University of Ibadan
Journal of Science and Logics in ICT Research

Back Propagation Neural Network and Chicken Swarm Optimization for Yoruba Indigenous Food Recognition

Modupeola Toyin Agbeniyi, Jumoke Falila Ajao Ronke S Babatunde
agbeniyitoyin@gmail.com, jumoke.ajao@kwasu.edu.ng, ronke.babatunde@kwasu.edu.ng

Department of Computer Science, Kwara State University, Malete, Nigeria

Correspondence email: agbeniyitoyin@gmail.com
Contact: 08139301607

Abstract

Back Propagation Neural Networks (BPNNs) are widely used to model complex systems in a steady state. However, BPNNs have a slow convergence rate. Several meta-heuristic algorithms have been used to speed up its convergence. Though, the modifications increased the number of parameters which affect the convergence rate, but more work needs to be done on BPNN in order to improve the performance of BPNN. This work introduced Chicken Swarm Optimization (CSO) technique to improve the performance of BPNNs in order to recognize Yoruba indigenous food. 1251 varieties of Yoruba indigenous foods such as Amala ogede, Iyan gbere, Puguru, Akara, Dele, Ekuru, Monu, Aseke, Abari, Sapala, and Egbo were prepared. The prepared foods were captured using a digital camera and were digitalized. The digitalized foods were subjected to preprocessing using Image resizing, Morphological, Edge scaling, Histogram, normalization and Sobel edge. All images were subjected to feature extraction using Local Binary Pattern (LBP) and the feature vectors gotten were optimized with Chicken Swarm Optimization (CSO) algorithm. The result of the optimized parameters were classified using Back Propagation Neural Network (BPNN). The recognition accuracy using BPNN yields 82.22 %, 87.59 %, 83.67 %, and 85.34 % for the four categories of food respectively, whereas BPNN-CSO yields 85.34 %, 91.90 %, 91.02 %, and 90.23 %, respectively. Sensitivity using BPNN yields 87.96%, 92.22%, 90.22%, 90.13% while BPNN-CSO delivers 90.13%, 97.78%, 97.74%, 96.71% respectively using recognition time and sensitivity standard term that are included in the result table. It was observed that, the recognition accuracy, sensitivity, specificity, and false positive ratio values of BPNN-CSO gives improved performance result compared to BPNN.

***Keywords**: Back Propagation Neural Network, Chicken Swarm Optimization, Computer Vision, Food Recognition, Local Binary Pattern*.

1. Introduction

Back Propagation Neural Networks (BPNN) are a powerful algorithm used in tackling complicated computing issues over the years. Despite the amazing success, it was noticed that BPNN takes longer training time to converge [1]. In order to circumvent this difficulty, numerous optimization techniques have been attempted to increase the performance of the BPNN, such as Particle Swarm Optimization [2]. To determine the correct values for these parameters is a complex issue for the best values depend on the data which are used for the training of the BPNN. In order to solve these problems, the research work used Chicken Swarm Optimization approach on Local binary pattern to find the ideal set of parameters for BPNN in order to reduce the training time of the BPNN. Over the years, BPNN has been applied areas of machine learning problems to solve face recognition problem, character recognition, word recognition, signature recognition and speech recognition. This work uses the BPNN to recognize Yoruba indigenous food by saving and preserving the cultural heritage and values of the Yoruba Nation from extinction.

The food recognition system is a set of machine learning and computer vision techniques that have demonstrated the development of systems that automatically recognize different foods.

Modupeola Toyin Agbeniyi, Jumoke Falila Ajao and Ronke S Babatunde (2023). Back Propagation Neural Network and Chicken Swarm Optimization for Yoruba Indigenous Food Recognition, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 9 No. 1, pp. 16 -

These systems are used at restaurants, grocery stores, canteens, cafeterias, hospitals and other facilities [2]. The task of food recognition is a tedious task in computer vision because of the various shapes, colours, textures and sizes. Hence, it means that the recognition algorithm's performance must be improved upon, in order to attain better recognition accuracy. The research employed Chicken Swarm Optimization with Local Binary Pattern approaches to identify the food parameters that reduced the convergence rate of BPNN used to classify the food based on the four food categories.

2. Related Works

Anthimopoulos *et al.* [3] developed a system that performed automatic food recognition, based on Bag-of-Features (BoF) model. A series of five major experiments were carried out for choosing and optimizing parameters of the system. The system achieved classification accuracy of 78%. Dehais *et al.* [4] detected and segmented the food already detected dishes in an image by combining region growing with a deep CNN-based food border detection, average accuracies of 88% and 92% were achieved. Kagaya, Aizawa and Ogawa [5] constructed a dataset of the most frequent food items in a publicly available food-logging system, the application of CNN to the tasks of food detection and recognition through parameter optimization was carried out which achieved 93.8% accuracy. Tahir & Loo [6] developed a survey of image-based food recognition and volume estimation techniques for diet.

According to the findings, approximately 66.7% of the surveys studied the use of visual features from deep neutral networks. It was observed that most of the researchers focused on intercontinental food recognition, food-logging, food segmentation, and automated monitoring on the dietary, but this research tends to consider Yoruba indigenous food recognition by saving the aforementioned food from going into total extinction. The composition of the food items and the process of the food preparation were presented in digital image form in order to make it available and accessible to everyone in the future. The proposed work optimized back propagation neural network with chicken swarm optimization to reduce the slow convergence rate problem associated with BPNN.

3. Methodology

In this research, recognition of Yoruba indigenous food was performed on several food images. The input to the system is the image of Yoruba indigenous foods, which were acquired through a digital camera after it has been prepared and served. These images were pre-processed using preprocessing techniques such as conversion to grayscale, morphological filter to denoise image and histogram equalization to enhance the image quality. The Sobel edge detection technique was used for image segmentation. The segmented image served as input to local binary pattern extraction technique. The result obtained was optimized using Chicken Swarm Optimization technique and the classification was performed using back propagation neural network. The image below describes the process flow of the food recognition system as shown in Figure 1, Figure 2 Flowchart of BPNN-CSO Training and Testing Model.

This system involves data acquisition, preprocessing, feature extraction by LBP, parameter optimization using CSO and classification BPNN for both training and recognition stages, once the training and recognition stage is completed the system then the classification and lists of the recipe for the various food categories will be displayed.

3.1 Data Collection

The research datasets were built from different categories of manually prepared foods by elders from different villages in the Yoruba territory. Approximately 1,251 images of ready foods were obtained. The datasets were distributed in four food categories. This has been done based on the main recipe, i.e., (i) Agbado (corn varieties), (ii) Ewa (bean varieties), (iii) Isu (yam varieties) and (iv) Okele (Morsel). Data sets were categorized to enhance the performance of the recognition system. Samples of data collected for this research are shown in Figures 3, 4, 5, and 6.

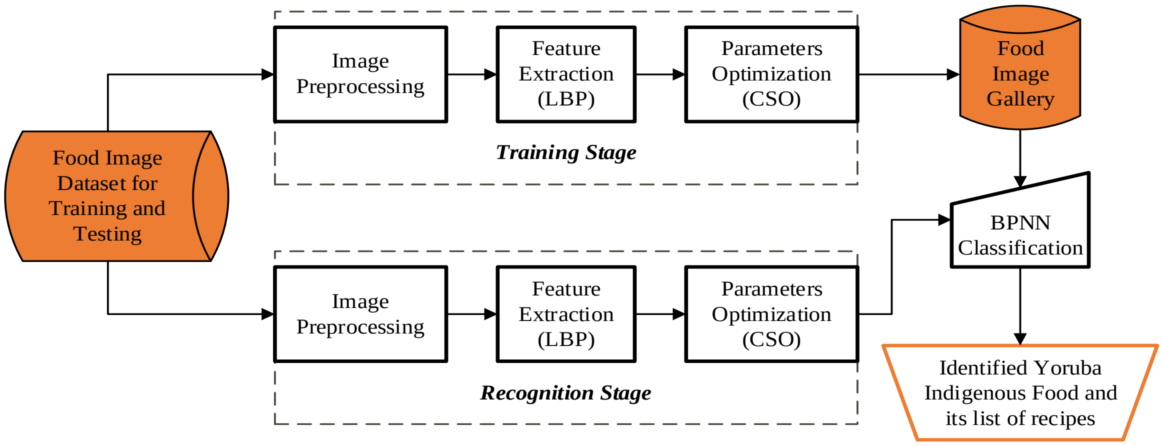

Figure 1: Description of the Process Flow of the Food Recognition System.

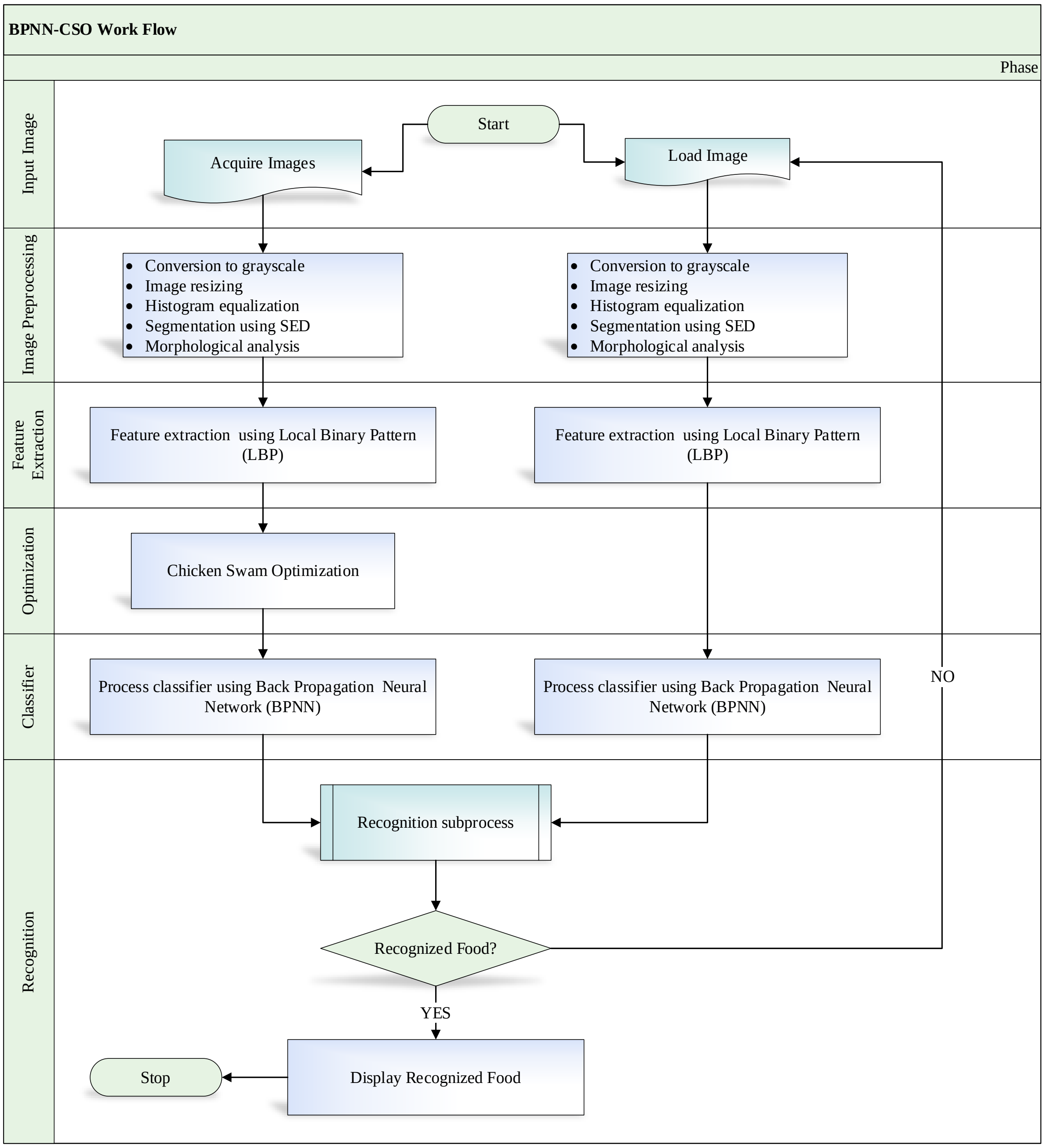

Figure 2: Flowchart of BPNN-CSO Training and Testing Model.

Figure 3: Data sample obtained for Agbado category

Figure 4: Data sample obtained for Ewa category

Figure 5: Data sample obtained for Isu category

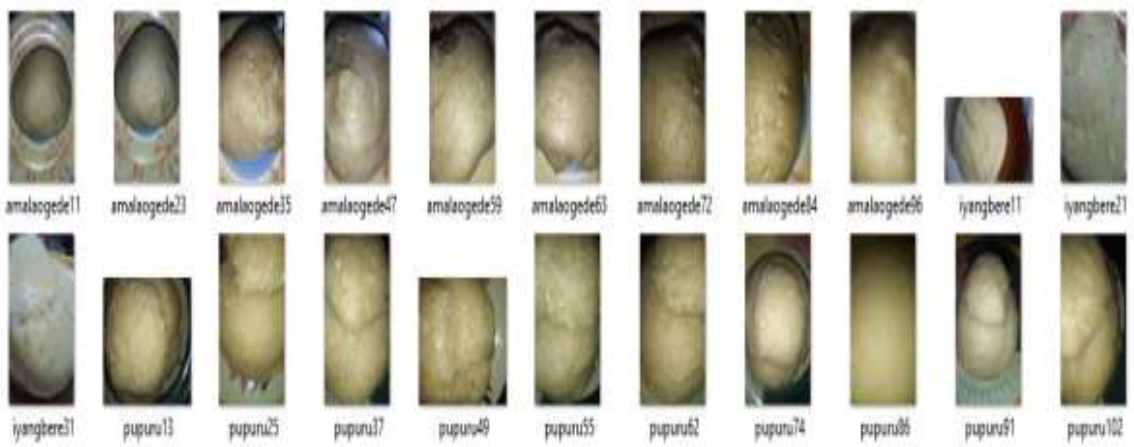

Figure 6: Data sample obtained for Okele category

3.2 Image Acquisition

The images of locally, cooked food were captured by Digital Camera. The camera had an angle of about 25 degrees. Only ambient fluorescent illumination was used to prevent violent shadows. Each object was photographed several times, once in a five-degree rotation. All images were histograms stretched, that is, the intensity of the brightest pixel was 255 and the intensities of the other pixels were scaled.

3.3 Image Pre-processing

For the most part, the data collected is not in useable form. Consequently, it is essential that the correct data format is introduced into the machine learning algorithm in order for the problem to be solved. The study ensured that these datasets were resized, converted to grayscale, filtered to prevent interference with image pre-processing and, adjusted contrast and brightness to compensate for non-uniform image illumination.

3.4 Conversion to Grayscale

The images acquired were in RGB and were to be converted to grayscale (two-dimensional form (2-D)) with a pixel value between 0 and 255. Each of the grayscale images was expressed and stored as a matrix in MATLAB that was converted into a vectorial image for other processes. The conversions to food vector were made to aid the normalization process.

3.5 Normalization of Food Image

The normalization of the images was carried out by applying the histogram equalization technique to the converted grayscale images to improve the contrast in the images by stretching out the intensity range. This enhances the brightness in the grayscale images for a clearer view of the food of each type. The normalization phase removes some common features that all the food images shared together so that each food image is left with unique features. The common features were discovered by finding the average food vector of the whole training set (food images). Then, the average food vector was subtracted from each of the food vectors which results in a normalized food vector.

3.6 Feature Extraction

Local Binary Pattern (LBP), a simple and commonly utilized texture analysis approach, was employed to extract image features. In texture images, the LBP operator encodes the pixel-wise information. A local pattern in the LBP technique describes the associations between a pixel and its neighbors. Eight local neighborhood pixels were used. All neighbors that have a higher value were given a value of 1, while all those lower values were given a value of 0. The binary values associated with the neighbors were then acquired consecutively, clockwise, to generate a binary number that was used to characterize the local texture. Equation1 explains the LPB algorithm.

LBP operator is defined as $LPB_{P,R}u^2$. The (P,R) neighborhood is represented in subscript. U_2 stands for using only uniform patterns and labeling all other patterns with a single label. As soon as a labeled image fl(x,y) has been generated, the LBP histogram can be defined as

Hi = Histogram

$$H_i = \sum x, yl\{fl(x, y = i)\}, \text{for } i = 0\ldots\ldots, n-1 \quad (1)$$

where n is the number of labels that the LBP operator produces, it is necessary to normalize the histograms as:

$$N = \mathrm{H_i} \sum \mathrm{n - 1j = 0H_j} \quad (2)$$

Where N= number of the total label that LBP operator generated.

3.7 Parameters Optimization

To achieve the "best" design in terms of a set of priority criteria or constraints, optimisation must be carried out on the dataset. Productivity, robustness, reliability, longevity, efficiency, accuracy and use are just some of the factors that need to be considered to maximize outcomes. Optimisation problem: Increasing or decreasing the value of a function relative to a set, which is often a representation of the range of options available in a given situation. In order to determine which options are "best", the function lets users compare and contrast options. Examples of common applications include cost minimisation, profit maximization, error minimisation, optimal design, optimal management and principles of variation.

Chicken Swarm Optimization (CSO) is an innovative optimization algorithm that takes its inspiration from biological systems. The hierarchical orders within the chicken swarm and the behaviour of the chicken swarm were replicated. The chicken swarm can be divided into several groups, each of which consists of a rooster and a large number of hens and chicks. Different chickens are transported according to different laws of movement [7]. Chicken behaviour varies with gender. The leader rooster aggressively hunted for food and battled the chickens invading the territory of the group. The dominant chickens would forage with the head roosters. In order to find food, the rooster will shift its position in accordance with the Equations (3-4). The new rooster position will be equal to the sum of the old rooster position and the randomized old rooster position. The product of a normally distributed random number and the rooster's old position is referred to as randomized old position [8].

$$x_{ij}^{t+1} = x_{i,j}^{t} * \left(1 + Randn(0, \sigma^2)\right) \qquad (3)$$

$$\sigma^2 = \begin{cases} 1 & \text{if } f_i < f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \in}\right) & \text{otherwise}, k \neq i \end{cases} \qquad (4)$$

Where $x_{i,j}^{t+1}$ and $x_{i,j}^{t}$ denote the new and old positions of the *ith* rooster in the *jth* dimension, respectively; *t* denotes the iteration number; Randn denotes the normal distribution with a mean of zero and a standard deviation of two. The second equation yields the value of 2, where f_i and f_k are the fitness values of rooster *i* and rooster *k*, respectively. The value of *k* is chosen at random from the other groups. To avoid indefinite values, a small amount is added to the denominator. Equations 4 and 5 are used to change the positions of the hens.

$$x_{ij}^{t+1} = x_{i,j}^{t} + S1 * \text{Rand1} * \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + \text{S2} * Rand2 * \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \qquad (5)$$

$$S1 = \exp\left(\frac{f_i - f_{r1}}{|f_i| + \in}\right) \qquad (6)$$

$$S2 = \exp\left(f_{r2} - f_i\right) \qquad (7)$$

In Equation (5), $x_{i,j}^{t+1}$ and $x_{i,j}^{t}$ denote the new and old position of *ith* hen in *jth* dimension, respectively; Equations (5), (6) and (7) determines the final position of the chicks by the equation (8):

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \text{FL} * \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \qquad (8)$$

$x_{i,j}^{t+1}$ and $x_{i,j}^{t}$ a chick's mother is chosen at random from among the group's two hens, and FL is assigned a random number between 0 and 2. The swarm is divided into groups of three, with a rooster and two hens assigned to each group. The remaining chickens are chicks.

Local Binary Pattern – Chicken Swarm Optimization (LBP-CSO)

In this study, the extracted feature vector from Local binary Pattern was optimized using Chicken Swarm Optimization in order to produce optimized parameters to reduce the convergence rate of Back Propagation Neural Network. Algorithm 1 shows details of how the LBP-CSO algorithm works.

Algorithm 1. The Developed LBP-CSO Algorithm

Step 1: *Set g_c which corresponds to the gray value of the center pixel*

Step 2: *Set g_p as the gray values of the "n" neighbour pixels*

Step 3: Set $M = \begin{Bmatrix} 1, & if\ g_c \geq 0 \\ 0, & if\ g_c < 0 \end{Bmatrix}$

Step 4: *Compute LBP features as described thus;*

$$LBP_{p,r}(x_c, y_c) = \sum_{p=0}^{n-1} M(g_p - g_c) * 2^p$$

Where x_c and y_c represent the horizontal and vertical component of the image; Mg_p and Mg_c are neighborhood patterns, P represent the bit binary number resulting in 2^P distinct values for the LBP code. P

Step 5: *Output selected LBP features*
Step 6: *Generate the initial parameters, the total population size, the roosters accounts Nr, the hens accounts Nh, the mother hens accounts Nm, updating frequency of the chicken swarm G and the maximum number of generations itermax*
Step 7: *Generate a population $x = (x_1, x_2, \dots, x_i, \dots, x_{popsize})$ of popsize chickens with random solutions.*
Step 8: *Calculate the fitness (x_i) and find the best solution x_{best} of the population.*

Step 9: *for iter = 1: itermax*
Step 10: *If iter%G == 1 or iter = = 1*
Step 11: *Sort all population individuals according to their fitness.*
Step 12: *Divide total population individuals into three subpopulations (Rooster population, hen population, and chicken population) according to their sort criteria, and establish the relationship between the chicken and its mother (hen).*
Step 13: *End*
Step 14: *Update the rooster population individuals according to Equation (5)*
Step 15: *Update the hen population individuals according to Equation (7)*
Step 16: *Update the chicken population individuals according to Equation (8)*
Step 17: *Calculate the fitness of each member in the population.*
Step 18: *Update the personal best position $x*i$ and the global optimal position $x$best.*
Step 19: *Perform local search for the global optimal individual.*
Step 20: *End*
Step 23: *Output the best solution $x$best*

Classification using BPNN

The optimized parameters by Chicken Swarm Optimization technique were classified using Back Propagation Neural Network (BPNN). This technique was employed to measure the similarity between the test vector and the reference vectors in the gallery. Back propagation is a supervised learning algorithm for artificial neural networks using gradient descent. The method calculates the gradient of the error function with respect to the weights of an artificial neural network. The delta rule for perceptron is extended to multilayer feed forward neural networks. The name "backwards" comes from the fact that the gradient is calculated backwards, starting with the final layer of weights and ending with the first layer of weights.

Partially computed gradients from one layer are reused in the next layer's gradient computation. Rather than calculating the gradient of each layer separately, this backwards flow of error information allows for efficient gradient computation. Equation of feed forward neural network which must take place first before back propagation neural network takes place, which all processes are done in abstraction.

Error function in classic BPNN is the mean squared error

$$\mathrm{E}(\mathrm{X},\theta)=\frac{1}{2\mathrm{N}}\sum_{i=1}^{N}(\hat{y}-yi)^{2} \qquad (9)$$

$$a_i^k=\mathrm{b}_{\mathrm{i}}^{\mathrm{k}}+\sum_{\mathrm{j=1}}^{\mathrm{rk=1}}\mathrm{w}_{ji}^{k}\,o_j^{k-1}=\sum_{j=0}^{rk=i}w_{ji}^{k}o_j^{k=1} \qquad (10)$$

Input – Output gradient derivation

$$\mathrm{E}(\mathrm{X},\theta)=\frac{1}{2\mathrm{N}}\sum_{i=1}^{N}(\hat{y}-yi)^{2} \qquad (11)$$

Sum of each derivative function

$$\frac{\partial E(X,\theta)}{\partial w_{ij}^{k}}=\frac{1}{N}\sum_{d=1}^{/n}\frac{\partial}{\partial w_{ij}^{k}}\left(\frac{1}{2}(\hat{yd}-yd)^{2}\right)=\frac{1}{N}\sum_{d=1}^{/n}\frac{\partial E_d}{\partial w_{ij}^{k}} \qquad (12)$$

$$E=\frac{1}{2}\ (\hat{yd}-yd)^{2} \qquad (13)$$

Application of chain rule to error function partial derivation

$$\frac{\partial E}{\partial w_{ij}^{k}}=\frac{\partial E}{\partial a_j^{k}}\ \frac{\partial a_j^{k}}{\partial w_{ij}^{k}} \qquad (14)$$

1.) $\delta_j^k \equiv \ \frac{\partial E}{\partial a_j^k}$ (15)

2.) $\frac{\partial a_j^k}{\partial w_{ij}^k}=\frac{\partial}{\partial w_{ij}^k}\left(\sum_{j=0}^{rk=i}w_{ji}^{k}o_j^{k=1}\right)=\mathrm{o}_{\mathrm{i}}^{\mathrm{k=1}}$ (16)

$$\frac{\partial E}{\partial w_{ij}^{k}}=\delta_j^k\, o_j^{k-1} \qquad (17)$$

The Output Layer

$$E=\frac{1}{2}(\hat{y}-yi)^{2}=\frac{1}{2}(g_o(a_1^m)-y)^{2} \qquad (18)$$

$$\delta_j^k=(g_o(a_1^m)-y)g_o^{/}(a_1^m)=(\hat{y}-y)g_o^{/}(a_1^m) \qquad (19)$$

$$\frac{\partial E}{\partial w_{i1}^{m}}=\delta_i^m\, o_i^{m-1}=\ (\hat{y}-y)g_o^{/}(a_1^m)\, o_i^{m-1} \qquad (20)$$

The Hidden Layer

$$\delta_J^K = \frac{\partial E}{\partial a_j^k} = \sum_{l=1}^{rk+1} \frac{\partial E_d}{\partial a_l^{k+i}} \frac{\partial a_l^{k+1}}{\partial a_j^k} \quad (21)$$

$$\delta_j^k = \sum_{l=1}^{r^{k+1}} \delta_l^{k+1} \frac{\partial a_l^{k+1}}{\partial a_j^k} \quad (22)$$

$$a_l^{k+1} = \sum_{j=1}^{r^k} w_{jl}^{k+1} \, g(a_j^k) \quad (23)$$

$$\frac{\partial a_l^{k+1}}{\partial a_j^k} = w_{jl}^{k+1} g'(a_j^k) \quad (24)$$

$$\delta_j^k = \sum_{l=1}^{r^{k+1}} \delta_l^{k+1} W_{jl}^{k+1} g'(a_j^k) = g'(a_j^k) \sum_{j=1}^{r^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (25)$$

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_j^k o_i^{k=1} = g'(a_j^k) o_i^{k=1} \sum_{j=1}^{r^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (26)$$

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_j^k o_i^{k=1} \quad (27)$$

$$\delta_1^m = (g_o'(a_1^m))(\hat{y}_d - y_d) \quad (28)$$

$$\delta_j^k = g'(a_j^k) \sum_{l=1}^{r^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (29)$$

$$\frac{\partial E(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^{N} \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{2} (\hat{y}_d - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^{N} \frac{\partial E_d}{\partial w_{ij}^k} \quad (30)$$

Updated Weight

$$\Delta w_{ij}^k = -\alpha \frac{\partial E(X,\theta)}{\partial w_{ij}^k} \quad (31)$$

Where w_{ji}^k : weight for node j in layer l_k for incoming node I

b_i^k : bias for node I in layer l_k

a_j^k : product sum plus bias for node i in layer l_k,

o_i^k : Output for node i in layer l_k,

rk : number of nodes in layer l_k .

g: activation for the hidden layer nodes,

g_o: activation for the outer layer nodes.

E: change in Error function,

X: input and output pairs (combination of error function)

m: Number of layers.

The Algorithm for Back Propagation Neural Network is given as:

Algorithm 2: Back Propagation Neural Network

Step 1:	*Calculate the forward phase for input-output pair ((*x_d*,* y_d*) and store the result* y_d, a_j^k and a_j^k *for each node layer k by proceeding from layer 0, the input layer m, the output layer.*
Step 2:	*Calculate the backward phase for each input-output pair (*x_d*,*y_d*) and store the result* $\frac{\partial E_d}{\partial w_{ij}^k}$ *for each weight* w_{ij}^k *connecting node I in layer k-1 to node j in layer k by preceding from layer m, the output layer, to layer 1, the input layer* *(a) Evaluate the error term for the final layer* ∂_1^m *by using the second equation* *(b) Back propagate the error terms for the hidden layers* d_j^k *working backwards from the final hidden layer k=m by repeated using the third equation* *(c) Evaluate the partial derivatives of the individual error* E_d *with respect to* w_{ij}^k *by using the first equation.*
Step 3:	*Combine the individual gradients for each input-output pair* $\frac{\partial E_d}{\partial w_{ij}}$ *to get the total gradient* $\frac{\partial E(X,0)}{\partial w_{ij}^k}$ *for the entire input-output pairs* $X = \{(x_1, y_1)\ldots.,(x_N, y_N)\}$ *by using the fourth equation.*
Step 4:	*Update the weight according to the learning rate* α *and total gradient* $\frac{\partial E(X,0)}{\partial w_{ij}^k}$ *by using the fifth equation direction of the negative gradient.*

4. Results and Discussion

The implementation phase begins with the collection of some Yoruba Indigenous food images from Kwara – South part of Kwara State in Nigeria. The food images were loaded into matlab for processing. The features extracted by Local Binary Pattern were optimized by Chicken Swarm Optimization techniques then finally into Back Propagation Neural Network for classification. For the comparison of the results of BPNN and BPNN-CSO. The features extracted by Local Binary Pattern go into Back Propagation Neural Network for classifying the images. The training of the developed system with the dataset was done before the testing stage.

The results of the BPNN and BPNN-CSO were validated using 1,251 dataset of the varieties of the prepared food. All the food categories were tested on the system with BPNN alone, and the same categories of food were also tested on the BPNN-CSO. The result showed significant improvement in terms of recognition accuracy, sensitivity, and specificity. From the result, it was observed that the recognition time of BPNN was reduced compared to the recognition time of BPNN-CSO except for the solid food categories like Okele and Agbado. Details of the results are shown in Table 1, 2 and 3 respectively, The graphical representation of the results are shown in Figure 7,8 and 9 respectively.

Table 1: Result of Back Propagation Neural Network on the Indigenous Food Dataset without Optimization.

Food Type	Recognition Accuracy	Sensitivity	Specificity	Recognition Time
Okele	**82.2222**	**87.9630**	**76.9231**	**816.7073**
Ewa	**87.5949**	**92.2222**	**83.7209**	**1.033**
Agbado	**83.6735**	**90.2255**	**75.8929**	**771.0660**
Isu	**85.3448**	**90.1316**	**81.6327**	**1.7188**

Table 1. shows the result of Backpropagation Neural Network on the indigenous food dataset which are 1,251 in total collected without optimizing the considered parameters which are Recognition accuracy of about 82% - 88% , sensitivity rate of 88% - 92% , specificity rate of 76% - 84%, and false positive ratio varies from 16.3 – 24.1 on the four categories of foods mentioned namely; okele 198 (Amala oogede, Iyan gbere, pupuru), ewa 395 (akara,dele,ekuru,monu,aseke), agbado 246 (abari,sapala,egbo) and isu 312 (ojojo ewura,ojojo koko, ojojo isu, asaaro, ikokore). The conclusion of the result is well represented in Figure 7.

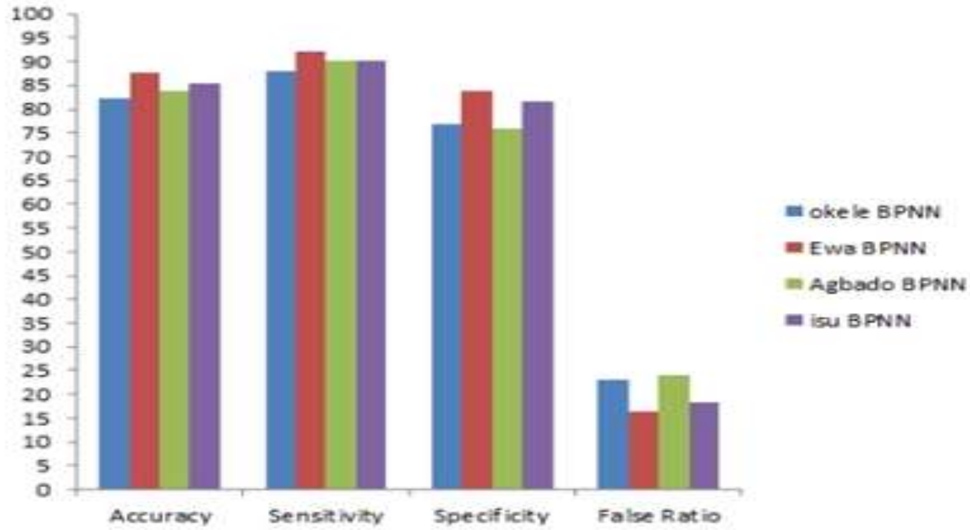

Figure 7: Graphical Representation of BPNN result

Table 2: Back Propagation Neural Network with Chicken Swarm Optimization Result

Food Type	Recognition Accuracy	Sensitivity	Specificity	Recognition Time
Okele	**85.3448**	**90.1316**	**81.6327**	**819.9031**
Ewa	**91.8987**	**97.7778**	**86.9767**	**1.037**
Agbado	**91.0204**	**97.7444**	**83.0357**	**774.2634**
Isu	**90.2299**	**96.7105**	**85.2041**	**1.7183**

Table 2. illustrates the result of Backpropagation Neural Network with Chicken Swarm Optimization on the indigenous food dataset which area 1251 in total collected with optimizing the considered parameters which are Recognition accuracy of about 85% - 92% , sensitivity rate of 90% - 98% , specificity rate of 82% - 87%, and false positive ratio varies from 13 – 18 on the four categories of food mentioned namely; okele 198 (Amala oogede, Iyan gbere, pupuru), ewa 395 (akara,dele,ekuru,monu,aseke), agbado 246 (abari,sapala,egbo) and isu 312 (ojojo ewura,ojojo koko, ojojo isu, asaaro, ikokore). The conclusion of the result is well represented in Figure 8.

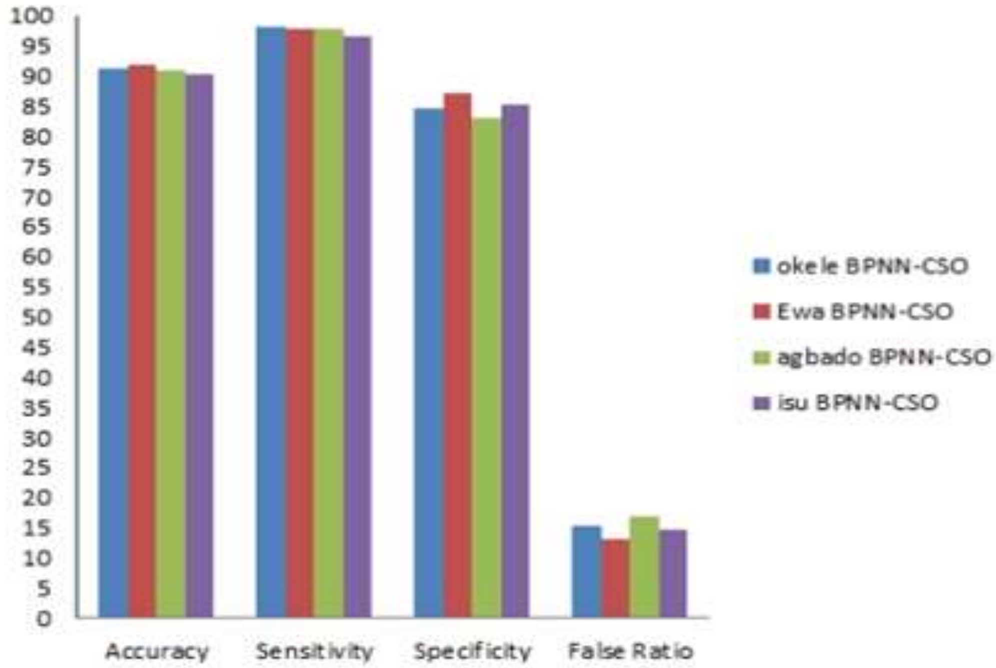

Figure 8: Graphical Representation of BPNN result

Table.3: Comparison of BPNN with BPNN-CSO

Food Type	Algorithm	Accuracy	Sensitivity	Specificity	False Ratio	R.Time
Okele	BPNN	**82.2222**	**87.9630**	**76.9231**	**23.0769**	**816.7073**
	BPNN-CSO	**91.1111**	**98.1481**	**84.6154**	**15.3846**	**819.9031**
Ewa	BPNN	**87.5949**	**92.2222**	**83.7209**	**16.2791**	**1.0333e+03**
	BPNN-CSO	**91.8987**	**97.7778**	**86.9767**	**13.0233**	**1.0374e+03**
Agbado	BPNN	**83.6735**	**90.2255**	**75.8929**	**24.1071**	**771.0660**
	BPNN-CSO	**91.0204**	**97.7444**	**83.0357**	**16.9643**	**774.2634**
Isu	BPNN	**85.3448**	**90.1316**	**81.6327**	**18.3673**	**1.7188**
	BPNN-CSO	**90.2299**	**96.7105**	**85.2041**	**14.7959**	**1.7183**

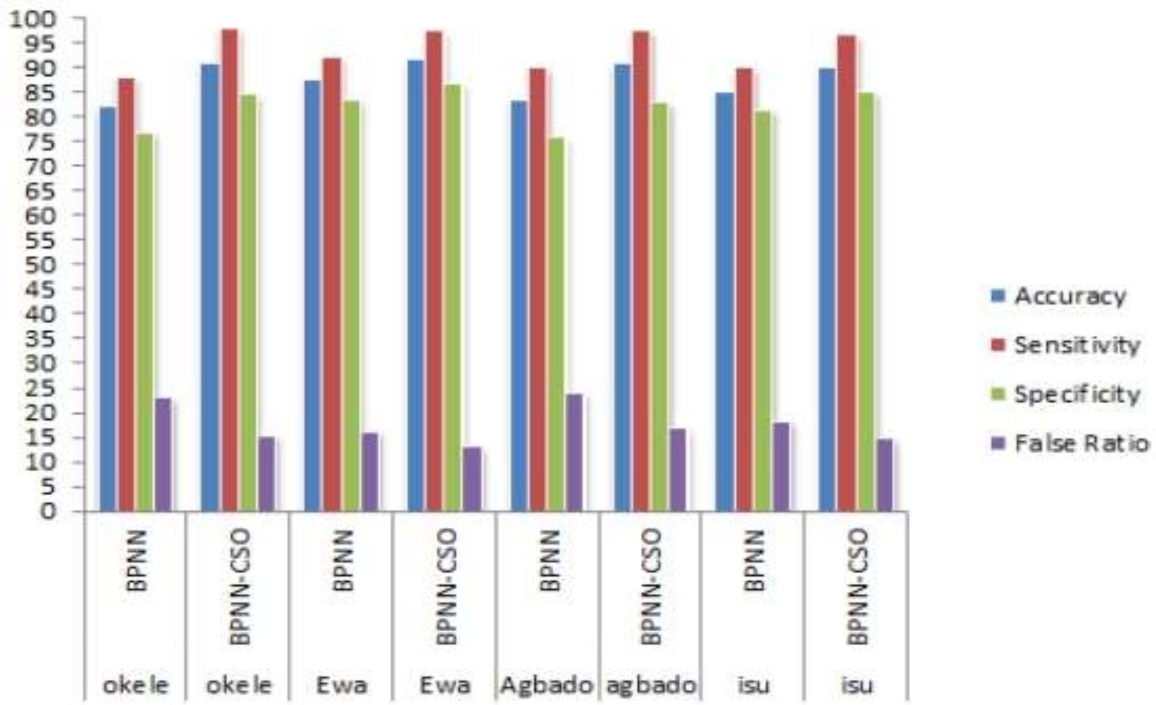

Figure 9: Graphical representation of BPNN and BPNN-CSO RESULT.

Table 3 illustrates Comparison of Backpropagation Neural Network result and Back Propagation Neural Network Chicken Swarm Optimization result on the indigenous food dataset which area 1251 in total collected. The considered parameters which are Recognition accuracy of about 82% - 88% , sensitivity rate of 88% - 92% , specificity rate of 76% - 84%, and false positive ratio varies from 16.3 – 24.1 for BPNN outcome while Recognition accuracy of about 85% - 92% , sensitivity rate of 90% - 98% , specificity rate of 82% - 87%, and false positive ratio varies from 13 – 18 for BPNN-CSO on the four categories of food mentioned namely; okele 198 (Amala oogede, Iyan gbere, pupuru), ewa 395 (akara,dele,ekuru,monu,aseke), agbado 246 (abari,sapala,egbo) and isu 312 (ojojo ewura,ojojo koko, ojojo isu, asaaro, ikokore). The conclusion of the result is well represented in figure 9, as the parameters were slightly changed in data on the graph.

5. Conclusion

The research work developed Yoruba indigenous Food recognition system using CSO with LBP and BPNN. A total of 1,251 datasets of Yoruba indigenous food images were used to test the proposed classification scheme. When compared to several other methods, experimental results show that BPNN-CSO has a 92.00% recognition accuracy with 1.0304sec as a training time , while BPNN recognition accuracy ranges from 82.6% to 87.54% with longer delay computational time for okele and agbado food categories. The optimization techniques applied to the BPNN parameters resulted in an increase in recognition accuracy for BPNN-CSO. From the results obtained, it is evident that CSO has improved the performance of BPNN.

6. Recommendation

This research work can be extended to other local food in the six geopolitical zones of Nigeria and other African countries in order to preserve and prevent their local foods from going into total extinction.

References

[1] Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., ... & Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, *241*, 111716.

[2] Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., and Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE Journal of Biomedical and Health Informatics*, *18*(4), 1261–1271. https://doi.org/10.1109/JBHI.2014.2308928

[3] Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., & Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE journal of biomedical and health informatics*, *18*(4), 1261-1271.

[4] Dehais, J., Anthimopoulos, M., & Mougiakakou, S. (2016, October). Food image segmentation for dietary assessment. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management* (pp. 23-28).

[5] Kagaya, H., Aizawa, K., & Ogawa, M. (2014, November). Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 1085-1088).

[6] Tahir, G. A., & Loo, C. K. (2021, December). A comprehensive survey of image-based food recognition and volume estimation methods for dietary assessment. In *Healthcare* (Vol. 9, No. 12, p. 1676). Multidisciplinary Digital Publishing Institute.

[7] Meng, X., Liu, Y., Gao, X., and Zhang, H. (2014). *A New Bio-inspired Algorithm: Chicken Swarm Optimization. A New Bio-inspired Algorithm: Chicken Swarm Optimization*. *October*. https://doi.org/10.1007/978-3-319-11857-4

[8] Bharanidharan, N., & Rajaguru, H. (2020). Improved chicken swarm optimization to classify dementia MRI images using a novel controlled randomness optimization algorithm. *International Journal of Imaging Systems and Technology*, *30*(3), 605-620.