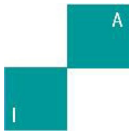
Inteligencia Artificial 26(72), 15-29
doi: 10.4114/intartif.vol26iss72pp15-29



INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

PB3C-CNN: An integrated Parallel Big Bang-Big Crunch and CNN based approach for plant leaf classification

Sukanta Ghosh[1, A], Amar Singh[1,B,*], Shakti Kumar[2, C]
[1]School of Computer Application, Lovely Professional University, Phagwara, Punjab, India
[2] Office of Director, Panipat Institute of Engineering and Technology, Panipat, Haryana, India

[A]sukantaghoshmca@hotmail.com, [B]amar.23318@lpu.co.in, [C]shaktik@gmail.com
* Corresponding Author: Amar Singh

Abstract Plant identification and classification are critical to understand, protect, and conserve biodiversity. Traditional plant classification requires years of intensive training and experience, making it difficult for others to classify plants. Plant leaf classification is a challenging issue as similar features appear in different plant species. With the development of automated image-based classification, machine learning (ML) is becoming very popular. Deep learning (DL) methods have significantly improved plant image identification and classification. In the last decade, convolutional neural networks (CNN) have entirely dominated the field of computer vision, showing outstanding feature extraction capabilities and significant identification and classification performance. The capability of CNN lies in its network. The primary strategy to continue this trend in the literature relies on further scaling networks in size. However due to increase in network size, costs increase rapidly, while performance improvements may be marginal. Hence, there is a need to optimize the CNN network to get the desired result with optimal size of machine learning model. This paper proposes a parallel big bang-big crunch (PB3C) based approach to automatically evolve the architecture of CNN. The proposed approach is validated on plant leaf classification application and compared with other existing machine learning-based approaches. From the comparision results we observed that the obtained it was found that the proposed approach was able to outperforms all the 11 existing state-of-the-art techniques.

Keywords: Image Classification, Plant Protection, Machine Learning, Deep Learning, Nature-Inspired Computing

1 Introduction

The earth is home to plants, which are indispensable resources. As part of society, they play a significant role in protecting the environment, developing medical technology, developing agriculture, and developing food [1]. Plant identification and classification are essential parts of botany, medicine, and ecology evolution studies relevant to all age groups. Key organ features, such as leaves, flowers, and seeds, were used to classify plants in traditional plant taxonomy, but molecular profiles of plants are widely used in modern classification [1]. The complexity of plant-related work, such as the identification and evaluation of plant species and diseases is increasing. It is vital to begin any plant-related project by identifying plant species, which refer to plants' physiologic and biochemical characteristics, like their colour, shape, texture, etc., that are determined by genes and the environment. Traditionally, plant species are identified by using artificial identification methods, phytochemical classifications, anatomical methods, morphological methods, and genetic methods, all of which are difficult to implement, inefficient, and have unstable accuracy [1].

As computer vision technology has developed and become more popular, image recognition technology is becoming increasingly mature and is used in various fields, including face recognition, object detection, medical imaging, etc. [2,3]. Using image processing technology to identify plant species has become a popular research topic, leading to new advances and increased accuracy. Image recognition has been developed to assist botanists with quickly identifying and distinguishing plant species and has reduced subjectivity in plant identification [4,5].

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Traditionally, image recognition algorithms used machine learning to design features and train classifiers. Plant recognition does not guarantee the accuracy, and recognizing plants in a timely manner is further hampered by slow recognition speeds.

Most studies in the past few years focused on identifying plant species using single plant organs such as leaves [6] and flowers. Research has mainly focused on the physical characteristics of leaves, including shape, texture, and venation [7]. Flowers have also been used to identify species. In recent years, advances in computer vision have made it possible for botanists to classify plants more efficiently. Compared to previous approaches that required handcrafted features, deep learning is potent at recognizing objects. An image of a plant can be processed with deep convolutional neural networks to produce a feature representation which can improve image recognition accuracy; the features are derived automatically based on end-to-end deep learning algorithms.

In recent years, researchers have begun focusing on developing automated plant classification systems, and some have used CNN to learn the basic features of a plant. NB-CNN was evolved using naïve bayes data fusion approach [8]. Several methods have been proposed for DL-based plant species recognition. For plant disease detection and diagnosis, a novel CNN was evolved using an orthogonal learning particle swarm optimization algorithm [9]. In Lee [10], the DeepPlant network was used to recognize plant leaf images, and an adaptive two-stream CNN was proposed to capture different scales of discrimination information. Convolution neural networks can identify medicinal plant species and improve identification efficiency, reduce difficulties associated with manual intervention, and prevent misclassification. In spite of its strength at learning discriminative features, CNN struggles to focus on subtle differences between objects and performs only moderately well. Limited training data and high intra-class variance among plant species made fine-grained classification challenging [11].

When solving more complex problems in the real world, more layers of neural network architecture are preferred. Much computational time, a lot of data, and fast computing resources are required for proper training, which can be challenging. Handcrafted features are employed in traditional machine learning methods. As a result of its powerful performance, Deep Learning (DL) advances traditional feature learning methods. As the number of hidden layers in-creases, the complexity of deep learning methods increases. Many hyperparameters need to be adjusted in deep learning methods. Human expertise is needed to select these network hyperparameters. CNN has a wide range of computer vision applications, making it very popular among deep networks. Nature-inspired algorithms have been used to optimize parameter selection in deep networks in the past. CNNs have been developed for the optimal selection of hyperparameters by researchers [12–17].

This article proposes a Parallel Big Bang-Big Crunch based approach to automatically evolve the optimal architecture of CNN. The major contribution of this research work is as follows:, the main objective is to propose a Parallel Big Bang Big Crunch (PB3C) system1. Proposed a soft-computing based approach to evolve the optimal architecture of CNN.2. The proposed approach is validated on plant classification application. We tested the proposed approach Mendeley plant leaf image dataset [31].

Section 2 of this paper provides the details of related works. The CNN and PB3C algorithms are discussed in Section 3. The proposed approach is provided in Section 4. The experiment work, results and discussion are given in Section 5. Section 6 concludes the paper.

2 Related Work

A lot of research has proposed different approaches and frameworks of CNN and optimization methods for CNN optimization for plant leaf image identification and classification. CNN, DBN, RNN and SAE were reviewed for their performance on plant image datasets for image classification, where CNN overpowered all other frameworks. Their advantages and disadvantages were also discussed in the article [1]. Major deep learning (DL) algorithms are reviewed with 5 datasets where the accuracy was around 95% with DenseNet whereas the LeNet was able to provide an accuracy of 70% [2]. Generative adversarial networks (GANs) [35] was first used in 2014 for image classification task. GANs provide impressive results in plant classification, disease detection, and other agricultural tasks [18]. Using state-of-the-art deep learning models, most studies found they could improve performance and classification accuracy by fine-tuning pre-trained models on plant datasets. Additionally, the results of the experiments show that a sufficient amount of labelled data of each class is available for training the models with very high accuracy.

Researchers have achieved high accuracy in limited experimental setups, such as on small datasets of a few crops and weeds. Computing speed in the recognition process is another issue that limits the deployment of herbicide spraying on fast-moving vehicles [19]. 57 tree species were detected with help of CNN combined with hand crafted features. When trained using Lagrangian Support Vector Machine (LSVM) it surpasses the results of pure CNN. Preprocessed leaves further improve the performance in tree classification using tree leaves [20].

Random 10-fold cross-validation with pure CNN has been used on plant image identification, provided a good result of 97% [21]. A novel image augmentation technique for few-shot learning has been used to classify plant image-based classification with a large dataset of images. 9% improvement has been found than the pure CNN and after augmentation there is improvement in F1-score by 12% [22]. Using the flutter framework, a plant classification mobile application was developed in which a user could upload an image to identify the plant and get details about it. Deep learning was used for the whole system, which was divided into two sections. The first section, called leaf segmentation, involves cropping the leaf part and feeding it into the classification model. A mAP score of 71% was achieved for Leaf Segmentation and a 79% accuracy for Leaf Classification [23]. The models were deployed on local computers .

Deep learning techniques power image recognition and the residual network is used as the backbone network. By applying the weakly supervised attention mechanism, the target area is assessed further to obtain a more accurate picture of the distinguishing characteristics of the target. This improves visual information processing accuracy [24]. In the PlantVillage dataset, 39 classes of plant leaves were classified using the EfficientNet deep learning architecture. Comparing the proposed architecture to the state-of-the-art deep learning architectures used in plant leaf disease detection in the literature, the proposed architecture showed comparable success. The original PlantVillage dataset and augmented versions were used in experiments. The B4 and B5 models were superior to other CNN architectures in terms of average accuracy and average precision on the original and augmented datasets. The original Dataset produces an accuracy of 99.91% and a precision of 98.42 % for the B5 model but 99.97% and 99.39%, respectively, for the B4 model [25].

Most of the researchers have used CNN for the classification of plants using leaves. But few of them tried to optimize CNN for better accuracy. Particle swarm optimization-based deep-learning approach was used for vehicle image classification. Based on Kaggle data, eight classes of vehicles are maintained and balanced using augmentation. Deep learning is used to extract features from the input images using the pre-trained GoogleNet model. A nature-based optimization algorithm, PSO, is used to optimize and reduce the features [26]. A review of relevant state-of-the-art methods shows that evolutionary optimization methods are well suited to neural architectural search and hyperparameter optimization. Reducing human experts' overhead to determine the architecture and its hyperparameters is imperative, as this is computationally expensive and may result in inefficient solutions. With the proposed MPSO approach, satisfactory results have been achieved in a limited search area that could be further enhanced to reach a more effective structure of CNNs [27].

The GWO technique is used to develop a cost-effective and automated system of classifying skin cancer based on the input pictures. This paper adopts the GWO algorithm to select hyperparameters for CNN to optimize its architecture in terms of its performance in addressing the problem of multiclass classification of skin cancer [28]. Existing metaheuristic optimization methods were used to evolve deep neural networks for various research tasks, areas, and applications, such as in training DNNs fine-tuning DNNs, networks architecture search (NAS), and DNNs hyperparameter tuning [29]. Using the SSPSO algorithm, the CNN hyperparameter optimization problem is tuned. An optimization approach improves the classification accuracy of the proposed model. The convolutional neural network layer is used to identify peripheral blood cells accurately. A sensitivity analysis using a variety of performance metrics such as Cross-Entropy, Precision, F1-Score, Accuracy, and True Positive Rate shows that the proposed technique is very accurate [30].

3 The proposed method

3.1 Preliminary

An image can be defined as two dimensional function denoted as $f(x,y)$, where x and y are known as spatial or plane coordinates and the amplitude of f at any point of coordinates (x,y) is known as the intensity or grey level of an image. In image processing, a digital image is converted into digital form, and specific operations are performed to transform it into a better image or extract useful information [32-34]. When processing a digital image, specifically plant leaf images, the following stages are required, as mentioned in Figure 1:

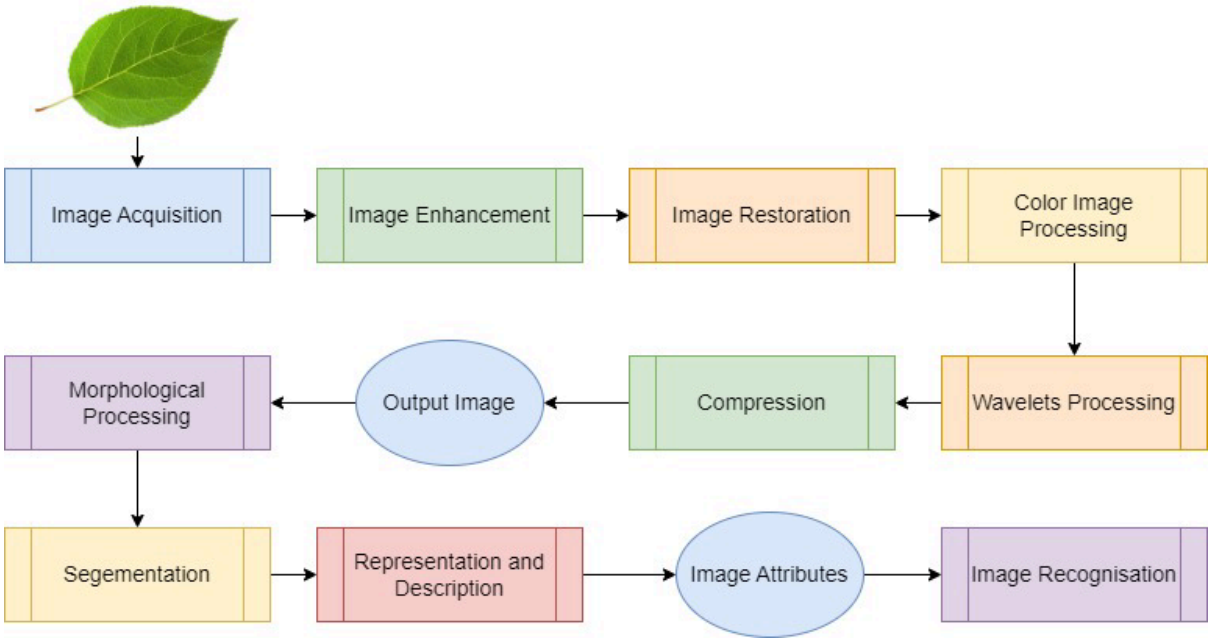


Figure 1. Processing of plant leaf image

Image Acquisition: It involves capturing digital images of plant leaves using various imaging systems such as cameras, scanners, and microscopes. The quality of the acquired images is crucial for the success of any subsequent image processing and analysis.

Image Enhancement: The process involves adjusting various parameters of an image to enhance its visual appearance, such as brightness, contrast, color balance, and sharpness. One of the primary reasons for applying image enhancement techniques to plant leaf images is to remove unwanted noise or artifacts that may have been introduced during image acquisition. These noise sources may include image blur, uneven illumination, and camera sensor noise. Applying image enhancement techniques can effectively reduce the effects of such noise sources, improving the quality of the images and making them easier to analyze.

Image Restoration: The restoration process involves mathematical modeling of the degradation process that occurred during image acquisition, followed by the estimation of the original image's parameters. This estimation is usually achieved by applying inverse filtering or regularization techniques that exploit statistical properties of the image.

Colour Image Processing: In plant leaf image classification, the first step is to acquire a high-quality color image of the plant leaf. The acquired image is then preprocessed to enhance its quality and reduce any noise or artifacts that may be present in the image. This is done by applying various image processing techniques such as filtering, normalization, and contrast enhancement.

Wavelets and Multiresolution Processing: These techniques performs processing and analyzing plant leaf images. These techniques enable the identification and extraction of features at different scales, which is essential for accurate plant species identification and classification. The use of wavelets and multiresolution processing techniques has the potential to improve the accuracy and efficiency of plant leaf image classification.

Compression: It enables the storage and transmission of large quantities of image data while reducing computational resources. Lossless compression techniques ensure that no critical features are lost during compression, while lossy compression techniques enable the compression of large quantities of data while preserving essential features.

Morphological Processing: In plant leaf image classification, morphological processing techniques are used for feature extraction, segmentation, and object recognition. We have used morphological operations such as erosion and dilation to extract leaf boundaries and veins, essential for accurate plant species identification and classification. Morphological operations were used for texture analysis, enabling the identification of different leaf textures and patterns.

Segmentation: In general, segmentation procedures divide an image into its constituent parts or objects. It enables the identification and extraction of essential features for accurate plant species identification and classification. We have used edge-based segmentation to segment the images for classification purpose.

Representation and Description: In most cases, representation and description follow the output of a segmentation stage which is usually raw pixels, whether they represent the border of a region or all the points within it. The choice of representation is only one part of transforming raw data into a computer processable format.

Object recognition: Recognition is the process that assigns a label, such as "plant leaf" to an object based on its descriptors. We have provided manual labelling of every images. Later on, the images were put in different folders to accumulate them in form of classes.

3.2 Convolution neural network

Deep neural network models using CNN are successfully employed in computer vision problems such as object detection and image classification. As CNN has been successfully applied in different fields, the research community has experimented and proposed new CNN architectures. The use of pre-trained CNN models for extracting features from an image is becoming increasingly popular. CNNs are specialized neural networks that take as input an image or video frame, use a layered approach to extract the valuable features, and assign importance to different parameters of an image so that the model can perform labelling and object recognition tasks. There are different layers in a CNN model, including convolution layers, pooling layers, fully connected layers, and softmax or sigmoid functions.

There are different hyperparameters in the convolution layer, such as the number of filters, the filter size, the strides, the activation, etc. Filters are convolved with input images to create a reduced image with the information needed to predict. Activation maps describing low-level features, such as edge structure, gradient, colour, etc. are obtained from the first convolution layer. Following the second layer, the model can learn high-level features by establishing relationships between low-level features. CNN can capture spatial and temporal relationships by applying filters to an input image. Layering reduces the size of the image matrix. The pooling layer reduces dimensionality by downsampling the input along its spatial dimensions based on the dominant features. Among the parameters of the pooling layer are the type of pooling (maximum or average), the size of the pool, the strides, and the padding. The feature map of the last convolution or pooling layer is used to input a fully connected layer. A fully connected layer or dense layer connects every input to every output based on weight. Each layer's number of layers and neurons play an important role in designing the CNN model. Softmax is used for multiclass classification. The softmax function is given in equation (1).

$$P(y = 1|x; \theta) = \frac{\exp(\theta_i^t x)}{\sum_{k-1}^{k} \exp(\theta_i^t x)} \quad (1)$$

Where, x = model input, θ = dimension of the feature vector of x, and y_i = output categories.

The features of deep learning models are automatically learned during training. As more data is added, more insights into the problem are gained. By increasing the training data, overfitting can be reduced. A neural network trained with more data is more likely to generalize well to unseen data. Various techniques such as zooming, flipping, rotating, shearing, and cropping are used to increase the number of samples. The test accuracy will decrease if the additional data is noisy and irrelevant. Plant leaf images acquired from different sources with varying illumination conditions, viewing angles, backgrounds, scales, etc. need to be preprocessed before input into the model.

3.3 Parallel Big Bang–Big crunch

The Big Bang Big Crunch theory is widely accepted among all theories of the universe's origin and evolution. A Big Bang Big Crunch (BB-BC) optimization algorithm was developed based on this theory (Erol and Eksin 2003), which we call a simple BB-BC algorithm. BB-BC is the single population based algorithm. Many applications of BB-BC optimization theory have been successfully applied. Despite this, the algorithm performed poorly for higher-dimensional problems. In some cases, the algorithm was found to be trapped in local minima. From local minima, moving towards global minima proved to be complicated.

PB3C algorithm extends the single population based BB-BC algorithm. Instead of single population of candidate solution, PB3C algorithm consists more than one populations of candidate solutions. The use of multiple populations improves the searching capability of algorithm. As shown in algorithm 1, the search first begins independently by all the populations simultaneously. To avoid local minima, the local best of individual populations interacts with the global best as the algorithm proceeds.

An energy distribution like a big-bang phase is produced by a randomly distributed finite search space and a gravity attraction like a big-crunch phase converges the solution to a global optimum using a fitness function. Using the given equation (2) the centre of the mass is calculated.

$$Sc = \sum_{i-1}^{N} \frac{1}{fi} * \frac{Si}{\sum_{i=1}^{N} \frac{1}{fi}} \quad (2)$$

where S_c = point in search space, fi = fitness value, and N = population size.

Among the population, the fittest individual is selected to carry the mass in case of a big-bang big-crunch algorithm. Similarly, for PB3C the population will be "n"; for each"n", the centre of mass will be calculated. The new population for each given n population is generated is normally distributed around the centre of individual population masses and is obtained by equation (3).

$$S_k^{new} = Sc + \sigma \quad (3)$$

S_k^{new} the i_{th} candidate solution for the individual population. σ is the standard deviation and is calculated with help of equation (4).

$$\sigma = r\alpha(Smax - Smin) =/t \quad (4)$$

where, r = random number between 0 and 1, α = parameter for reducing the size of search space, S_{min} = lower limit of candidate solution and S_{max} = upper limit of candidate solution. At the end of Big-Crunch, the global best candidate solution obtained from previous generations is used to obtain a new population and this will hold true for every population and then the global best is chosen from the local best of each population.

3.4 PB3C Optimization Algorithm [36]

The algorithm for PB3C for plant image classification is an extension of Big-Bang Big-Crunch algorithm. The algorithm of PB3C for plant image classification is as follow and given in table 1:

Table 1: Algorithm for Parallel Big Bang-Big Crunch optimization algorithm

BEGIN

Step 1: Generate a set of *N* populations for each *NC* candidate solution, chosen randomly from within the search space. Here the population refers to the hyperparameters of CNN, which need to be optimized to get better accuracy.

Step 2: For each given population, calculate the fitness function. In this case, the fitness function will be CNN.

Step 3: For each population, the local best is searched and based on this new population is created with the help of the following equation (5).

$$Xc = \frac{\sum_{i=1}^{NC} \frac{1}{fi} * xi}{\sum_{i=1}^{NC} \frac{1}{fi}} \quad (5)$$

Where, X_c = position of center of mass, x_i = position of candidate i, fi = fitness value of candidate i. The best fit candidate may be chosen for equation (4) in place of the mass centre. So, we will have *"N"* local best candidates *lbest* for each of the *"N"* populations. That is, we will have *"N"* accuracies calculated by CNN on plant leaves for *"N"* populations.

Step 4: Using the fitness values, evaluate the Global Best (*gbest*) candidate from among the *"N"* local best candidates. That is, we will find the best accuracy known as global best out of given *"N"* local best accuracies calculated by CNN on plant leaves.

Step 5: With a given probability, replace the gene of the *lbest* candidate with the gene of the *gbest* candidate.

Step 6: For each population, calculate new candidates by adding or subtracting a normal random number whose value decreases over time. This can be done with the help of equation. (6).

$$Xnew = Xc + \frac{l(rand)}{k} \quad (6)$$

Where, X_c = center of mass, l = upper limit, *rand* = normal random number and k = k_{th} iteration.

Step 7: Return to Step 2 until the stopping criteria, the best accuracy from CNN on plant leaves, has been found.

Step 8: Stop when the best accuracy has been found.

END

3.5 The proposed integrated PB3C and CNN approach

In general, CNN can solve most image recognition problems efficiently. The problem of CNN is that we have to identify its architecture manually. Manually selection of CNN architecture is a time consuming and challenging issue. The CNN architecture includes the number of layers, the number of filters in each layer, the size of the filters,

the number of neurons in the fully connected layers, the batch size, the optimizer type, and the number of epochs, etc. To automatically evolve the optimal architecture of CNN, we implemented PB3C algorithm. For the optimal architecture development purpose we evolve different hyperparameters of CNN namely convolution layers, filters, filter size, number of neuron, batch size, epochs and optimizer. Table 2 shows the hyperparameters bounds we used for CNN architecture development.

Table 2: CNN hyperparameters for optimization

Hperparameters	Range
Convolution Layers	Lower limit = 1 and Upper limit = 10
Filters	Lower limit = 1 and Upper limit = 64
Filter Size	Lower limit = 1 and Upper limit = 10
Number of Neuron	Lower limit = 32 and Upper limit = 1024
Batch Size	Lower limit = 8 and Upper limit = 512
Epochs	Lower limit = 1 and Upper limit = 25
CNN Model Optimizer	ADAM, SGD, RMSProp, Adadelta, Adagrad, Adamax

As shown in algorithm 1 initially, we use a single convolution layer, max pooling, and dropouts, followed by fully connected layers. As we are working on images and the available data is in two-dimension, so we have taken Convo2D layers for the implementation purpose. Max-pooling is generally preferred over average pooling in image classification tasks because it preserves the most salient features of the input image and is effective at capturing spatial invariance, hence max-pooling has been used for this study. The dropout rate of 0.3 has been used as we have configure the model to drop out one-third of the network during training. PB3C generates $''N''$ random initial population S. As hidden layers are added, the CNN hyperparameters increase. A variable-length chromosome represents these hyperparameters. Within the chromosome, the solution is encoded. Convolution layers, filters, filter size, neurons in FC layers, batch size, epoch, and optimizer are encoded on the genome. The chromosome length changes with the addition of a hidden layer to optimize the additional hyperparameters. The CNN model is trained and tested to determine the classification error and testing accuracy. Fitness is calculated from the test accuracy. Using the fitness function, the candidate CNN model is tested to determine how good it is at classifying plant species with the help of plant leaves. Using the PB3C search and optimization algorithm, Figure 2 illustrates how to optimize a CNN model.

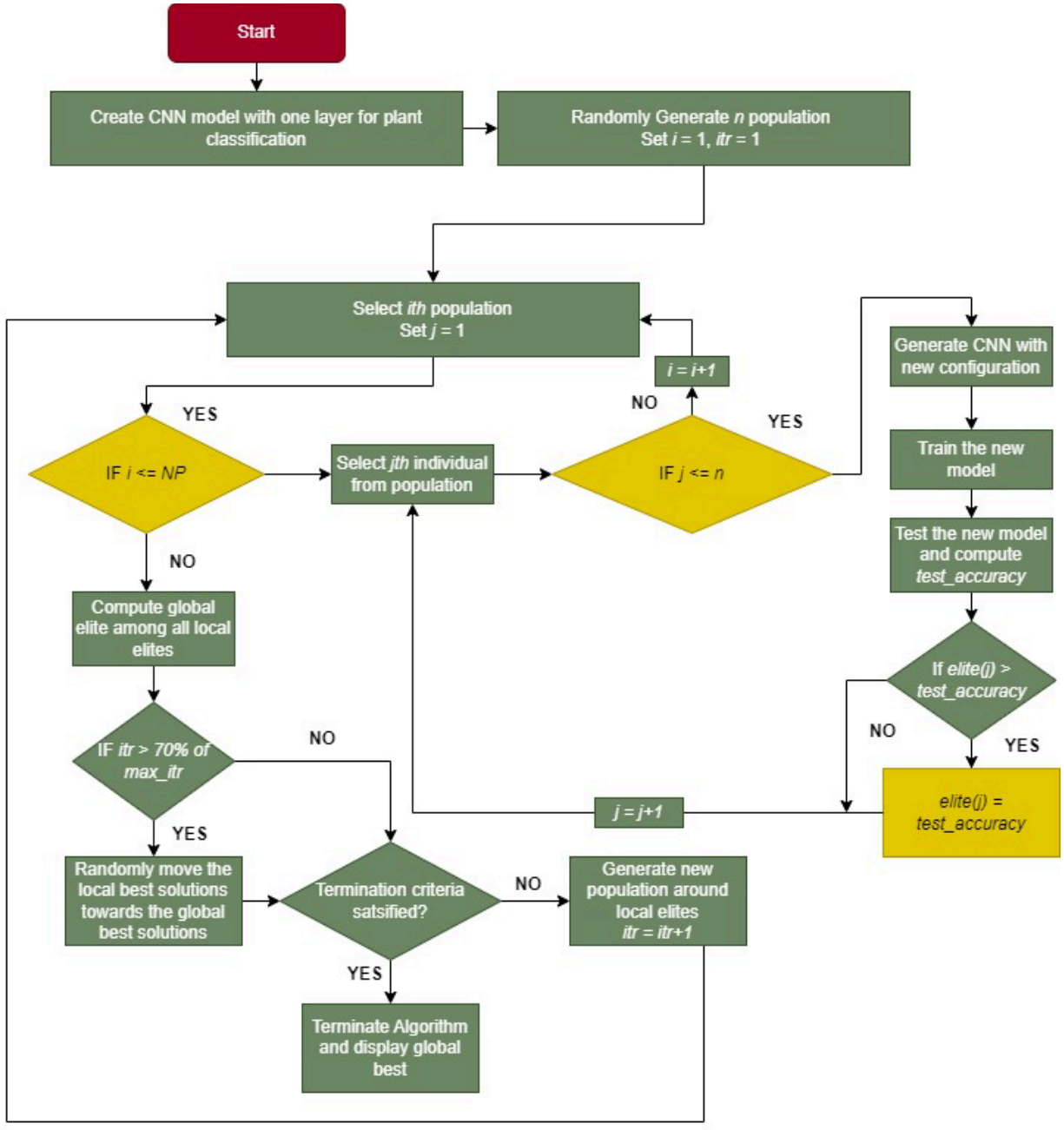


Figure 2. PB3C based approach for CNN optimization

A generation's elite is the individual with the best fitness. It is compared with the global best (the best fitness value across all generations), and the global best is updated accordingly. The same operation goes with multiple population and local best is calculated for each population, and global best is picked from all given local best.

The global best represents the optimal CNN configuration if the stopping criteria are met; otherwise, the next iteration recommends the new population based on the global best. We add a new convolution layer to the CNN model if the required number of generations is not met along with the stopping criteria. Once again, the PB3C is used to optimize the CNN. Once the stopping criteria have been satisfied, adding a new convolution layer and optimizing continues. A convolution layer cannot improve test accuracy with an accuracy of above 98% as the stopping criterion for the proposed approach. The algorithm for the above figure 2 is given in table 3:

Table 3:Algorithm for PB3C based approach for CNN optimization

Initialization: itr: Current Iteration max_itr: Maximum number of iteration

NP: No. of population $elite(j)$: Current elite, the best $test\_accuracy$: Current test accuracy
Step 1: BEGIN Step 2: Create CNN model with one layer for plant classification. Step 3: Randomly generate n population (Set i=1, $itr = 1$) Step 4: Select ith population (Set j=1) Step 5: IF $i <= NP$ THEN, Step 6: Select the jth individual from population Step 7: IF $j <= n$ THEN, Step 8: Generate CNN with new configuration Step 9: Train the new model Step 10: Test the new model and compute $test\_accuracy$ Step 11: IF $elite(j) > test\_accuracy$, THEN Step 12: $elite(j) = test\_accuracy$ Step 13: $j = j + 1$ Step 14: Go to STEP 6 Step 15: ELSE Step 16: Go to STEP 13 Step 17: Go to STEP 6 Step 18: ENDIF Step 19: ELSE Step 20: $i = i + 1$ Step 21: Go to STEP 4 Step 22: ENDIF Step 23: ELSE Step 24: Compute global elite among all local elites Step 25: IF $itr > 70\%$ of max_itr, THEN Step 26: Randomly move the local best solution towards the global best solutions Step 27: IF termination criteria is satisfied, THEN Step 28: Terminate the algorithm and display the global best Step 29: ELSE Step 30: Generate new population around local elites ($itr = itr + 1$) Step 31: ENDIF Step 32: ELSE Step 33: Go to STEP 27 Step 34: ENDIF Step 35: ENDIF Step 36: END

4 Experiment and results

4.1 Dataset

The Dataset of plant leaves has been taken from Mendeley data (Chouhan 2020). DThe Dataset was collected initially in Shri Mata Vaishno Devi University [31]. The Dataset has recently been updated on 19-January-2022 [31]. For this purpose, twelve economically and environmentally beneficial plants were selected: Mango, Arjun, Alstonia Scholaris, Guava, Bael, Jamun, Jatropha, Pongamia Pinnata, Basil, Pomegranate, Lemon, and Chinar. Leaf images of these plants in healthy and diseased conditions have been acquired and shared between two modules. We have used healthy plants for our research.

In the first step, the acquired images are categorized and labelled according to the types of plants. There are 22 subject categories ranging from 0000 to 0022 for the plants named P0 to P11. In this Dataset there are 2277 images of plant leaves. The Dataset is balanced with the equal number of images shown inthe number of images shown inFigure 3.

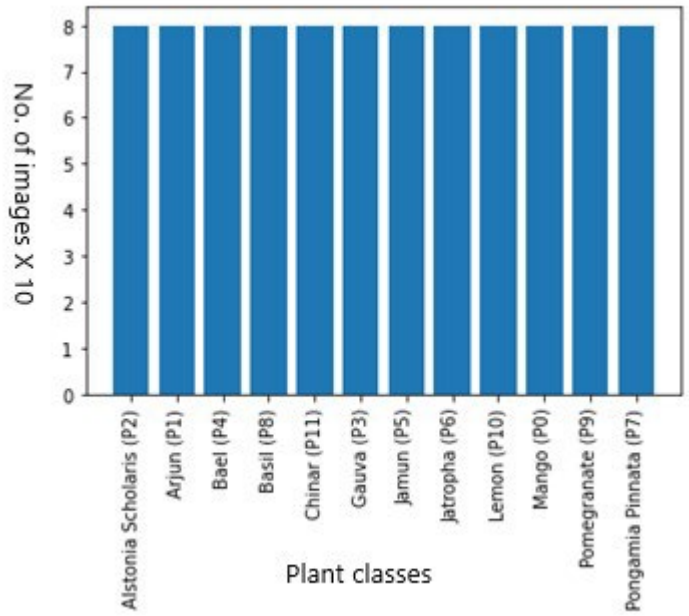


Figure 3. Balanced Dataset of plant leaf for classification

An enclosed environment is used to capture the images. Wireless communication was used during the acquisition process. A Nikon D5300 camera is used to capture all the images, with a performance timing of 0.58 seconds/frame (for shooting JPEG in single shot mode) and 0.63 seconds for RAW+JPEG. Photographs were taken with an 18-55mm lens, 24-bit depth, 2-resolution unit, ISO 1000, and no flash in .jpg format. All the images were cropped and resized to 300 x 300. Figure 4 shows the sample plant leaf images from Mendeley dataset.

Figure 4. Sample images of plant leaf Dataset

Data augmentation has been used to collect all possible cases. For data augmentation Keras flow_from_directory method has been used during the learning phase of CNN. The applied data augmentation techniques used for plant image classification are random rotation, random cropping, random flipping, random color jittering and random noise addition. This ensures that we have covered all the possible cases which might occur in the real-world. The total number of images 2277 with 207 images per class has been used for training, while 462 images, with 42 images per class, have been used for validation.

4.2 Experiment Setup

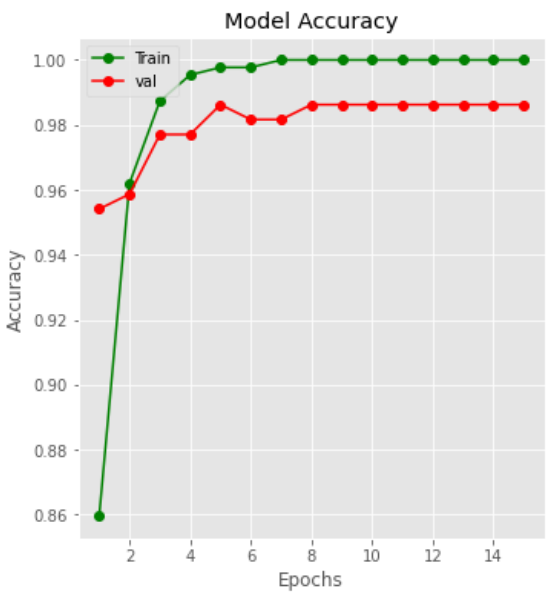
To apply and evaluate the proposed approach and compare it with other state-of-the-art machine learning approaches for plant leaf image classification, we have used Python 3.7.10. The experiment was performed on Linux 4.9.0 and on Kaggle platform for validation purpose. The hardware configuration of CPU was Intel(R) Xeon(R) CPU @ 2.30GHz with 16 CPU cores and 118.61 GB of RAM. The GPU configuration was NVIDIA K80 GPUs in kernels with 12.5X speed. Configuration of Kaggle kernals was 16 GB of RAM, high-memory CPU instances with up to 32 CPU cores and 208 GB of RAM, and GPU instances with up to 8 NVIDIA Tesla V100 GPUs and 256 GB of GPU memory.Results and Discussion

A total of 100 iterations were run to check the model's performance on the plant leaf dataset. The best accuracy of 93.20% was found in the 92th generation. For single layer conv2d layer CNN the best accuracy was 88.56%. The best accuracies for 1,2,3, and 4 conv2d layers with their generation are given in the table 4.

Table 4: Best accuracy according to number of layers

No of conv2d layers	Best Accuracy' 's Generation Number	Best Accuracy
1	72	88.56%
2	85	89.20%
3	83	91.50%
4	92	93.20%

The fitness criteria for this model were any accuracy above 90% which was not satisfied with single convo2d layer so new layers have been added to increase the accuracy and executed again the accuracy gradually increased to 89.20% with 2 conv2d layers. Further 3rd and 4th layers have increased the accuracy to 91.50% and 93.20%. Figure 5 shows the model accuracy against the number of epochs and figure 6 shows the model loss against the number of epochs.



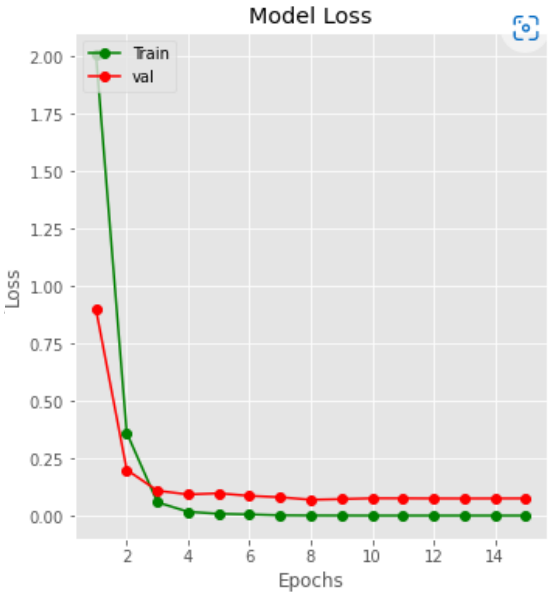


Figure 5. Model Accuracy against the number of epochs Figure 6. Model loss against number of epochs

A dataset's classification accuracy is defined as the total number of correct predictions divided by the total number of predictions. The proposed model is validated using the performance metrics of accuracy, sensitivity, specificity, precision, and f1-score. The precision of an imbalanced classification problem with more than two classes is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes and is given by the equation (7).

$$Precision = True\ Positive/True\ Positive + False\ Positive \quad (7)$$

In an imbalanced classification problem with more than two classes, recall is calculated by dividing the total number of true positives across all classes by the total number of true positives and false negatives across all classes and is given by the equation (8).

$$Recall = True\ Positive/True\ Positive + False\ Negative \quad (8)$$

The F-score is a measure that captures both accuracy and recall and is given by the equation (9). Neither precision nor recall tell the whole story alone. Alternatively, we can have excellent precision with terrible recall, or terrible precision with excellent recall. An F-score provides a way to express both concerns in a single number.

$$F1\ Score = 2 * Precision * Recall / Precision + Recall \quad (9)$$

The performance of the proposed model is given by precision value recall vale and f1-score value. The precision value of the proposed model is 0.9322, recall value is 0.9518 and the F1-score is 0.9417. The precision, recall and f1-score has been compared with other state-of-the-art classifiers and is given in table 5. Table 6 shows the comparison of proposed algorithm with the state-of-the-art transfer learning algorithms.

Table 5: Comparison of the results with state-of-the-art image classifiers

Model	Accuracy	Precision	Recall	F1-Score
SVM	82.32%	0.8242	0.8282	0.8254
KNN	75.53%	0.7555	0.7542	0.7575
Decision Tree	78.20%	0.7854	0.7888	0.7845
Random Forest	82.54%	0.8232	0.8281	0.8279
CNN	90.52%	0.9032	0.9002	0.9052
Proposed Approach	93.20%	0.9322	0.9518	0.9471

Table 6: Comparison of the results with state-of-the-art transfer learning models

Model	Accuracy	Trainable Parameters	Precision	Recall	F1-Score
VGG16	76.00%	24,588	0.7621	0.7625	0.7698
Inception V3	82.50%	512,010	0.8225	0.8262	0.8245
MobileNet V2	87.92%	15,372	0.8747	0.8745	0.8725
ResNet 50	88.00%	70,458	0.8788	0.8787	0.8764
DenseNet 121	88.00%	7,222,755	0.8888	0.8878	0.8875
Xception	88.26%	75,252,722	0.8845	0.8245	0.8278
Proposed Algorithm	93.20%	286,348	0.9322	0.9518	0.9471

According to Table 7, transfer learning models and PB3C optimized models use the same number of trainable parameters. Therefore, the proposed approach leads to an efficient neural network with fewer parameters to train. In general, parameters refer to the number of weights the model learns during training. These weights are updated by back-propagation during the training phase. CNN uses the weights to predict the future classification. In addition to the computations required to run the CNN model, adding hidden layers may lead to overfitting. An overfitted model performs poorly on the test dataset. The network must be layered appropriately and have trainable parameters that are optimal.

Table 7: Pre-trained models and proposed model parameters.

Models	Trainable Param	Non-Trainable Param	Total Param
ResNet50	70,458	256	70,714
MobileNetV2	15,372	2,257,984	2,273,356
VGG16	24,588	14,714,688	14,739,276
VGG19	175,653	20,024,384	20,200,037
InceptionV3	512,010	21,802,784	22,314,794
DenseNet121	7,222,755	86,208	7,308,963
Proposed Algorithm	286,348	640	286,988

5 Conclusion

This paper proposed an application of the Parallel Big Bang Big Crunch algorithm (PB3C) for searching for optimal Convolutional Neural Network (CNN) architecture. The proposed approach was applied to the plant image dataset. It has been observed that as the generations of PB3C increase, the classification error decreases. There are however no improvements after a certain number of generations. It has been observed that the given approach performs well with fewer number of trainable parameters. This paper has demonstrated an automated method for finding the optimal settings for CNN by tuning the hyperparameters. The performance metrics such as accuracy, precision, recall and F1-score were used for the comparison. PB3C requires planning the number of convolutional layers, specifying the number of filters, the size of filters, and the number of generations for each layer, then letting the system run. The PB3C presents the CNN settings once the simulation is completed for the generations specified. The parameters obtained for CNN demonstrate a sufficient level of classification accuracy. For future work, this approach can be more optimized by adding new hyperparameters.

Acknowledgements

The authors would like to convey our gratitude to my students who help us in data labelling process during the initial phase of study.

References

[1] J. Xiong , D. Yu , S. Liu, L. Shu, X. Wang and Z. Liu. A Review of Plant Phenotypic Image Recognition Technology Based on Deep Learning, *Electronics*, 10(1):81, 2021. https://doi.org/10.3390/electronics10010081

[2] A. Alhudhaif, A. Saeed, T. Imran, M. S. Kamran, A. Alghamdi, A.O. Aseeri, and S. Alsubai. A Particle Swarm Optimization Based Deep Learning Model for Vehicle Classification, *Computer Systems Science and Engineering*, 40(1), 223–235, 2022. https://doi.org/10.32604/csse.2022.018430

[3] A. Alotaibi. Deep Generative Adversarial Networks for Image-to-Image Translation: A Review, *Symmetry*, 12(10), 1705, 2022. https://doi.org/10.3390/sym12101705

[4] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions, *Electronics*, 9(7), 1177, 2020. https://doi.org/10.3390/electronics9071177

[5] M. Atila, M. Uçar, K. Akyol, and E. Uçar. Plant leaf disease classification using EfficientNet deep learning model, *Ecological Informatics*, 61, 101182, 2021. https://doi.org/10.1016/j.ecoinf.2020.101182

[6] K. Y. Chan, H. K. Lam, and H. Jiang. A genetic programming-based convolutional neural network for image quality evaluations, *Neural Computing and Applications,* 2022. https://doi.org/10.1007/s00521-022-07218-0

[7] P. L. Charitha, M. Mydhili, N. Khyathi, P. Pavithra, and G. Anuradha. Detection of Weed Plants Using Image Processing and Deep Learning Techniques, *Lecture Notes in Networks and Systems*, 523–532, 2021. https://doi.org/10.1007/978-3-030-84760-9_44

[8] F. C. Chen, and M. R. Jahanshahi. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion, *IEEE Transactions on Industrial Electronics*, 65(5), 4392–4400, 2018. https://doi.org/10.1109/tie.2017.2764844

[9] A. Darwish, D. Ezzat, and A. E. Hassanien. An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, *Swarm and Evolutionary Computation*, 52, 100616, 2020. https://doi.org/10.1016/j.swevo.2019.100616

[10] S. J. Lee, Chee Seng Chan, and Paolo Remagnino. Multi-Organ Plant Classification Based on Convolutional and Recurrent Neural Networks, *IEEE transactions on image processing*, 27(9), 4287–4301, 2018. https://doi.org/10.1109/tip.2018.2836321.

[11] O. K. Erol, and I. Eksin. A new optimization method: Big Bang–Big Crunch, *Advances in Engineering Software*, 37(2), 106–111, 2006. https://doi.org/10.1016/j.advengsoft.2005.04.005

[12] P. Ghamisi, Y. Chen, and X. X. Zhu. A Self-Improving Convolution Neural Network for the Classification of Hyperspectral Data, *IEEE Geoscience and Remote Sensing Letters*, 13(10), 1537–1541, 2016. https://doi.org/10.1109/lgrs.2016.2595108

[13] A. Ghosh and P. Roy. An automated model for leaf image-based plant recognition: an optimal feature-based machine learning approach, *Innovations in Systems and Software Engineering,* 2022. https://doi.org/10.1007/s11334-022-00440-y

[14] J. Hu, Z. Chen, M. Yang, R. Zhang and Y. Cui. A Multiscale Fusion Convolutional Neural Network for Plant Leaf Recognition, *IEEE Signal Processing Letters*, 25(6), 853–857, 2018. https://doi.org/10.1109/lsp.2018.2809688

[15] X. Jin, J. Che, and Y. Chen. Weed Identification Using Deep Learning and Image Processing in Vegetable Plantation, *IEEE Access*, 9, 10940–10950, 2021. https://doi.org/10.1109/access.2021.3050296

[16] P. S. Kanda, K. Xia, and O. H. Sanusi. A Deep Learning-Based Recognition Technique for Plant Leaf Classification, *IEEE Access*, 9, 162590–162613, 2021. https://doi.org/10.1109/access.2021.3131726

[17] R. Kumar, S. Joshi, and A. Dwivedi. CNN-SSPSO: A Hybrid and Optimized CNN Approach for Peripheral Blood Cell Image Recognition and Classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 35(05), 2157004, 2020. https://doi.org/10.1142/s0218001421570044

[18] S. H. Lee, C. S. Chan, S. J. Mayo and P. Remagnino. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, 71, 1–13, 2017. https://doi.org/10.1016/j.patcog.2017.05.015

[19] V. Partel, S. Charan Kakarla, and Y. Ampatzidis. Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence, *Computers and Electronics in Agriculture*, 157, 339–350, 2019, https://doi.org/10.1016/j.compag.2018.12.048.

[20] S. Liu, Q. Shi, and L. Zhang. Few-Shot Hyperspectral Image Classification With Unknown Classes Using Multitask Deep Learning, *IEEE Transactions on Geoscience and Remote Sensing*, 59(6), 5085–5102, 2021. https://doi.org/10.1109/tgrs.2020.3018879

[21] Y. Liu, Y. Li, Y. Zhao, and X. Na. Image Classification and Recognition of Medicinal Plants Based on Convolutional Neural Network, *2021 IEEE 21st International Conference on Communication Technology (ICCT),* 2021. https://doi.org/10.1109/icct52962.2021.9658028

[22] R. Mohakud, and R. Dash. Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection, *Journal of King Saud University - Computer and Information Sciences,* 2021. https://doi.org/10.1016/j.jksuci.2021.05.012

[23] Y. M. Oo, and N. C. Htun. Plant Leaf Disease Detection and Classification using Image Processing, *International Journal of Research and Engineering*, 5(9), 516–523, 2018. https://doi.org/10.21276/ijre.2018.5.9.4

[24] B. M. Quach, V. C. Dinh, N. Pham, D. Huynh and B. T. Nguyen. Leaf recognition using convolutional neural networks based features, *Multimedia Tools and Applications,* 2022. https://doi.org/10.1007/s11042-022-13199-y

[25] P. Singh, S. Chaudhury and B. K. Panigrahi. Hybrid MPSO-CNN: Multi-level Particle Swarm optimized hyperparameters of Convolutional Neural Network, *Swarm and Evolutionary Computation*, 63, 100863, 2021. https://doi.org/10.1016/j.swevo.2021.100863

[26] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder. Automated plant species identification—Trends and future directions, *PLOS Computational Biology*, 14(4), e1005993, 2018. https://doi.org/10.1371/journal.pcbi.1005993

[27] P. Wang, T. Niu, Y. Mao, Z. Zhang, B. Liu and D. He. Identification of Apple Leaf Diseases by Improved Deep Convolutional Neural Networks With an Attention Mechanism, *Frontiers in Plant Science*, 12, 2021. https://doi.org/10.3389/fpls.2021.723294

[28] R. Mohakud and R. Dash. Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection, *Journal of King Saud University - Computer and Information Sciences*, 2021. https://doi.org/10.1016/j.jksuci.2021.05.012.

[29] J. Xiong, D. Yu, S. Liu, L. Shu, X. Wang and Z. Liu. A Review of Plant Phenotypic Image Recognition Technology Based on Deep Learning, *Electronics*, 10(1), 81, 2021. https://doi.org/10.3390/electronics10010081

[30] C. L. Zhou, L. M. Ge, Y. B. Guo, D. M. Zhou and Y. P. Cun. A comprehensive comparison on current deep learning approaches for plant image classification, *Journal of Physics: Conference Series*, 1873(1), 012002, 2021. https://doi.org/10.1088/1742-6596/1873/1/012002

[31] S. S. Chouhan. A Database of Leaf Images: Practice towards Plant Conservation with Plant Pathology [Dataset], 2020. https://data.mendeley.com/datasets/hb74ynkjcn/4

[32] S. Ghosh and A. Singh. The scope of Artificial Intelligence in mankind: A detailed review, *Journal of Physics: Conference Series*, 1531 (1), 012045, 2020. https://dx.doi.org/10.1088/1742-6596/1531/1/012045

[33] S. Ghosh, A. Singh, Kavita, N.Z. Jhanjhi, M. Masud and S. Aljahdali, SVM and KNN Based CNN Architectures for Plant Classification, *CMC-Computer Materials & Continua*, 71 (3), 4257-4274, 2022.

[34] S. Ghosh and A. Singh. The Analysis of Plants Image Classification Based on Machine Learning Approaches, *In Emergent Converging Technologies and Biomedical System,* 133-148, Springer, Singapore, 2022.

[35] C. Seward, T. Unterthiner, U. Bergmann, Nikolay Jetchev, and S. Hochreiter. First Order Generative Adversarial Networks, *arXiv (Cornell University)*, Feb. 2018.

[36] S. Kumar, A. Singh, S. Walia. Parallel Big Bang–Big Crunch Global Optimization Algorithm: Performance and its Applications to routing in WMNs, *Wireless Pers Commun, Springer,* 100, 1601–1618, 2018. https://doi.org/10.1007/s11277-018-5656-y