(L_j\right)\right\rfloor$. Here, the fitness value considered is the accuracy of the proposed system. In SSA, the discoverers with better fitness get the food in the food search process.

3.4.2. Locate the Discoverers

Since the discoverers are responsible for leading the population in the food source location, the discoverers search for the food in a wide range of places than the Scroungers. In each iteration step, the best discoverer position is updated using the formula,

$$L_j^{I+1} = \begin{cases} L_j^I * e^{\left(-j/_{\delta . I_{\max}}\right)} & If \;\; A_v < S_T \\ L_j^I + N.M & If \;\; A_v \geq S_T \end{cases} \qquad (18)$$

Where I denotes the current iteration, δ is a random variable of range 0 to 1, Nis the random variable that follows a random distribution, A_v and S_T indicates the alarm value and safety threshold, respectively. M represents $1 \times d$ matrix in which values of all elements are 1. If $A_v < S_T$, the environment is free from enemies and the discoverer continues to search for food. If $A_v \geq S_T$ this means a scrounger is identified in the population and the sparrows should fly to safer areas.

3.4.3. Calculation of Scrounger's Location

When the sparrows are alarmed by the scrounger, the scouter population will conduct anti-predator behaviour. The position of the scrounger location is identified using an equation, which is expressed as:

$$L_j^{I+1} = \begin{cases} L_{best} + \mathrm{mod}\left(L_j^I - L_{best}\right) B^+ .M & If \;\; j \leq \dfrac{\tau}{2} \\ N.e^{\left(P_{worst} - L_j^I / j^2\right)} & If \;\; j > \dfrac{\tau}{2} \end{cases} \qquad (19)$$

Where L_{best} is the best location of the discoverers in the $I+1$ iteration, and the $1 \times d$ matrix $B^+ = B'\left(BB'\right)^{-1}$ with the values 1 or (–1), P_{worst} denotes the worst position of the sparrow population. If j>τ/2 then, the scrounger obtains the worst fitness value and does not get any food. j>τ/2 denotes the scrounger and is close to the best discoverer and gets food.

3.4.4. Location of Scouters

To alert danger to the population, some 10%-20% of sparrows are randomly chosen as Scouters. The location of the Scouters can be mathematically expressed as follows:

$$L_j^{I+1} = \begin{cases} \log\left(P_{best}\right) + \omega\left[\mathrm{mod}\left(L_j^I - \log\left(P_{best}\right)\right)\right] & for \;\; f\left(L_j^I\right) > f\left(P_{best}\right) \\ L_j^I + R\left(\dfrac{\mathrm{mod}\left(L_j^I - P_{worst}\right)}{f\left(L_j^I\right) - f\left(P_{worst}\right) + \chi}\right) & for \;\; f\left(L_j^I\right) = f\left(P_{best}\right) \end{cases} \qquad (20)$$

Where P_{best} defines the current best position of the population, X is a constant with a small value to avoid the zero division error, the f() function defines the fitness values, also R is a random number range from [-1,1], ω is the parameter to control step size and also shows the direction of the sparrow movement.

If $f\left(L_j^I\right) > f\left(P_{best}\right)$ the sparrow is near the boundary, it is vulnerable to attack if $f\left(L_j^I\right) = f\left(P_{best}\right)$ the individual sparrow is near to P_{best}, which is in the centre of the population, which approaches other sparrows to conduct anti-predator behaviour.

The new location of the sparrow is obtained using the above equations. If the new location is not better than the old location, return to the initialisation process and iterate through the above steps. Based on the best fitness value, the features are selected, and the selected features set (δ_f) is represented mathematically as:

$$\delta_f = \{f_1, f_2, f_3, \ldots, f_n\} \qquad (21)$$

3.5. Recognition Phase

The essential features obtained from the feature selection process (δ_f) are given as input to the Brownian motion-based Convolutional Neural Network (BMCNN) to recognise Kannada characters. The CNN consists of an input layer, convolution layers, pooling layers, dense layers, and output layers. In traditional CNNs, the weight values are assigned randomly, affecting the accuracy and taking more time to perform the recognition process. Hence, in this framework, the CNNs utilise Brownian Motion (BM) to select optimal weights in the convolution, pooling, and softmax layers.

The convolution layers extract the features from the input image, and the dense layer uses the output from the convolution layer to get the output. Initially, the input image of size 32x32x1 is convolved with a kernel of dimension 5x5 to give the convolved feature of size 28x28x6. Then, the convolved feature is pooled with a 2x2 kernel and reduced to 14x14x6.

The pooling layer 1 output is given to the convolution 2 and pooling 2 layers, which further reduces the image's dimensions. Next, the pooling layer 2 output is flattened and given to the fully connected layer. The final image is obtained at the softmax activation layer, and the input handwritten character will be recognised. The proposed BMCNN architecture model is given the Figure 2.

The input of the CNN model is the $\delta f = \{f_1, f_2, f_3,.., f_n\}$. The weight value for each input is optimised using Brownian motion. The BM is a method inspired by the free movement of air and liquid particles. BM is the random movement of particles in the liquid or gas when colliding with the fast-moving particles in the liquid or gas. Here, the molecule values are considered the weight value of neurons.

3.5.1. Convolution Layer

The convolution layer can capture all low-level features, such as border, colour, gradient, orientation, etc. The convoluted feature map C_f is represented as follows,

$$C_f = (\delta_f * W_g)(t) = \int_{-\infty}^{\infty} \delta_f(\phi) W_g(t-\phi)\, dt \qquad (22)$$

Where, * denotes the convolution operation and W_g represents the Kernel values, the 2D array of weights, t the time period, and the pixel position.

Then, the weight value W_g is optimised by the Brownian motion. The mathematical expression of BM-based weight updating is written as,

$$W_g = \frac{\beta t - \delta_f}{\sigma} \qquad (23)$$

Where, β and σ denotes the drift rate at the time t and the standard deviation for the selected pixel dimensions.

The pseudocode for the proposed ISSA is given as follows:

```
Input   :  Set of features F_e = {f_1, f_2,......,f_m}
Output  :  Selected optimal features
Begin
        Initialise the parameters D_r, Y_r, N_s, τ, ϑ and It_max
        Calculate fitness value
        Set iteration It = 1
        While (It ≤ It_max) do
                Update the position of discoverers using equation (18)
                Calculate the best and worst sparrow location
                Locate Scroungers by the combined usage of L_best, M, B^+ and L_j^i
                Update the position of Scouters
                If (f(L_j^I) > f(P_best)){
                        Located near the boundary and attacked
                } Else if (f(L_j^I) = f(P_best)){
                        Perform anti-predator behaviour
                }
                End if
                Update the current new location
                If (L_j^{I+1} > L_j){
                        Select fitness
                } Else {
                        Repeat
                }
                End If
                It = It + 1
        End While
        Return the best fitness
End
```

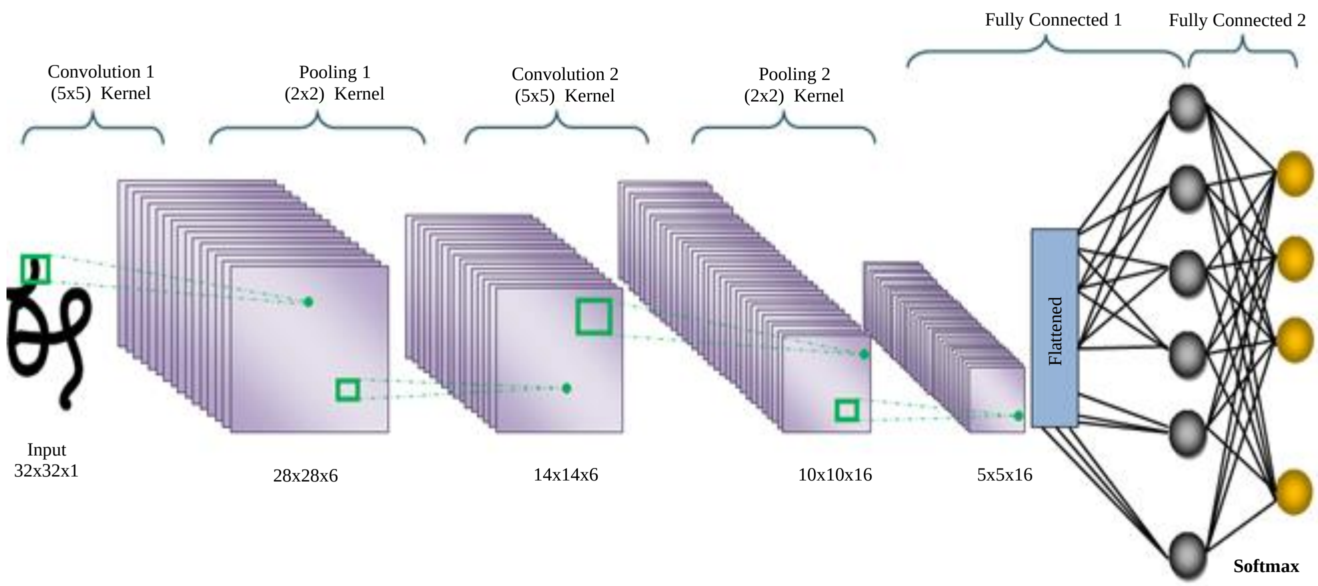


Fig. 2 Proposed BMCNN model

3.5.2. Pooling Layer

The pooling layer reduces the dimensional size of the input feature map. Hence, the computation in the network can be reduced. The output feature map obtained from the convolution unit is given as input for the pooling layer. The process of pooling layer is mathematically expressed as,

$$\rho_v = \frac{C_f - W_g}{\xi} + 1 \tag{24}$$

Where ρ_v denotes the output feature map volume ξ denotes the stride, which means the number of pixels shifts over the input feature matrix.

3.5.3. Softmax Layer

After the pooling layer, the image is flattened by the fully connected layer. The fully connected network uses softmax layers at the end to recognise the handwritten Kannada characters. The softmax layer converts the output value to corresponding probabilities $T(\rho_v)$ using the formula,

$$T(\rho_v)_a = \frac{e^{\rho_v}}{\sum_{v=1}^{R} e^{\rho_v}} \tag{25}$$

Where, $e^{\rho v}$ represents the non-normalised output of the pooling layer and R denotes the number of neurons in the output layer, $\sum_{v=1}^{R} e^{\rho_v}$ represents normalisation, which ensures that all output values of the softmax function will sum to 1, and also each output in the range between 0 and 1.

3.5.4. Loss Function

The loss function is calculated as a cross-entropy loss for multi-object recognition. The mathematical expression of the loss function $\gamma(T(\rho_v)_a, Q_a)$ is given as,

$$\gamma(T(\rho_v)_a, Q_a) = -\sum_{a=1}^{R} Q_a \log(T(\rho_v)_a) \tag{26}$$

Where, Q_a denotes the target output value of the classifier. If minimum loss function is obtained, the training is continued. If the maximum loss function is obtained, update the weight value using BMCNN. Finally, the handwritten Kannada character is recognised using the BMCNN classifier.

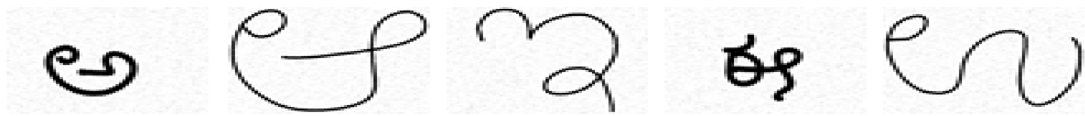
4. Results and Discussions

This section performs the experimental analysis of the proposed handwritten recognition of Kannada characters. Here, the handwritten Kannada character recognition is performed on the working platform of MATLAB. The input images from the Chars74K dataset are collected for performance analysis. Some of the samples are given in Figure 3.

4.1. Database Description

For training purposes, the Chars74K dataset is utilised. In the Chars74K dataset, the symbols used in both English and Kannada are available. The dataset contains 64 classes, 7705 characters from the images, 3410 handwritten characters, and 62992 characters from the computer fonts. Of which 657+ classes are Kannada characters, with 25 characters in each class. The dataset is collected from the segmented characters in the natural scene and hand-drawn characters.

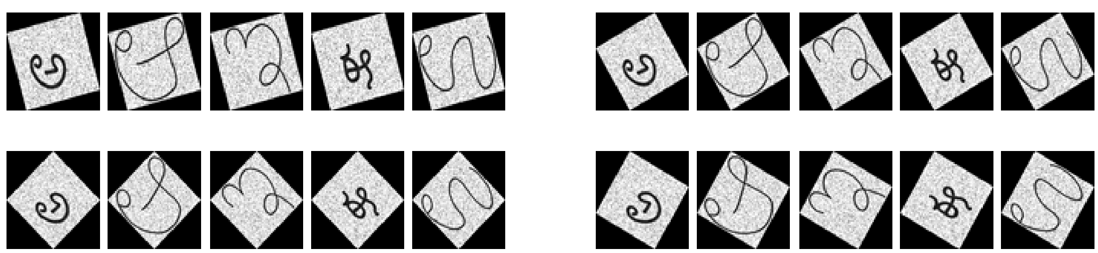
a) Input image



b) Noise-reduced image

c) Contrast-enhanced image

d) Resized image



e) Angle rotated image

f) Mirror rotated image

Fig. 3 Preprocessed output of handwritten Kannada character image obtained from chars74k database

4.2. Performance Analysis

Here, the proposed recognition method of handwritten Kannada characters is performed in three sections: filtering technique, feature selection, and recognition phase.

4.2.1. Performance Analysis of Filtering Technique

Here, the performance analysis of the proposed Mean Modified Median (3M) filter is analysed and compared with the prevailing median filter, Weiner and Gaussian filters in terms of Peak Signal to Noise Ratio (PSNR), Structural Index Similarity Method (SSIM), and Mean Square Error (MSE) values. Table 1 shows the experimental analysis of the proposed 3M filter conducted and compared with the existing Weiner, Gaussian, and Median noise-reducing techniques. The estimation of the amount of error in a system by averaging the squared difference between the observed and predicted values is termed MSE. Hence, the MSE value should be lower for a good technique. Here, the MSE value of Median filtering (0.0000999) is lower than all other existing algorithms. However, the MSE value of the proposed 3M filtering value (0.0000388) is lower than the median filtering. Hence, it is proved that the proposed filter performs better in reducing the noise in the input image than other prevailing techniques.

Table 1. A comparative analysis of MSE

Filtering Techniques	MSE
Proposed 3M	0.0000388
Median	0.0000999
Weiner	0.001092
Gaussian	0.061373

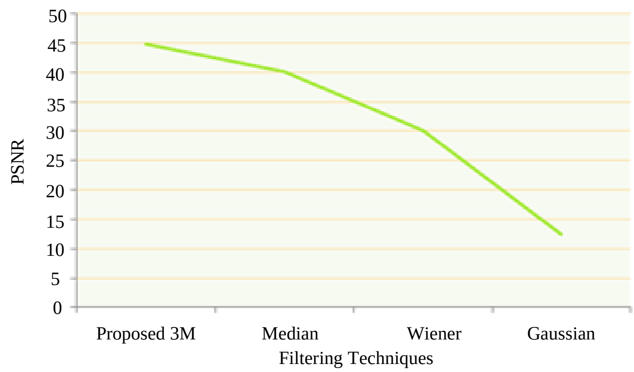


Fig. 4 PSNR analysis

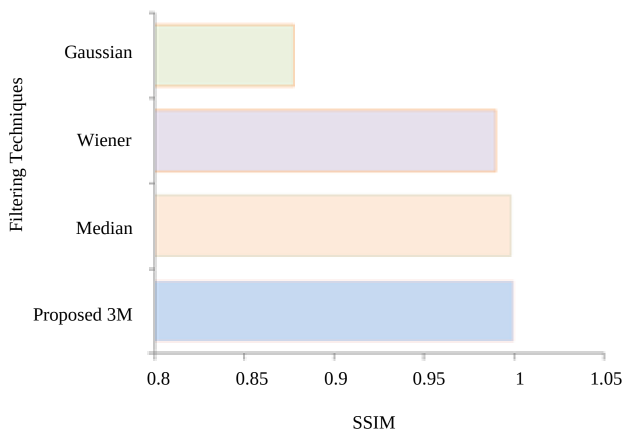


Fig. 5 SSIM analysis for the proposed 3M filtering

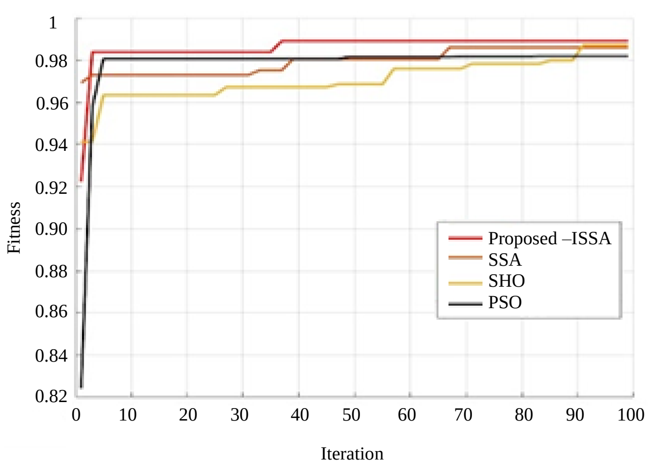


Fig. 6 Fitness function analysis

The PSNR value is analysed to compare the filtered image quality. So, an efficient system should possess a higher PSNR value. Figure 4 shows the comparative analysis between the proposed 3M filtering and the prior median, Weiner, and Gaussian filtering based on PSNR value. Here, the Gaussian filtering shows the worst PSNR value (12.51053). Compared to existing techniques, the proposed 3M filtering has a better PSNR value (44.8112). Hence, the image obtained using the proposed filtering has better quality than the prevailing techniques.

The comparative analysis of the proposed 3M filtering and the prevailing filtering techniques based on the SSIM parameter is shown in Figure 5. SSIM is a metric to calculate the perceived image quality. A system with an SSIM value closer to 1 means the quality of the image is better.

Here the proposed 3M filtering has the SSIM value of 0.99926, which is higher than the prevailing Median (0.00790), Weiner (0.990122), and Gaussian (0.877399) filtering techniques. Hence the quality of the obtained image using the proposed 3M filtering is better than the other prior techniques.

4.2.2. Performance Analysis of Feature Selection

In this section, the performance of the proposed Improved Sparrow Search Algorithm (ISSA) is analysed in comparison with the existing Sparrow Search Algorithm (SSA), Spotted Hyena Optimisation (SHO), and Particle Swarm Optimisation (PSO) algorithm. The fitness vs iteration is carried out for the performance analysis of the proposed ISSA methodology.

The fitness vs iteration analysis for the proposed ISSA methodology is shown in Figure 6. From the graph, it is observed that the fitness values increase when the number of iterations increases. The fitness vs iteration analysis identifies whether the system selects the critical features essential for the classification. Here, the fitness values of the proposed ISSA technique are higher when compared with the existing SSA, SHO, and PSO algorithms.

The fitness of the proposed ISSA initially starts below 0.93 and reaches above 0.98 within ten iterations and further increases between 30 and 40 iterations, which is relatively higher than the prevailing algorithms. Hence, it is verified that the proposed ISSA algorithm selects only the essential features needed for recognition.

4.2.3. Performance Analysis of the Recognition Phase

Here, the experimental analysis of the proposed Brownian Motion-based Convolutional Neural Network (BMCNN) algorithm is performed and is compared to the prevailing Convolutional Neural Network (CNN), Deep Neural Network (DNN), and Deep CNN (DCNN) algorithms.

The performance analysis is based on various metrics, such as accuracy, precision, recall, F-Measure, False Positive rate (FPR), False Negative rate (FNR), False Rejection Rate (FRR), Negative Predictive Value (NPV), and Mathews Correlation Coefficient (MCC).

Table 2. Experimental analysis of the proposed BMCNN with the existing CNN, DNN, and DCNN algorithms

Performance Metrics	Proposed BMCNN	CNN	DNN	DCNN
Accuracy	0.97995	0.97988	0.97984	0.97970
Precision	0.94169	0.86005	0.80320	0.64139
Recall	0.94169	0.86005	0.80320	0.64139
F-Measure	0.94169	0.86005	0.80320	0.64139

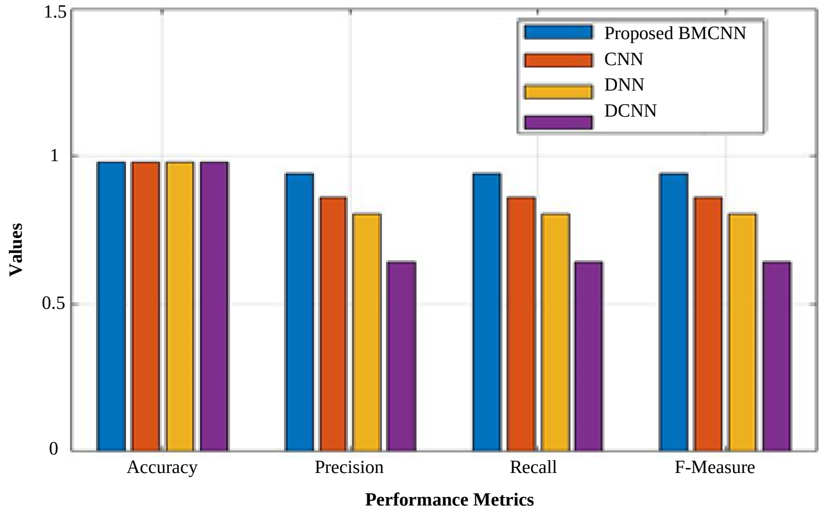


Fig. 7 Accuracy, precision, recall and f-measure analysis for the proposed BMCNN

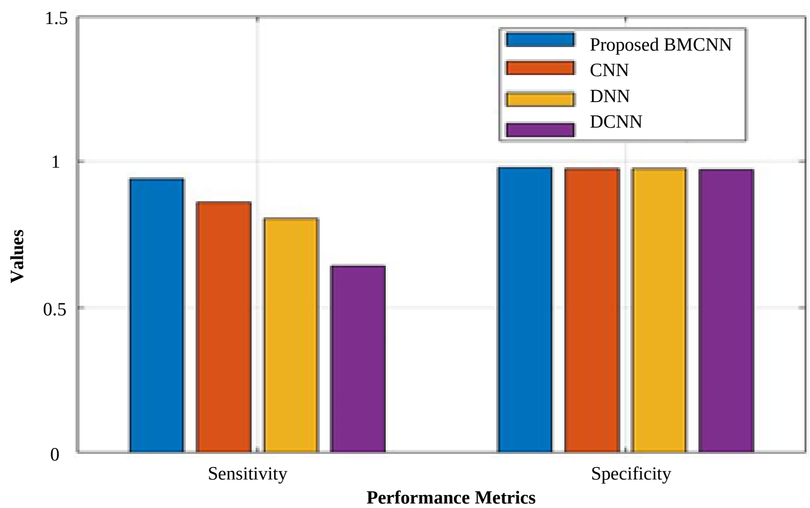


Fig. 8 Sensitivity and specificity analysis

The comparative analysis between the proposed BMCNN algorithm and the existing algorithms based on performance metrics such as accuracy, precision, recall, and specificity can be understood in Table 2. From the table, it is proven that the existing DCNN algorithm shows poor performance than all other algorithms. The accuracy defines how many characters the algorithm recognised correctly. Here, the accuracy of the proposed algorithm is 97.99%, which is higher than the existing CNN (97.98%) and DCNN (97.97%) algorithms. The F-Measure is calculated using the precision and recall values. Here, the precision and recall values of the proposed method are higher than the existing CNN, DCNN, and DNN algorithms. From this, it can be concluded that the performance of the proposed BMCNN algorithm based on the F-Measure is also better than the prevailing algorithms. The pictorial representation of accuracy, precision, recall, and F-Measure is shown in Figure 7.

Table 3. Comparative analysis between the proposed and the prevailing methods

Performance Metrics	Proposed BMCNN	CNN	DNN	DCNN
FPR	0.97956	0.97953	0.97951	0.97944
FNR	0.058309	0.13994	0.19679	0.35860
FRR	0.058309	0.13994	0.19679	0.35860
FDR	0.058309	0.13994	0.19679	0.35860
MCC	0.94047	0.85714	0.79910	0.63392

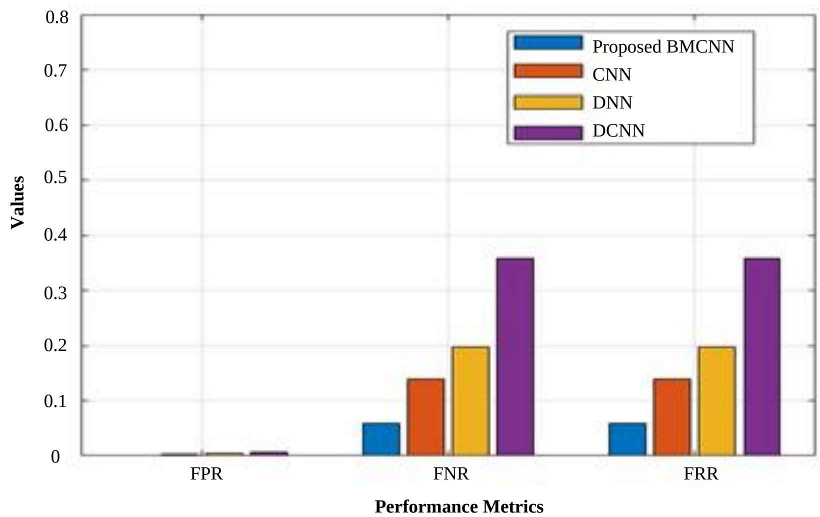


Fig. 9 Experimental analysis of the proposed BMCNN algorithm based on FPR, FNR, and FRR metrics

Figure 8 shows the performance analysis of the proposed and existing algorithms based on sensitivity and specificity. Sensitivity is the proportion of true positive rates predicted and specificity is the prediction of the true negative rate predicted by the algorithm. Here, the specificity of the proposed algorithm is higher (0.97878), followed by existing CNN (0.97708), DNN (0.9759), and DCNN (0.97252) algorithms. Hence, the proposed BMCNN methodology performs better in the prediction process than other algorithms.

Table 3 describes the performance analysis of the proposed BMCNN algorithm compared to the existing CNN, DNN, and DCNN algorithms in terms of FPR, FNR, FRR, FDR, and MCC. Here, CNN can be concluded as a better method among the existing algorithms. However, the proposed BMCNN algorithm performs better than the existing CNN algorithm. For instance, the FRR value of the proposed BMCNN algorithm is 0.058309, 58% lower than that of the existing CNN.

The FRR is the probability of the system failing to detect the match between the input image and the image in the database. Hence, a lower FRR value means the proposed model is better than other methodologies. Also, the FPR value of the proposed methodology (0.97956) is higher than the existing CNN (0.97956), DNN (0.97951), and DCNN (0.9744) algorithms, which means the character is identified correctly. The pictorial representation of the performance analysis based on the values of Table 3 is shown in Figure 9. Hence, from the above analysis, it can be concluded that compared with the existing CNN, DNN, and DCNN algorithms, the proposed BMCNN algorithm gives better results during handwritten Kannada character recognition.

5. Conclusion

This work proposed a novel method for feature extraction technique and BMCNN classifier for recognising handwritten Kannada characters. In preprocessing, the 3M filtering method removes noise from the input image. The ISSA methodology selects the critical features, and the BMCNN classifier is employed in the final phase to recognise the input characters accurately. The data training is based on the handwritten characters obtained from the chars74K dataset.

The performance of the proposed algorithm is analysed and compared with existing algorithms based on various metrics. The results proved that the proposed 3M filtering performs better in noise reduction than the prior techniques by attaining a low MSE value (0.0000388) than other prevailing algorithms. Next, the analysis of feature selection using ISSA also proved that the proposed model selected the essential features. Finally, the BMCNN algorithm in the recognition phase is analysed based on accuracy, precision, recall, F-Measure, sensitivity, specificity, FRR, FNR, FPR, MCC, and FDR metrics the proposed technique performed better in the recognition phase than other existing methodologies. For instance, the precision obtained by the proposed method is 0.94169, which is 9.49% higher than the CNN technique and 46.82% higher than the DCNN algorithm. Hence, all the results proved that the proposed model is better for Kannada HCR than the prevailing models. The proposed framework can be enhanced by applying advanced algorithms for recognising handwritten Kannada characters and characters available in other languages.

References

[1] Dibyakanti Mahapatra, Chandrajit Choudhury, and Ram Kumar Karsh, "Generator Based Methods for Off-Line Handwritten Character Recognition," *Advanced Communication Technologies and Signal Processing (ACTS)*, pp. 01-06, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Kusumika Krori Dutta et al., "Kannada Alphabets Recognition using Decision Tree and Random Forest Models," *3rd International Conference on Intelligent Sustainable Systems*, pp. 534-541, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Kusumika Krori Dutta et al., "Kannada Character Recognition using Multi-Class SVM Method," *11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 405-409, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Bindu Madhuri Cheekati, and Roje Spandana Rajeti, "Telugu Handwritten Character Recognition using Deep Residual Learning," *4th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 788-796, 2020. [CrossRef] [Publisher Link]

[5] Kevin George Joe, Meghna Savit, and K. Chandrasekaran, "Offline Character Recognition on Segmented Handwritten Kannada Characters," *Global Conference for Advancement in Technology (GCAT)*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] G. S. Veena, T. N. R. Kumar, and A. Sushma, "Handwritten Off-Line Kannada Character/Word Recognition using Hidden Markov Model," *Proceedings of International Conference on Cognition and Recognition*, pp. 357-369, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] K. Asha, and H. K. Krishnappa, "Kannada Handwritten Document Recognition using Convolutional Neural Network," *3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions*, pp. 299-301, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Shobha Rani N et al., "Kannada Confusing Character Recognition and Classification using Random Forest and SVM," *3rd International Conference on Signal Processing and Communication (ICPSC)*, pp. 537-541, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Kishorjit Nongmeikapam et al., "Increasing the Effectiveness of Handwritten Manipuri Meetei-Mayek Character Recognition using Multiple-HOG-Feature Descriptors," *Sādhanā*, vol. 44, no. 5, pp. 1-14, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] LiQiang ZhangJuan, "An Improved Plate Recognition Method Based on Template Matching," *International Journal of Recent Engineering Science*, vol. 5, no. 3, pp. 12-16, 2018. [CrossRef] [Publisher Link]

[11] Shobha Rani N et al., "Deep Learning Network Architecture Based Kannada Handwritten Character Recognition," *2nd International Conference on Inventive Research in Computing Applications*, pp. 213-220, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Parashuram Bannigidad, and Chandrashekar Gudada, "Historical Kannada Handwritten Character Recognition using K-Nearest Neighbour Technique," *Journal of Artificial Intelligence Research & Advances*, vol. 6, no. 1, pp. 1-7, 2019. [Publisher Link]

[13] Parashuram Bannigidad, and Chandrashekar Gudada, "Age-Type Identification and Recognition of Historical Kannada Handwritten Document Images using HOG Feature Descriptors," *Computing, Communication and Signal Processing*, vol. 810, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ivan Christian, and Gede Putra Kusuma, "Improving OCR Performance on Low-Quality Image using Pre-Processing and Post-Processing Methods," *International Journal of Engineering Trends and Technology*, vol. 71, no. 6, pp. 396-405, 2023. [CrossRef] [Publisher Link]

[15] Abhishek S Rao, "Exploring Deep Learning Techniques for Kannada Handwritten Character Recognition: A Boon for Digitization," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 11078-11093, 2020. [Google Scholar] [Publisher Link]

[16] Nazal Modhej, "Pattern Separation Network Based on the Hippocampus Activity for Handwritten Recognition," *IEEE Access*, vol. 8, pp. 212803-212817, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shobha Rani N et al., "Deformed Character Recognition using Convolutional Neural Networks," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 1599-1604, 2018. [Google Scholar]

[18] Kowsalya S, and Periasamy P S, "Recognition of Tamil Handwritten Character using Modified Neural Network with Aid of Elephant Herding Optimization," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 25043-25061, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Raymond Ptucha et al., "Intelligent Character Recognition using Fully Convolutional Neural Networks," *Pattern Recognition*, vol. 88, pp. 604-613, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Sagar Badgujar et al., "Real Time Number Plate Recognition and Tracking Vehicle System," *SSRG International Journal of Computer Science and Engineering*, vol. 2, no. 12, pp. 5-8, 2015. [CrossRef] [Publisher Link]

[21] Nicole Dalia Cilia et al., "A Ranking-Based Feature Selection Approach for Handwritten Character Recognition," *Pattern Recognition Letters*, vol. 121, pp. 77-86, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mahesh Jangid, and Sumit Srivastava, "Handwritten Devanagari Character Recognition using Layer-Wise Training of Deep Convolutional Neural Networks and Adaptive Gradient Methods," *Journal of Imaging*, vol. 4, no. 2, pp. 1-14, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Faisel Mushtaq et al., "UrduDeepNet: Offline Handwritten Urdu Character Recognition using Deep Neural Network," *Neural Computing and Applications*, vol. 33, no. 22, pp. 15229-15252, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Saikiran Subbagari, "Leveraging Optical Character Recognition Technology for Enhanced Anti-Money Laundering (AML) Compliance," *SSRG International Journal of Computer Science and Engineering*, vol. 10, no. 5, pp. 1-7, 2023. [CrossRef] [Publisher Link]

[25] Manjusha K, Anand Kumar M, and Soman K P, "Integrating Scattering Feature Maps with Convolutional Neural Networks for Malayalam Handwritten Character Recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 187-198, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] Karthik S, and Srikanta Murthy K, "Deep Belief Network Based Approach to Recognize Handwritten Kannada Characters using Distributed Average of Gradients," *Cluster Computing*, vol. 22, no. 7, pp. 1-9, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[27] Yu Weng, and Chunlei Xia, "A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices," *Mobile Networks and Applications*, vol. 25, no. 7, pp. 402-411, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Saleh Albahli et al., "An Improved Faster-RCNN Model for Handwritten Character Recognition," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8509-8523, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] Vishweshwrayya C Hallur, and Hegadi R S, "Handwritten Kannada Numerals Recognition using Deep Learning Convolution Neural Network (DCNN) Classifier," *CSI Transactions on ICT*, vol. 8, no. 6, pp. 1-15, 2020. [CrossRef] [Google Scholar] [Publisher Link]