A Self-governing and Novel Hybrid Approach for all-inclusive Leaf Disease Detection using IoT and Robotics

Author 1: (Corresponding Author)
Arivukkarasan. R
Research Scholar,
Bharath Institute of Higher Education and Research
arivukkarasan@gmail.com

Author 2: (Guide)
Dr.Raguraman. D
Associate Professor,
Bharath Institute of Higher Education and Research
raguraman.auto@bharathuniv.ac.in

Abstract
Only when crops are handled efficiently are the insights they provide translated into profitable decisions. Current advancements in data management are causing Smart Farming to expand dramatically, since data has become the most important aspect of modern agriculture, assisting farmers in making crucial decisions. With the goal of improving production and sustainability, objective information gathered by sensors yields valuable benefits. This type of data-driven farm management relies on data that can boost productivity by preventing the waste of resources and the contamination of the environment. Data-driven agriculture, aided by robotic solutions using techniques of artificial intelligence, lays the foundation for the agricultural of the future. The Internet of Things (IoT) and other cloud-based solutions, along with the ever-evolving state of the internet world, present a once-in-a-generation window of opportunity for the creation of automated and robotic systems in the fields of urban agriculture, agriculture, and forestry. The advent of GPS, machine vision, lasers, and mechatronics have all contributed to the creation of cutting-edge robotic systems and intelligent technologies for use in precision farming. This study showcased a flexible Internet of Things (IoT) robotic system for greenhouse cultivation applications, with the goal of real-time illness diagnosis in plants. Instead with ADAM, the small robotic system employs a BRCNN (Backpropagation Recurrent Convolution Neural Network) module trained with MFO (Moth Flame Optimizer) as its Primary Optimizer to identify the presence of disease on a plant and SVM to characterize the type of sickness. Further, the system provides appropriate feedback remedies if and when they are required. There is no longer any need to spend a ton of money to examine plants and conduct thorough chemical research because this method permits a quick and preliminary classification of plant illnesses. The categorization findings show that the approach has a high rate of success in identifying plant diseases and placing them into categories. The utilization of the system has demonstrated that it is able to supply the user with real-time feedback by making use of a comprehensive database of treatments for each individual instance of plant ailment. Because we want to expand the plant disease dataset at some point in the future, the training dataset that we are using is currently being updated with new plant images.

Keywords: plant disease, MFO, BRCNN, agriculture.

1. Introduction

Developing countries mainly rely on agricultural products for their economy. But plant yield is greatly affected due to diseases, which produce a huge financial loss as well as a threat to food security. The traditional approach involves naked eye observation of plant diseases by experts as well as by farmers. Rapid identification of disease is a challenging task as many farmers lack expertise and the disease vary from plant to plant as well as varies at each stage of growth, which requires continuous monitoring and is expensive for large farms. Further many farmers are unaware of non-native diseases. Moreover, rural areas lack experts which force them to go long distances to get a suitable solution. Recent technological advancements paved a way for a computer-aided approach to automatically detect and classify diseases. A method to improve the classification accuracy using novel training function in back propagation recurrent CNN (BRCNN)

Geoffrey Hinton is known as the father of Deep Learning with his contribution to the discovery of Backpropagation algorithm [1]. Any standard multi-layered neural network uses this Backpropagation algorithm to train the model in supervised way and solve the problem. Backpropagation is an efficient technique to adjust the weights assigned to every neuron in the network, in such a way that the predictions are closer to accuracy [2]. Though several research works are conducted to solve a variety of computer vision tasks, a dedicated network architecture to handle image data was first developed by Yann LeCun. Yann LeCun proved that Convolutional Neural Network (CNN) excels in object recognition in image data [3].

This study demonstrated a greenhouse cultivation IoT robotic system for real-time plant disease diagnosis. The tiny robotic system uses a BRCNN (Backpropagation Recurrent Convolution Neural Network) module with MFO (Moth Flame Optimizer) as its Primary Optimizer to detect plant disease and SVM to classify it. The technology also gives necessary feedback solutions. The method quickly classifies plant illnesses, saving money on plant examination and chemical analysis. The method classifies plant diseases with an average success rate, according to classification data.

1. Related work

There are several approaches mentioned in the literature regarding detection of abnormalities in leaves. Traditional approaches suffer from specular reflections which drastically reduce the segmentation accuracy of the diseased region (Barbedo 2016). Many works cited in literature are subjected to constraints such as, not properly able to discriminate veins and diseased region, shape of the infected region, background complexity and colour (Barbedo 2014). The proposed method is immune to various constraints mentioned in literature.

Petrellis (2019) presented a plant disease diagnosis method to detect citrus diseases that can be realised with the application available in a mobile phone without using a remote server. The test images were taken using regular smart phone cameras with resolutions ranging from 5 to 23 MP, and then resized to 1024x576 pixels for high-speed processing. The images are snapped with a background much brighter or darker than the foreground image. It is assumed that the normal plant part and the disease spot has a distinct colour and each disease spot is surrounded by a halo. The features are extracted using weka tool and a simple fuzzy is used for the classification. A grade is given to the examined leaves based on the feature comparison. Because the lesion areas can be recognized interactively with such high precision, it is possible to achieve an accuracy of 90% in the majority of cases.

Soybean leaf photos were studied by Kaur et al. (2018), who used the semi-automatic rule-based concepts of k-means to identify healthy leaves from those affected by diseases such frog eye, downy mildew, and Septoria leaf blight. Using SVM classifiers for color features, texture features, and hybrids thereof, the approach is tested on 4775 photos taken from the Plant Village dataset. The average accuracy for all feature combinations is a respectable 90%.

Improved Fuzzy C-means (FCM) algorithm was proposed by Bai et al. (2017) to enhance cucumber leaf spot disease extraction in the presence of complex backgrounds. The intended leaf is separated from the rest of the forest by means of a marked-watershed algorithm. Instead of just using the gray information of the image without taking into account the contributor pixel's spatial information, as is done in FCM grayscale, the neighbourhood mean gray value of each pixel is determined as a sample point. The matrix weights the neighbourhood mean gray value and the gray value of each individual pixel in order to determine the robustness and accuracy. For 129 photos of cucumber illness, analysis revealed an average segmentation error of 0.12%.

Pethybridge and Nelson (2015) proposed an interactive smartphone application known as leaf doctor which is used to classify diseased and healthy leaves as well as quantify the percentage of the disease. The application allow users to select the colour of the captured leaf with predefined eight different healthy leaf colours to classify the leaf based on the selection. The study considers six different diseases of varying severity. The images are photographed 30 cm above the specimen with an iPad's 8-megapixel camera with/2.4 aperture

2. Methodology

This Implementation showed a flexible Internet of Things robotic system for greenhouse cultivation applications. The system's goal is to do plant disease diagnosis in real time. The size of the robotic system is relatively small, and it makes use of a BRCNN (Backpropagation Recurrent Convolution Neural Network) module with MFO (Moth Flame Optimizer) as its Primary Optimizer rather than ADAM in order to detect the presence of disease on the plant. It then uses an SVM to make classifications regarding the type of disease the plant is suffering from. In addition, the system provides beneficial remedial measures in the form of feedback whenever they are required. The method can be utilized for the purpose of providing a preliminary classification of plant diseases without the requirement for time-consuming and expensive plant examinations or in-depth chemical investigations. According to the results of the categorization, the method has a high rate of success when it comes to determining which plant diseases exist. The utilization of the system has been proved to be beneficial in providing the user with quick feedback by drawing on a comprehensive database of treatments that are individualized to each plant disease scenario. This has been demonstrated to be the case through testing. Several different research have found evidence to support this assertion. We are now working on increasing the training dataset with additional plant pictures in order to improve our ability to deal with a more extensive collection of plant illnesses.

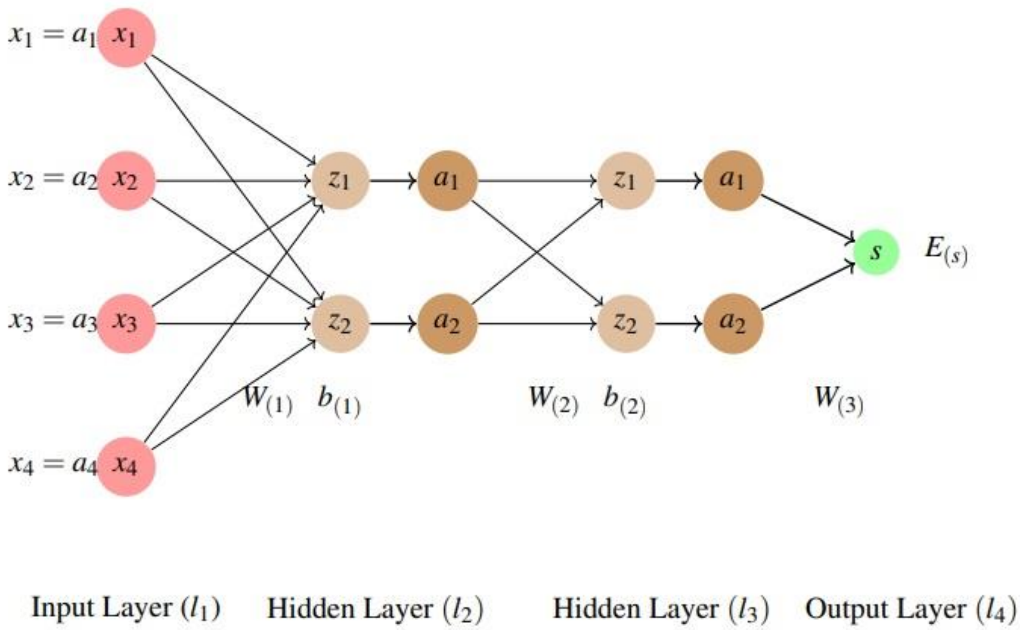

Fig 3. 1 Forward Propagation

Backpropagation is an algorithm used to keep track of the weights of every neuron in the network. It also updates the weights while the input is traversed towards the input layer in backward manner, hence the name Backpropagation, in short for backward propagation. Backpropagation uses chain rule to update the weights and biases for every backward pass [10]

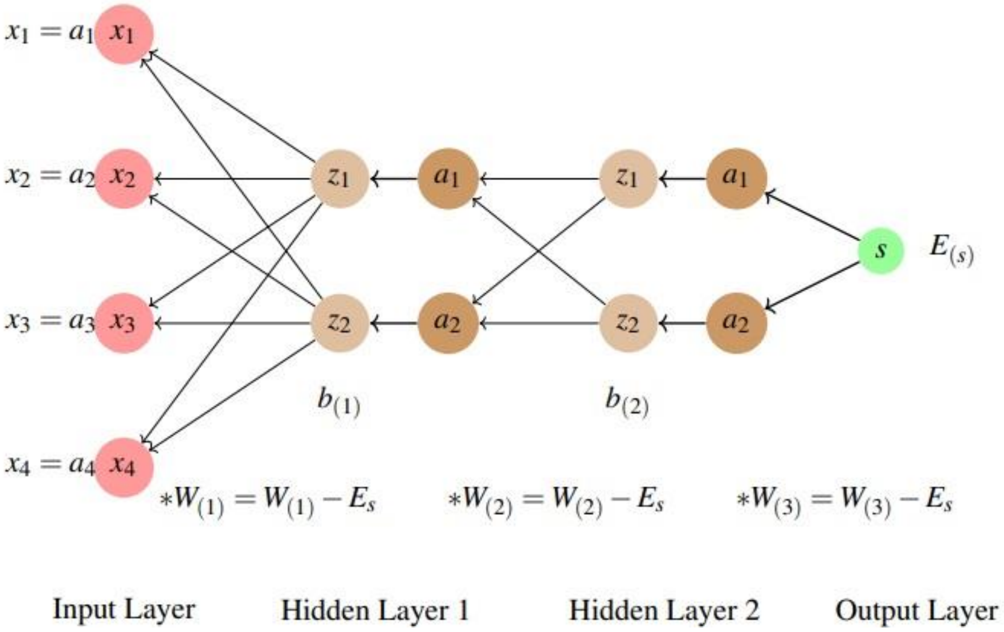

Fig 3. 2 Backpropagation

Algorithm 1: Backpropagation Algorithm

Input:ProblemSize, InputPatterns, Epochs, LearningRate, Momentum, Neural_Network
Output:Neural Network
Function *Main()*
    Neural_Network ← ConstructNetLayers();
    $\text{Neural\_Network}_{weights}$ ← InitalizeWeights (Neural_Network, ProblemSize);
    for *i=1 to Epochs* **do**
        Pattern_i ← SelectInputPattern(InputPatterns);
        Output_i ← ForwardPropagate(Pattern_i, Neural_Network);
        BackPropagateError(Pattern_i, Output_i, Neural_Network, LearningRate, Momentum);
        UpdateWeights (Pattern_i, Output_i, Neural_Network);
    end
    return (Neural Network)

In order to overcome the problem of handcrafted features, deep learning approach based Convolutional Neural Network (CNN) is used to extract features from the leaf images which is then classified using SVM.

Optimizers are used to minimize the loss value, calculated by an objective function, f(x) [12]. The objective function in neural network that is utilized in the output layer is referred to as Loss Function. Based on the results of these loss functions, the Classifier function categorizes the output obtained. A similar function is used in the preceding layers and are referred as Activation Function. The objective function f(x) is dependent on the network model's tuneable parameters weights and biases. The term Objective Function is used in a general scenario, where in an objective function's goal is to either minimize or maximize the given set of information. Loss Function and Activation Function are also a kind of objective function but have more specific purpose and details. Loss Function acts as a developer, adjusts the hyper-parameters in such a way, that it can develop a product close enough to expectation.

"The moth flame optimization algorithm (MFO) was developed by Mirjalili [11]. This algorithm was influenced by natural phenomena. This method was conceived after seeing a natural phenomenon known as "transverse orientation," which involves the navigation of moths. Moth is capable of covering a considerable distance by traveling in a direct route. It is verified both that it travels in a straight line and that it travels in a convoluted pattern around the lights. Due to this behavior of moth it easily converges towards flame and developed as an algorithm named moth flame optimisation algorithm.

Mirjaili is developed recently a new optimizing technique called as MFO [11].This technique depends on the routing of moth in environment. Moths move in night by putting a settled boundary with respective moon, an extremely efficient method for moving in a direct line for lengthy distances. The benefits of MFO technique are straightforward, easy to understand, simplicity of execution. These algorithm workouts real testing issues with in habited and in strange investigate interval. MFO method is amazing as far as high investigation, utilization & escaping of local optima, which inspire the authors to utilize this technique.

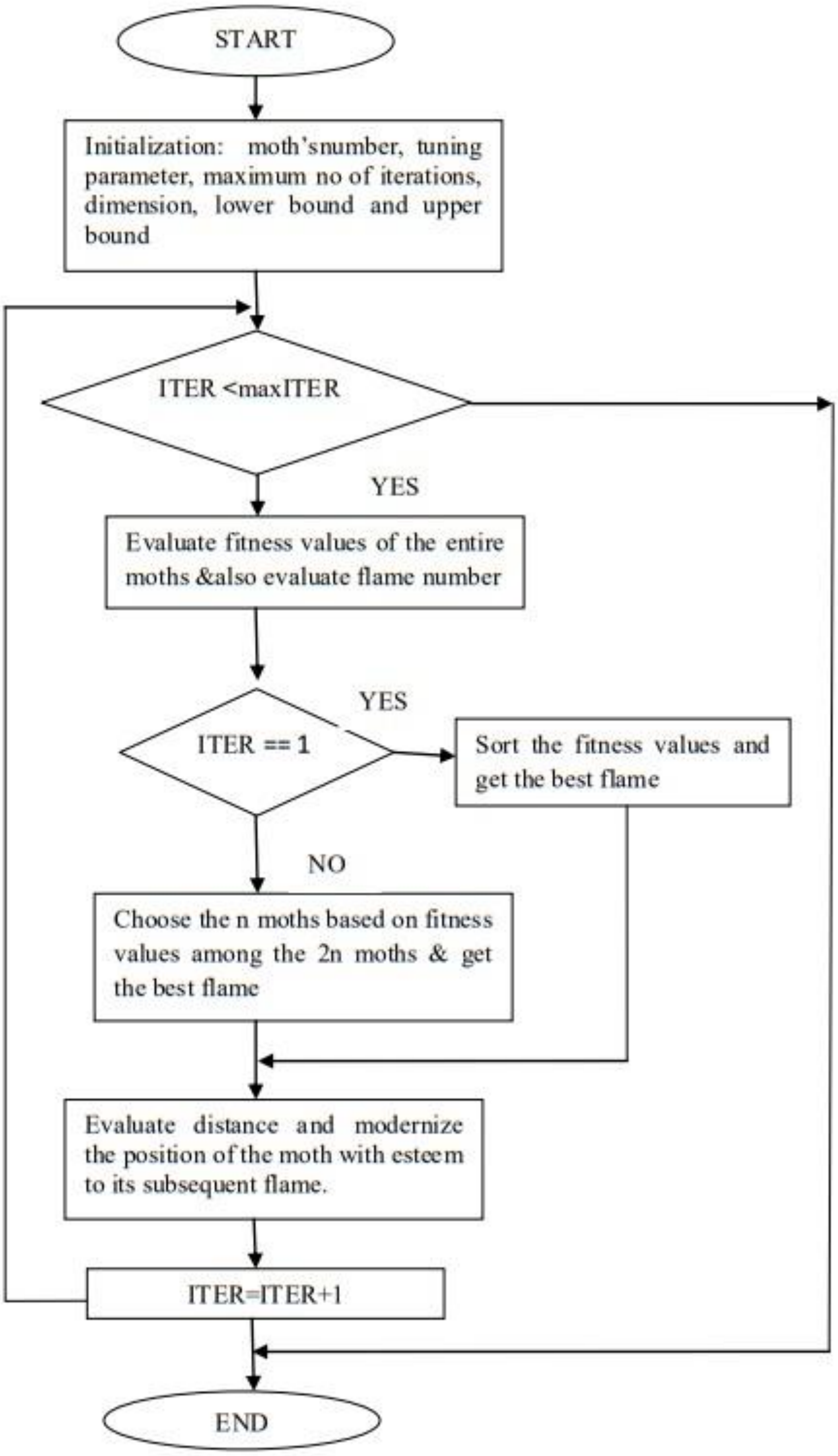

Fig 3. 3 Flow chart of MFO

Dataset Description

The Leaf Disease dataset consists of 6 labels of leaf disease named Bacteria, Fungi, Nematodes, Normal, and Virus.

https://www.kaggle.com/datasets/sizlingdhairya1/leaf-disease

3. Results and Discussion

In this section, we provide the suggested framework's simulation outcomes in detail, graphically. Details on the leaf disease dataset used in the simulation can be found in the section. The following sections detail the results for each dataset individually.

The backpropagation recurrent Convolutional Neural Network architecture and its hyperparameters define the nature of the model and also enable the model to detect leaf disease using SVM for classification. The backpropagation recurrent Convolutional Neural Networks (BRCNN) is used to extract the features from the image and it is evaluated using SVM classifiers.

```
[INFO] Loading images ...
************
[INFO] Processing 01a66316-0e98-4d3b-a56f-d78752cd043f___FREC_Scab 3003.JPG ...
[INFO] Processing 01f3deaa-6143-4b6c-9c22-620a46d8be04___FREC_Scab 3112.JPG ...
[INFO] Processing 03eccb1a-0368-4ac7-9f48-7546037b775a___FREC_Scab 3334.JPG ...
[INFO] Processing 0b170906-9436-4c0d-84c1-c396ad9d909b___FREC_Scab 3101.JPG ...
[INFO] Processing 0b4a52e3-e15e-4117-b2e8-7cdb5dca3ce9___FREC_Scab 3137.JPG ...
[INFO] Processing 0c620ec5-11cf-4120-94ab-1311e99df147___FREC_Scab 3131.JPG ...
[INFO] Processing 0cbfa4fa-63d8-43ce-9385-ff140e524b69___FREC_Scab 3164.JPG ...
[INFO] Processing 0d3c0790-7833-470b-ac6e-94d0a3bf3e7c___FREC_Scab 2959.JPG ...
[INFO] Processing 0d8d5b80-962d-4381-8d3b-9eca3f2f1bb0___FREC_Scab 3449.JPG ...
[INFO] Processing 0db71c1d-93d7-4481-b0d0-b73f995131a9___FREC_Scab 2976.JPG ...
[INFO] Processing 0e90fe4a-e8b6-4186-9429-a9fea180af9a___FREC_Scab 3391.JPG ...
[INFO] Processing 0ea78733-9404-4536-8793-a108c66269b3___FREC_Scab 3145.JPG ...
[INFO] Processing 1a21aabb-6f74-4644-8d9e-a517568b7e9c___FREC_Scab 3095.JPG ...
[INFO] Processing 1a304331-98b5-473f-bbb1-c33b8441052a___FREC_Scab 3296.JPG ...
[INFO] Processing 1a4047d7-23d6-4bba-ba10-b6e7005ea01b___FREC_Scab 3193.JPG ...
[INFO] Processing 1a41bab0-45e0-4dda-a798-9bf4a998f1b6___FREC_Scab 3450.JPG ...
[INFO] Processing 1b1004c8-99e9-4c85-8fe5-b1c11d558cf8___FREC_Scab 3300.JPG ...
```

Fig 4. 1 Loading the dataset

Figure 1 illustrates the loading images on the jupyter notebook for further processing on the dataset.

Transform Image Labels using Scikit Learn'sLabelBinarizer

```
In [8]: label_binarizer = LabelBinarizer()
        image_labels = label_binarizer.fit_transform(label_list)
        pickle.dump(label_binarizer,open('label_transform.pkl', 'wb'))
        n_classes = len(label_binarizer.classes_)
```

Print the classes

```
In [9]: print(label_binarizer.classes_)

        ['Bacteria' 'Fungi' 'Nematodes' 'Normal' 'Virus']
```

Split the dataset into training and testing in the ratio 80:20

Applying BRCNN and MFO Hybrid model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      896

 activation (Activation)     (None, 256, 256, 32)      0

 batch_normalization (BatchN  (None, 256, 256, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 85, 85, 32)       0
 )

 dropout (Dropout)           (None, 85, 85, 32)        0

 conv2d_1 (Conv2D)           (None, 85, 85, 64)        18496

 activation_1 (Activation)   (None, 85, 85, 64)        0

 batch_normalization_1 (Batc  (None, 85, 85, 64)       256
 hNormalization)

 conv2d_2 (Conv2D)           (None, 85, 85, 64)        36928

 activation_2 (Activation)   (None, 85, 85, 64)        0

 batch_normalization_2 (Batc  (None, 85, 85, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 42, 42, 64)       0
 2D)
 dropout_1 (Dropout)         (None, 42, 42, 64)        0

 conv2d_3 (Conv2D)           (None, 42, 42, 128)       73856

 activation_3 (Activation)   (None, 42, 42, 128)       0

 batch_normalization_3 (Batc  (None, 42, 42, 128)      512
 hNormalization)

 conv2d_4 (Conv2D)           (None, 42, 42, 128)       147584

 activation_4 (Activation)   (None, 42, 42, 128)       0

 batch_normalization_4 (Batc  (None, 42, 42, 128)      512
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 21, 21, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 21, 21, 128)       0

 flatten (Flatten)           (None, 56448)             0

 dense (Dense)               (None, 1024)              57803776

 activation_5 (Activation)   (None, 1024)              0

 batch_normalization_5 (Batc  (None, 1024)             4096
 dropout_3 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 5)                 5125

 activation_6 (Activation)   (None, 5)                 0

=================================================================
Total params: 58,092,421
Trainable params: 58,089,541
Non-trainable params: 2,880
```

Moths, which are based on spatial movements, are the potential solutions in this algorithm. Since the MFO algorithm operates on a population basis, the n-moth swarm is used as a search agent in the problem domain. To date, flames have been the best-documented n sites for moths. Each moth, in turn, looks for a better option and revises its flame accordingly. therefore, flames are d-dimensional statistical objects.

Applying SVM for classification

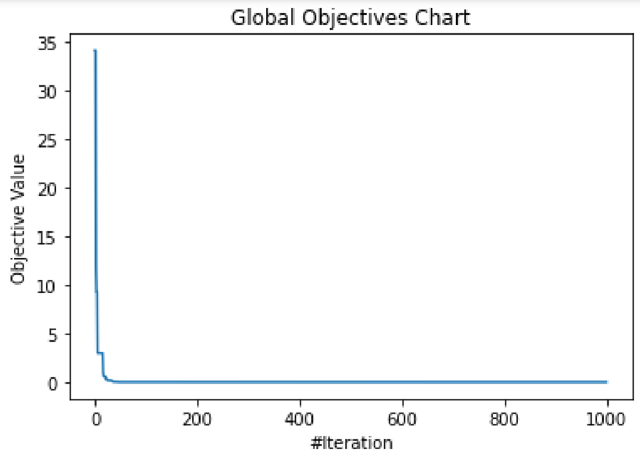

Fig 4. 2 Global Data Execution Trajectory per Epoch

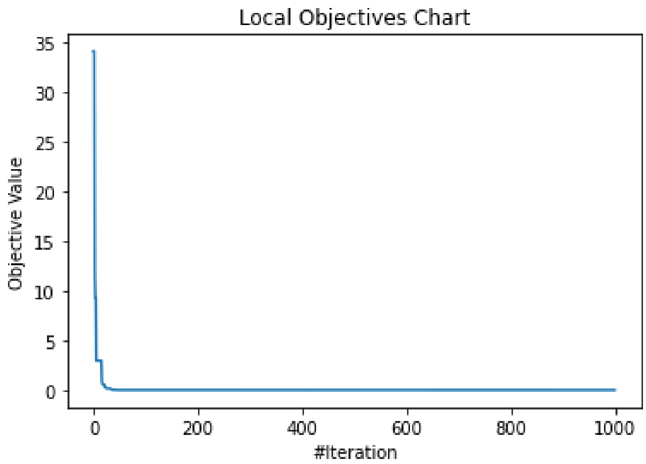

Fig 4. 3 Local Data Execution Trajectory per Epoch

In figure 4.2 and 4.3, the graph shows the trajectory of finding optimum points by clusters of optimizers at local level (inside cluster) and at global level (dataset level).

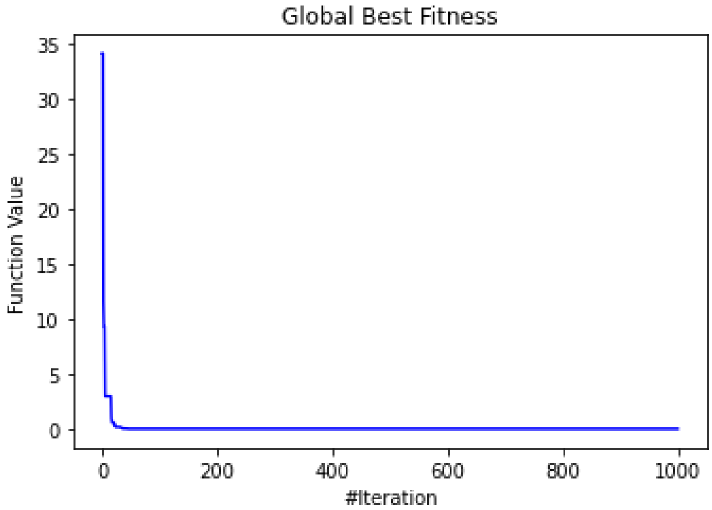

Fig 4. 4 Global Best Fitness

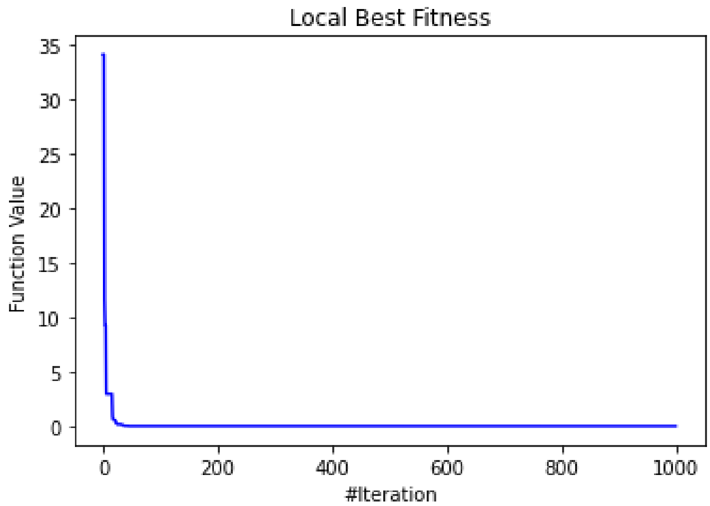

Fig 4. 5 Local Best Fitness

In Figure 4.4 and 4.5, the graph shows the trajectory of fitness between two optimum points by clusters of optimizers at the local level (inside cluster) and at the global level (dataset level).

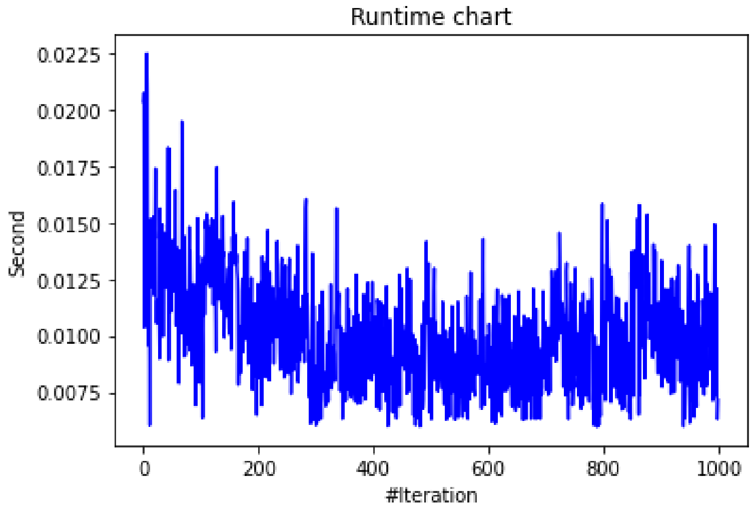

Fig 4. 6 Runtime Chart

In Figure 4.6, the graph shows the time taken by the optimizer to find an optimum point per epoch.

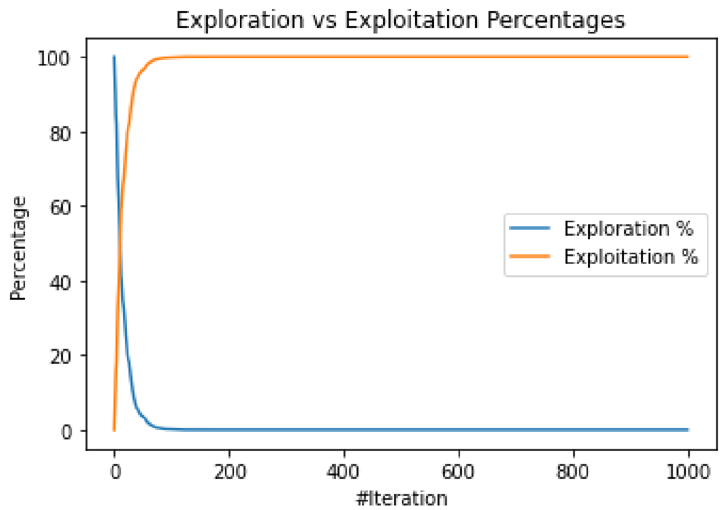

*Fig 4. 7*Exploration/Exploitation Chart

Figure 4.7 shows trajectory comparison b/w exploitation and exploration between two optimum points by clusters of optimizers.

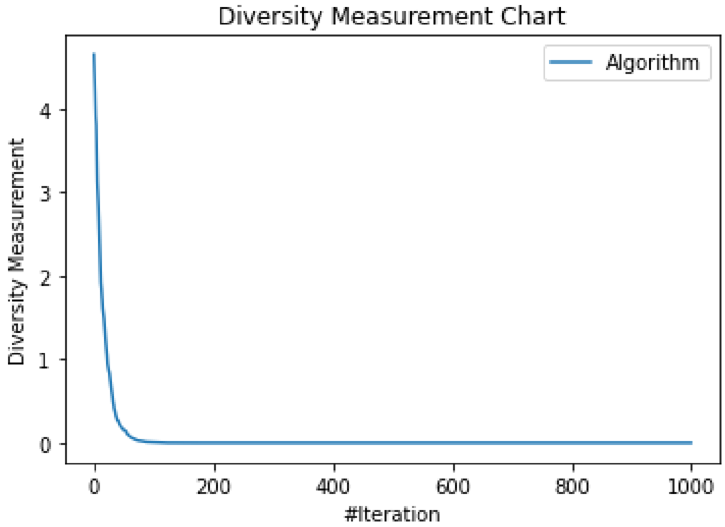

Fig 4. 8 Density Measurement Chart

```
Total Time:- 5.972784519195557
```

In Figure 4.8, the graph shows how much diverse one cluster's optimum point finding trajectory is different from the other cluster's optimum point finding trajectory.

4. Conclusion

This study focuses on the BRCNN (Backpropagation Recurrent Convolution Neural Network) module using MFO (Moth Flame Optimizer) as its Primary Optimizer rather than ADAM to detect the presence of disease on a plant and on the SVM (Support Vector Machine) to classify the sickness that the plant has. In addition, the system provides beneficial remedial measures in the form of feedback whenever they are required. The method can be utilized for the preliminary categorization of plant diseases without the requirement for time-consuming and financially burdensome plant examinations or in-depth chemical studies. According to the results of the categorization, the method has a high rate of success when it comes to determining which plant diseases exist. The use of the system has been shown to be effective at giving the user with instant feedback by drawing on a complete database of remedies that are personalized to each plant disease scenario. This has been demonstrated by a numberof studies. We are currently working on increasing the training dataset with fresh plant images in order to improve our ability to deal with a more extensive collection of plant diseases.

References

[1] Hinton, Geoffrey E, "Learning distributed representations of concepts," Proceedings of the Eighth Annual Conference of the Cognitive Science Society, vol.1, 1986.

[2] Rojas, Raul, "The backpropagation algorithm," Neural Networks, Springer, Berlin, Heidelberg, pp.149-182, 1996.

[3] LeCun, Yann, et al., "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol.1, no.4, pp.541-551, 1989.

[4] Barbedo, JGA 2016, 'A novel algorithm for semi-automatic segmentation of plant leaf disease symptoms using digital image processing', Tropical Plant Pathology, vol. 41, no. 4, pp. 210-224.

[5] Barbedo, JGA 2014, 'An automatic method to detect and measure leaf disease symptoms using digital image processing', Plant Disease, vol. 98, no. 12, pp. 1709-1716.

[6] Petrellis, N 2019, 'Plant disease diagnosis for smart phone applications with extensible set of diseases', Applied Sciences (Switzerland), vol. 9, no. 9.

[7] Kaur, S, Pandey, S & Goel, S 2018, 'Semi-automatic leaf disease detection and classification system for soybean culture', IET Image Processing, vol. 12, no. 6, pp. 1038-1048.

[8] Bai, X, Li, X, Fu, Z, Lv, X & Zhang, L 2017, 'A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images', Computers and Electronics in Agriculture, vol. 136, pp. 157-165

[9] Pethybridge, SJ & Nelson, SC 2015, 'Leaf doctor: A new portable application for quantifying plant disease severity', Plant Disease, vol. 99, no. 10, pp. 1310-1316.

[10] Rumelhart, David E., et al., "Learning representations by back-propagating errors," Nature, vol.323, no.6088, pp.533-536, 1986.

[11] S.Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm", Knowledge-Based Systems, vol.89, pp. 228–249, 2015.

[12] Cochocki, A., and R. Unbehauen, "Neural Networks for Optimization and Signal Processing," John Wiley & Sons," 1993.