International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

An Improved Particle Swarm Optimization integrated Dilation Convolutional Neural Network for Automatic Soybean Leaf Disease Identification

Bhabanisankar Jena[1], Ashanta Ranjan Routray[2], Janmenjoy Nayak[3]

[1] *Department of Computer Science, Fakir Mohan University, Balasore, Odisha, India*
[2] *Department of Computer Science, Fakir Mohan University, Balasore, Odisha, India*
[3] *Department of Computer Science, MSCB University, Baripada, Odisha, India*

E-mail address: bhabanisankarjena94@gmail.com, ashan2r@gmail.com, mailforjnayak@gmail.com

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Leaf disease is a prominent and destructive ailment that affects plants. Timely identification and early detection are crucial for enhancing the future probability of leaf diseases that impact plants. The investigation of soybean plant leaf disease detection has gained importance owing to its major impact on soybean growth, leading to decreased productivity and quality. The traditional method of identifying soybean leaf diseases mostly relies on agricultural specialists, resulting in a significant amount of time being utilized. Deep Learning (DL) models are promising techniques to identify soybean leaf disease detection. However, various ongoing investigations are going on to achieve an effective model with efficient practical application. To address this problem, this study proposes the use of a hybrid, smart and intelligent model based on dilation Convolutional Neural Network (CNN) to identify diseases of soybean leaves. Selecting and designing the ideal model structure is still a difficult task, even though DL networks demonstrate remarkable efficacy. The accuracy of plant disease detection based on leaf analysis may be improved by fine-tuning the values of the hyper-parameter of dilation CNN. The proposed framework has been trained using a dataset of 1620 soybean leaf images that have been divided into six different diseased groups. The Velocity Pausing Particle Swarm Optimization (VP_PSO), a well-studied metaheuristic technique, is employed to optimize the hyper-parameters of the dilation CNN. This optimization aims to improve the effectiveness of the dilation CNN in accurately recognizing diseases present in soybean plant leaves. The suggested hybrid model performs better than other standard hybrid models such as classical CNN, VGG16, MobileNetV2, ResNet101, dilation CNN and PSO_Dilation_CNN. As per the experimental research, the suggested VP_PSO _Dilation CNN model has a detection accuracy of 95.32%.

Keywords: CNN, Dilation CNN, Hyper-parameters, Optimization, PSO, VP_PSO, DL, Leaf Disease

1. Introduction

In agricultural farming, leaf diseases are predominant challenges faced by the plants in producing a large number of agricultural products. Diseases that appear in plants have a detrimental effect on crop productivity. These leaf diseases must be identified in time. If these diseases cannot be detected in due time, there will be an adverse effect on crop production [1]. Early detection of leaf diseases plays a key role in the crop production management process [2]. This popular approach is implemented in identifying diseases in various crop fields like tomato [3], potato [4], rice [5], apple [6], soybean [7] etc. However, soybean has become a remarkably popular oil-producing crop and one of the top five most significant food crops in the world [8]. It has become the main source of edible oil now and also it produces almost a quarter i.e. 25% of the total edible oil produced worldwide [9]. The nutritional characteristics of soybeans can help people prevent themselves from the high risk of cardiovascular disease and diabetes to some extent. The meal produced with soybean oil is highly composed of proteins making it a popular choice for human consumption. In addition to being consumed by humans, it is also often employed in aquaculture production systems [10]. It is also extensively used in many industries for the preparation of detergents, inks, lubricants and paints and is also used in the field of bio-diesel production [11]. To maintain human requirements in the largest growing population country India along with the requirements in industrial production, the demand for the edible oil prepared from soybean will undoubtedly be very high as compared to the current

E-mail:author's email

production. Overcoming the challenges of production of soybean in future will require the caring of soybean plants to be protected from diseases with timely identification of diseases, which can be done through technological computations.

Many agricultural specialists still rely on naked-eye observations as a fundamental method for detecting leaf diseases in plants. Nevertheless, conventional procedures have some inherent weaknesses. For instance, it is time-consuming for big farms to manually detect diseases and the cost is significantly high due to the substantial expenses of agricultural experts specialized in high-frequency observation. As a result, several automated strategies have been implemented to speed up the human observation procedure for identifying leaf diseases. These techniques are very practical and important since they aim to identify diseases by analyzing leaf images. In the past, these automated detection procedures were carried out by carefully creating a specific classifier to categorize the sample leaf images into healthy and unhealthy images. Machine learning (ML) the subfield of Artificial Intelligence (AI) has been utilized to identify plant leaf diseases at later stages of technological development [12]. Plant leaf disease detection in ML makes use of several techniques such as K-Nearest Neighbor (KNN) [13], Random Forest (RF)[14], Support Vector Machine (SVM)[15], Decision Tree (DT)[16], etc. On the other hand, when applied to big and diverse leaf image datasets, these ML methods rely on hand-crafted features for feature extraction using methods like Principal Component Analysis (PCA), Gabor transform and Histogram of oriented gradient [17]. As a result, these methods could fail to identify complex features or patterns. Therefore, these methods are not particularly reliable in big and diverse datasets.

In contrast, another developing branch of AI popularly known as DL particularly the CNN has been widely utilized in the domain of leaf disease detection due to its strong ability to automatically extract and learn features from raw leaf images [18]. These networks are derived from the visual brain of humans and have been designed to identify complicated patterns observed in plant leaves. This CNN framework is widely implemented by various researchers in different agricultural fields to detect leaf diseases which lead to enhanced crop productivity. In the field of plant leaf disease detection, CNN has illustrated substantial potency through an end-to-end detection process [19]. However, CNN demands high computation resources for training and testing the model and also requires a huge amount of memory space due to the massive amount of parameters used in its fully connected layer which is approximately 80% of the entire neural network. Due to enormous parameters, sometimes it may face the problem of overfitting [20]. Many researchers have set the value of dropout in its fully connected layer to minimize the parameter size to create a robust model, which can avoid the problem of overfitting [21]. It often encounters the problem of analyzing the multi-scale feature information from the leaves required for identifying the symptoms of disease as this framework is fixed with certain receptive fields. But, for this, manual optimization of the dropout parameter is required which heavily depends upon neural network experts. Thus, some specific techniques are often implemented to enhance the disease detection capability of the CNN model by reducing the high computation time, which refers to the increment of the receptive field and also the expansion of the image to its original size. However, some feature information may be lost and that small feature information cannot be regenerated. To address this problem, enhancement of the receptive field is required without dropping the spatial resolution. This can be treated as the fundamental limitation of CNN due to its fixed size of receptive field. Dilated CNN can enhance the receptive field without minimizing the spatial resolution with the introduction of gaps in between the element of the kernel [22] and also it attains good identification results [23]. Therefore, superior accuracy and robustness may be acquired by implementing this in comparison to the basic CNN approach.

An appropriately configured self-learning environment is crucial for the dilation CNN model. This environmental configuration is extensively associated with the values of the hyper-parameter. Proper selection of certain hyper-parameters, such as dilation rate, batch size, learning rate, network depth and number of epochs significantly affects the overall performance of the dilation CNN model. Improper selection of hyper-parameter values may lead to a reduction in the performance of the dilation CNN model. Due to inappropriate selection of hyper-parameter values loss function may not be adequately minimized which leads to incorrect results [24]. In general, the selection of hyper-parameter values involves a trial-and-error approach due to the absence of particular mathematical equations. This procedure needs the expertise of neural network professionals and requires a significant amount of effort and time. The manual procedure of trial-and-error primarily relies on experience with the DL model and the independent selection of hyper-parameter values by the network designer. Therefore, manually selecting the ideal values of hyper-parameters is an extremely challenging operation [25]. Numerous researches have been conducted to enhance the performance of DL models by optimizing their hyper-parameter values using grid search and random search methods to determine suitable hyper-parameter values, while grid search and random search are considered superior alternatives to manually selecting hyper-parameter values. Both methods are inadequate when dealing with high-dimensional search spaces and also require significant computational time [26]. Currently, many researchers consider the task of determining the right values of hyper-parameters as an optimization issue [27].

Optimization strategies are automated approaches used to improve the performance of a model by adjusting the values of hyper-parameters. These techniques have consistently outperformed the results produced by manual human professionals. Swarm Intelligence is an effective alternative to the automated strategy for implementing the complexities of dilation CNN to determine the optimal values of hyper-parameters [28]. This approach is inspired by the collective social behaviour of animals, birds, fish etc. available in natural environments. Many researchers have created various hybridized categories of CNN through different optimization algorithms used for hyper-parameter selection [29-31]. Among different optimization algorithms, Particle Swarm Optimization (PSO) algorithm was introduced by Kennedy and Eberhert [32], which is particularly inspired by the social behaviour of flocking birds. Other nature-inspired algorithms like Genetic Algorithm (GA) [33], Ant Colony Optimization (ACO) [34], Firefly Algorithm (FA)[35], Cuckoo Search Optimization (CSO)[36] etc. are working as similarly as the operation of PSO. However, PSO is the most preferred and popular optimization algorithm amongst researchers because it contains smaller parameters with simple computation and formulation. Currently, PSO is applied in various emerging and trending fields of neural networks, which creates a source of inspiration for this study. This optimization encounters challenges like premature convergence and insufficient local optimization. PSO is hindered by early convergence, heightened update velocity, memory requirements and suboptimal solutions. This is observed due to insufficient exploration of the complex search space. Furthermore, PSOs have demonstrated satisfactory outcomes on a particular class of optimization problems, but when this approach deals with a variety of challenges, their effectiveness has declined. The introduction of the VP_PSO optimization is suggested to address the limitations associated with this optimization technique. The VP_PSO incorporates the capability to stop and modify the particle velocity, facilitating more flexible control over the balance between exploration and exploitation, which precisely adjusted parameters for models used in disease diagnosis. This enhances the resilience and efficiency of VP_PSO making it a superior option for agricultural diagnostics and plant leaf disease monitoring.

The main contribution of this paper is as follows:

- This research primarily focuses on improving the accuracy of detecting soybean leaf diseases and classifying them using a swarm intelligence optimization approach based on dilation CNN architecture.
- The primary goal of this study is to enhance the process of selecting optimal hyper-parameter values for the dilatation CNN architecture. This will be achieved by integrating the VP_PSO approach with this dilation CNN. The basic concept is to develop a complex and intelligent dilation CNN architecture that can accurately detect and classify soybean leaf diseases.
- The efficiency and performance of the suggested VP_PSO_dilation CNN model are compared to various variants of the CNN framework. The results demonstrate its superiority, achieving an accuracy percentage of 95.32%.

The subsequent sections of the article are structured as sections 2, 3, 4, 5, 6 and 7. Section 2 describes a comprehensive evaluation of previous methodologies that are relevant to the soybean plant leaf disease detection field. Section 3 presents a detailed description of the suggested technique with other essential methodologies. Section 4 describes the detailed architecture and algorithm used in the proposed methodology. Section 5 explains the detailed information on the dataset and the specific environment in which the implementation took place. Section 6 discusses a thorough explanation of the findings obtained from the proposed model. The analysis also includes a comparison with several versions of the CNN model. Finally, this investigation is finalized in section 7 referred to as conclusion and future scope.

2. RELATED WORK

Currently, researchers are placing significant emphasis on using ML-based neural network architecture to detect soybean leaf diseases. The implementation of the DL strategy for leaf disease classification has significantly improved the accuracy of the model compared to a traditional ML model. This section discusses the implementation of DL-based classification models for the detection of diseases in soybean plant leaves as shown by many research. Karlekar and Seal [37] have presented a dual-component model. The first step is removing the complex background from the leaf image to obtain the appropriate leaf portion. In the second section, the authors have presented a CNN model called SoyNet. This model is mainly designed for detecting diseases in soybean leaves using segmented leaf pictures obtained from the first section. The SoyNet architecture is primarily composed of six convolution layers with filters, six MaxPooling layers with filters and two fully linked layers. The researchers have successfully used it in the dataset "Image Database for Plant Disease Symptoms" and obtained an accuracy rate of 98.14%.

Additionally, many pre-trained models are shown to accurately detect soybean leaf diseases. Noah Bevers et al. [38] effectively implemented a pre-trained DL model called DenseNet201 combined with an automated classifier to classify eight various types of diseased leaves found in soybean plants. In their study, the authors utilized transfer learning, data augmentation and data engineering techniques to enhance the training of the model. The researchers used a dataset consisting of about

9500 synthetic soybean leaf pictures, which is stored in the "Auburn Soybean Disease Image Dataset". They set the learning rate to a fixed value of 0.0001 and employed stochastic gradient descent as the optimizer running it for 100 epochs. This suggested model was compared to various pre-trained techniques such as ResNet50, EfficientNetB0, Xception and VGG16 and achieved the highest accuracy of 96.8%.In conjunction with using pre-trained models, several researchers have implemented the conventional CNN architecture with specific predetermined values for hyper-parameters. Pan et al. [39] conducted an experiment using a "two-stage feature aggregation network framework" to identify soybean leaf diseases. In their study, the authors have used the feature representation capability in conjunction with the feature information derived from the convolution layers employed in the model. A feature fusion framework is created with dilated convolutions to prevent loss of information during feature extraction. This framework expands the receptive field. The collection consists of approximately 9648 soybean leaf images representing eight different categories of disease. These images were acquired from the Auburn soybean leaf disease picture library. The authors have merged the suggested model with the InceptionC model to achieve a high level of performance accuracy of 98.18%.

Qinghai Wu et al. [40] have developed an enhanced ConvNext model for accurately detecting soybean leaf diseases while also improving the model's resilience. To extract features, the authors have included a unique 'attention module' at various levels of depth. The LeakyReLU activation function is used to assist the attention module in producing feature maps. The soybean dataset has been obtained from Jilin College of Agricultural Science and Technology in Jilin Province, China. It has been employed to conduct experiments on the ConvNext model, which has been combined with the purpose of the Swin Transformer. The combination of both strategies results in an enhanced ConvNext model. This model utilizes convolution operations and a stride value of four to extract features from soybean leaf pictures. In this study, the scientists have included 96 distinct channels into the feature map and achieved a greater accuracy of 85.42% compared to the usual ConvNext, Swin Transformer, ResNet50, MobileNetV3, SqueezeNet and ShuffleNetV2 models. Miao Yu et al. [41] have employed an enhanced version of a residual neural network that incorporates a residual attention layer using an attention mechanism to address the issue of high computational requirements in identifying soybean leaf diseases. The Ostu Algorithm has been employed for the segmentation of soybean leaves. During the network training phase, the Adam optimization algorithm is implemented to determine different learning rates for different training parameters by manipulating first-order and second-order partial derivatives. The suggested model achieved a testing accuracy of 96.50%.

A comprehensive summary of the existing literature for the identification of soybean leaf diseases is described in Table 1. This table summarizes many investigations along with the dataset used, methodologies employed and the advantages and disadvantages of each strategy.

Table 1. Summary of existing studies

Year	Proposed Model	Dataset	Performance Accuracy	Advantages	Disadvantages	Ref.
2023	Two-stage Feature Aggregation Network Framework	Auburn Soybean Disease Image Dataset	98.18%	Less an amount of loss in feature information due to the use of dilated convolution	Setting the appropriate values for the size of the feature map is too difficult while designing the InceptionC model	[39]
2023	Improved ConvNext model	Soybean plantations at Jilin College of Agricultural Science and Technology, Jilin Province, China	85.42%	Model inactivity is improved when the input value is negative	Less accuracy in identifying diseases having high similarity-based features	[40]
2023	YOLOv5	soybean farm, Horticulture Research Center (Southern Illinois University)	95%	Hyper-parameter optimization is performed by Genetic Algorithm	Premature convergence to local optima in complex and high search space	[46]

2022	DenseNet 201	Auburn Soybean Disease Image Dataset	96.80%	The automated classifier is integrated with CNN to classify soybean leaf images	Due to the smaller learning rate computation time is too high. Small Dataset is used	[38]
2022	Improved Residual Neural Network	Synthetic	96.50%	The detection process is a speed	Works efficiently only in less number of dataset	[41]
2022	Improved Residual Attention Neural Network	Synthetic	98.49%	Produces a fast and accurate system	Difficulty in the selection of hyper-parameter values	[42]
2021	CNN and its variants	Multi-blade real crop image	99.04%	Improved technology for segmentation is used	For achieving superior accuracy proposed model can be improved with hyper-parameter fine-tuning	[43]
2021	NemaNet	NemaDataset	96.76%	Extraction of deep features in soybean leaf is improved	Hyper-parameter setting is too difficult	[44]
2020	Soy Net	Image Database of Plant Disease Symptoms	98.14%	Good accuracy is achieved due to the enhancement of diversified pooling operation	Selection of hyper-parameter values is manual	[37]
2019	Deep CNN	different sources of online databases such as plant village dataset, forestry image dataset, imagenet dataset	89.84%	Better recognition of important features	Lower accuracy and also values of Hyper-parameter setting is too difficult	[45]

Table 1 shows that there has been a significant development of plant leaf disease segmentation and classification techniques in recent years. These approaches have proven useful in detecting diseases that affect plant leaves. Still, some drawbacks have been recognized for each of these methods. The CNN model and its many variants are often utilized. Hyper-parameter adjustment is essential for optimizing the accuracy of the basic CNN. Hyper-parameter Tuning encompasses the use of data augmentation and regularization approaches to enhance the model's resilience and ability to generalize, hence increasing its reliability across various situations. Given this reason, significant focus is placed on creating efficient hybrid architecture of VP_PSO_dilation CNN. This involves automatically selecting hyper-parameters by traversing a wide search space and achieving fast convergence speed. The primary objective of this study is to create a highly efficient model that requires little computing resources and time.

3. METHODOLOGIES

This section presents a comprehensive explanation of the complex structure of the dilation CNN architecture. Subsequently, an analytical description of the VP_PSO has been described.

3.1 Dilation_CNN

Convolutional kernels are essential elements of CNN, which have been the predominant methods for most computer vision problems in recent years [47]. Their strength is derived from their capacity to hierarchically represent spatial characteristics across input areas, known as Receptive Fields (RFs), by layering many convolutional layers into deep structures [48]. In modern times, when it comes to constructing CNN architectures, it is typical to employ wide receptive fields (RFs) to gain better performance. Dilated Convolutional Kernels (DCKs) have been a popular option due to their simplicity and efficacy [49, 50]. DCKs, unlike traditional equivalents, can significantly increase RFs without expanding the size of the kernel. CNN models with dilated kernels have shown remarkable performance in essential tasks including semantic segmentation, object identification using multi-model and object classifications [50, 51]. In addition, DCKs outperform in some specialized applications showing substantial improvement in performance with the use of dilated convolutional kernels. The dilation CNN (Figure 1) has six different categories of layers, each with unique capabilities, as shown below.

3.1.1 Input Layer in Dilation CNN

The input layer receives the unprocessed data, usually in the form of an image and transfers it to the first dilation convolutional layer. The input data may be expressed as a tensor, which is a multi-dimensional array. For instance, in the case of an image, the input tensor (T) has dimensions that correspond to its height (h), width (w) and the number of channels (c).Equation 1 represents the input layer of dilation CNN.

$$dl\_cnn_{input} = T, T \in Image^{h*w*c} \quad (1)$$

3.1.2 Dilation Convolution Layer

The output feature map $dl\_cnn_{conv\_output}$ is created by the convolution of the input feature map dl_cnn_{input} with a filter $filter_{dl\_cnn}$ in a conventional convolution operation. A dilated convolution involves the introduction of a dilation factor $(d\_rate_{dl\_cnn})$, which determines the spacing between the components of the filter. Equation (2) and (3) shows the dilation convolution operation (Figure 2).

$$dl\_cnn_{conv(i,j,c)} = \left(\sum_{d=0}^{c} \sum_{p=0}^{fw-1} \sum_{q=0}^{fh-1} filter_{dl\_cnn\,(p,q,d,c)} * T_{(i+p*dlr,\, j+q*dlr,\, d)}\right) (2)$$

$$dl\_cnn_{conv\_output} = \sigma\left(dl\_cnn_{conv(i,j,c)}\right) (3)$$

$Where, dlr = Dilation\ factor\ and\ \sigma = Activation\ function$

3.1.3 Pooling Layer

Pooling layers in dilation CNN are used to decrease the spatial dimensions (height and breadth) of the input volume, resulting in improved computational efficiency and a reduction in the number of parameters, which helps to prevent overfitting. Max pooling and average pooling are the two most frequently performed forms of pooling procedures. Equation (4) represents the pooling procedure.

$$dl\_cnn_{pooling(i,j,c)} = \frac{1}{p_h * p_w} * \sum_{p=0}^{p_{h-1}} \sum_{q=0}^{p_{w-1}} T_{(i+p*s_h,\, j+q*s_w,\, d)} \quad (4)$$

$Where, p_h\ and\ p_w are\ height\ and\ width\ of\ pooling\ window, s_h\ and\ s_w\ are\ the\ stride\ along\ the\ height\ and\ width$

3.1.4 Activation Layer

Activation layers play a vital role in dilation CNN by introducing non-linearity into the model. This allows the model to learn and reflect more complex patterns in the input. The mathematical formulations of several activation functions, such as ReLU, Sigmoid, Tanh and Leaky ReLU, determine how these functions convert every component of the input tensor to generate the output tensor. Equation (5) shows the activation layer operation.

$$dl\_cnn_{activation} = \sigma(.) \ (5)$$

3.1.5 Fully Connected Layer

The fully connected layers of dilation CNN have a vital function in acquiring knowledge about overall patterns and connections within the data. They facilitate the connection of all the characteristics from the preceding levels to the output layer allowing the network to generate predictions based on these acquired representations. CNN offers great performance on tasks like image classification, object identification and segmentation by integrating local information acquired via a dilation convolutional layer and pooling layers with global information acquired through fully connected layers. Fully connected layers can be expressed in equation (6).

$$dl\_cnn_{fcl} = \sigma\left(dl\_cnn_{pooling(i,j,c)}{}^{flattened}\right) (6)$$

3.1.6 Output Layer

The configuration of the output layer in a dilated CNN is dependent upon the specific task that the network is designed to perform. The output layer of a neural network can produce meaningful predictions or segmentations by selecting a suitable activation function and layer design, which utilizes the learnt characteristics from the input data.

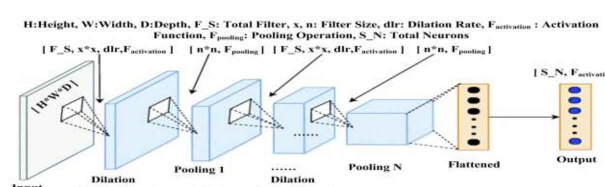


Figure 1. General architecture of dilation CNN

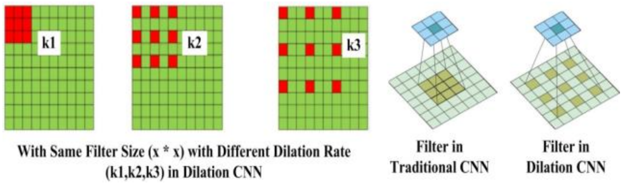


Figure 2. Dilation convolution operations

4. PROPOSED SYSTEM

This section presents a comprehensive description of the VP_PSO approach and the suggested framework for detecting plant leaf diseases.

4.1 Velocity Pause Approach with Particle Swarm Optimization

The VP_PSO method includes the idea of velocity pausing, which enables particles to keep away from changing their velocity throughout each iteration [52]. Furthermore, a particle can retain its velocity from the previous iteration. This principle allows particles to have the ability to move at three different speeds: a slower speed, a quicker speed and a constant speed. In contrast to the typical PSO method, which restricts particles from moving at either a faster or slower velocity, this approach allows for more flexibility in particle movement. Velocity pausing introduces a significant advantage by offering an additional choice for movement, such as maintaining a consistent pace. This option facilitates the preservation of a harmonious equilibrium between exploration and exploitation, hence mitigating the issue of premature convergence often seen in conventional PSO algorithms. To enhance the resistance of PSO to early convergence, one approach is to exclude the inertia weight component from the initial velocity element in the velocity equation of the typical PSO algorithm. Equation (7) introduces the first velocity term (k) to represent the velocity for VP_PSO.

$$V_i^{(k+1)} = V_i^{(k)^{RAND(0,1)*s^{(k)}}} + c1*r1*\left(S_{Lbest(i)} - X_i^{(k)}\right) + c2*r2*\left(S_{Gbest} - x_i^{(k)}\right) \quad (7)$$

$Where\ s^{(k)}\ can\ be\ expressed\ as: exp^{-\left(\frac{b_k}{K}\right)^b}, 'b' is\ a\ constant,\ c1\ and\ c2: Acceleration\ coefficients, r1\ and\ r2: Random\ coefficients,$ $V_i^{(k)}: Velocity\ component\ of\ swarm, X_i^{(k)}: Position\ of\ swarm,$ $S_{Lbest(i)}: Swarm\ individual\ best, S_{Gbest}: Swarm\ global\ best$

To update its velocity in the VP_PSO algorithm, a particle utilizes the velocity pausing perception technique while incorporating the modified velocity from Equation (7). Subsequently, the exact location of a particle is revised using Equation (8).

$$V_i^{(k+1)} = \begin{cases} V_i^{(k)}, & if\ RAND(0,1) < \alpha_{vppso} \\ V_i^{(k+1)}, & as\ per\ Eq.(10)\ \ Otherwise \end{cases} \quad (8)$$

The parameter α_{vppso} indicates the velocity pausing. If the value of the pause parameter α_{vppso} exceeds 1, each particle will adjust its velocity throughout each iteration using the normal PSO technique. This situation is unfavorable since it prevents any velocity interruptions. Conversely, a very small value of α_{vppso} will force particles to maintain a uniform velocity and restrict them from accelerating or decelerating. Therefore, it is crucial to choose the most suitable value for α_{vppso} to get a well-balanced velocity pausing situation, which may eventually lead to optimal performance. The value of α_{vppso} is assumed to be 0.3 according to reference [52].

The VP_PSO approach divides the whole population θ into two distinct swarms to maintain diversity and avoid premature convergence. The initial swarm consists of $\theta_{\ p1}$ particles that modify their velocities and positions using the conventional PSO technique, with a little alteration as follows: Equation (8) adds a revision to the first part of the velocity equation and includes the idea of velocity pausing. The second swarm consists of $\theta_{\ p2}$ particles that change their locations based on the global best (S_{Gbest}). Each particle inside the second swarm changes position according to Equation (9).

$$x_i^{(k+1)} = \begin{cases} S_{Gbest} + a^{(k)} * RAND(0,1) * |S_{Gbest}|^{a^{(k)}}, & if\ RAND(0,1) < 0.5\ (9.1) \\ S_{Gbest} - a^{(k)} * RAND(0,1) * |S_{Gbest}|^{a^{(k)}}, & Otherwise\ (9.2) \end{cases}$$

4.2 VP_PSO with Dilation CNN

The dilation CNN structure is represented as $\varphi = \{\varphi_{D\_CL}, \varphi_{PL}, \varphi_{FCL}\}$, where φ represents the collection of structural parameters that include the parameter annotations of dilation convolution layer (φ_{D_CL}), pooling (φ_{PL}) and fully-connected layers (φ_{FCL}). The dilation convolutional parameter set φ_{D_CL} is specified in the problem as $\varphi_{D\_CL} = \{c_1, c_1, \ldots . c_{k-1}\}$, where $'k'$ represents the number of dilation convolutional layers and $c_i = (c_{count}, c_{size}, c_{dilation\_rate})$ represents the configuration tuple of the i^{th} layer in the dilation convolutional layer. c_{count} reflects the number of kernels in each layer, whereas c_{size} defines the size of the kernel and $c_{dilation\_rate}$ describes the dilation rate in the i^{th} dilation convolutional layer. Similarly, φ_{PL} represents the set of pooling layers, denoted as

$\varphi_{PL} = \{pl_1, pl_2, \dots \dots, pl_{k-1}\}$. φ_{FCL} represents the set of fully connected layers, denoted as $\varphi_{FCL} = \{fcl_1, fcl_2, \dots \dots, fcl_{k-1}\}$. While the structural parameters of dilation CNN have an important role in its performance, the hyper-parameters such as the activation function (σ_{dl_cnn}), learning rate (α_{dl_cnn}), hidden neurons (μ_{dl_cnn}) and optimizer (ρ_{dl_cnn}) have a significant impact on obtaining optimal performance. Let θ be the set of possible hyper-parameters for a dilation CNN denoted as $\rho = \{f_{count}, f_{size}, \sigma_{dl\_cnn}, \alpha_{dl\_cnn}, \mu_{dl\_cnn}, \rho_{dl\_cnn}\}$. The objective is to identify the optimal configuration θ^* that minimizes the classification error. In the dilation CNN classification problem, each hyper-parameter in the search space corresponds to an individual swarm in the VP_PSO optimization. The boundaries are determined by upper and lower limitations as shown in Table 2. The suggested VP_PSO with dilation CNN complete construction framework is shown in Table 3. Additionally, various frameworks illustrated those that have been optimized using the PSO algorithms. The general structure of the proposed VP_PSO with the dilation CNN algorithm (Algorithm 1) is demonstrated in Figure 3.

Table 2. Hyper-parameter details with their specified search space

Hyper-parameters	**Specified search space**
Number of filters (f_{count})	[8,16,32,64,128]
Filter size (f_{size})	[2,3,4,5]
Activation function (σ_{dl_cnn})	['sigmoid', 'relu', 'tanh']
Learning rate (α_{dl_cnn})	[0-1]
No. of neurons for hidden layers (μ_{dl_cnn})	[32,64,128,256]
Optimizer (ρ_{dl_cnn})	['Adam', 'AdaDelata', 'AdaMax', 'Nadam', 'AdaGrad', 'SGD', 'RMSprop']

Table 3. Optimized hyper-parameters representation of dilation CNN model utilizing different optimization techniques

	Optimal hyper-parameters values	
Fixed Layers for all experimental dilation CNN model	**PSO with dilation CNN**	**VP_PSO with dilation CNN**
Dilation Convolution Layer: 3 Pooling Layer: 3 Hidden Layer in FCN: 2	$f_{count} = 64$ $f_{size} = 3$ $\sigma_{dl\_cnn} = 'tanh'$ $\alpha_{dl\_cnn} = 0.002$ $\mu_{dl\_cnn} = 64$ $\rho_{dl\_cnn} = 'SGD'$	$f_{count} = 32$ $f_{size} = 2$ $\sigma_{dl\_cnn} = 'relu'$ $\alpha_{dl\_cnn} = 0.001$ $\mu_{dl\_cnn} = 32$ $\rho_{dl\_cnn} = 'Adam'$

Algorithm 1. VP_PSO

1. Declare the essential parameters of VP_PSO
$\theta = Total\ no.of\ particles, Max_{itr} = Maximum\ iterations$
$\theta_{\ p1} = First\ group\ particles, \theta_{\ p2} = Second\ group\ particles$
$X_{(i)} = Particle\ position, V_{(i)} = Velocity\ of\ particles,$
$f_{dl\_cnn}() = Objective\ function, S_{Lbest} = Individual\ best\ of\ particle,$
$S_{Gbest} = Global\ best\ of\ particle$
$\rho = Potential\ search\ space\ of\ hyper-parameters,$
$r1, r2 = Randomized\ coefficients, \alpha = Velocity\ paused\ parameter$
$c1, c2 = Acceleration\ coefficients, D = Dataset, K = Dimension\ depth$
2. Randomly generate the particle position $x_{(i\_k)}, (i = 1,2,3, \dots, \theta), (k = 1,2, \dots, K)$
3. Assign 0 to all the particle velocity $V_{(i\_k)}, (i = 1,2,3, \dots, \theta), (k = 1,2, \dots, K)$
4. Compute the fitness value of each particle by $f_{dl\_cnn}(.)$ fitness function
5. Set all computed fitness values of each particle as individual best value $S_{Lbest(i\_k)}$
$S_{Lbest(i\_k)} = f_{dl\_cnn}(x_{(i\_k)})$, $(i = 1,2,3, \dots, \theta), (k = 1,2, \dots, K)$
6. Set best-computed fitness value among all populations as global best value $S_{Gbest(k)}$
$S_{Gbest(k)} = max\ (S_{Lbest})$
7. $for\ itr = 1\ to\ Max_{itr}, do$
8. $for\ i = 1\ to\ \theta\ , do$
9. $if\ i \le \theta_{\ p1}, then$
10. update the velocity $V_{(i\_k)}$ for each particle i by applying $Eq.$ (8)
11. update the position $x_{(i\_k)}$ for each particle i by applying $Eq.$ (9.1)
12. $else$
13. update the position $x_{(i\_k)}$ for each particle i by applying $Eq.$ (9.2)
14. $End\ if$
15. $End\ for$
16. $for\ i = 1\ to\ \theta\ , do$
17. compute fitness value of new particle position $f_{dl\_cnn}(x_{(i\_k)})$
18. $if\ i \le \theta_{\ p1}, then$
19. $if\ f_{dl\_cnn}(x_{(i\_k)}) > f_{dl\_cnn}(S_{Lbest(i\_k)})$
20. $S_{Lbest(i\_k)} = x_{(i\_k)}$
21. $End\ if$
22. $if\ f_{dl\_cnn}(P_{Lbest(i\_k)}) > f_{dl\_cnn}(S_{Gbest(k)})$
23. $S_{Gbest(k)} = S_{Lbest(i\_k)}$
24. $End\ if$
25. $else$
26. $if\ f_{dl\_cnn}(x_{(i\_k)}) > f_{dl\_cnn}(S_{Gbest(k)})$
27. $S_{Gbest(k)} = x_{(i\_k)}$
28. $End\ if$
29. $End\ if$
30. $End\ for$
31. $End\ for$
32. Obtain the optimal S_{Gbest}

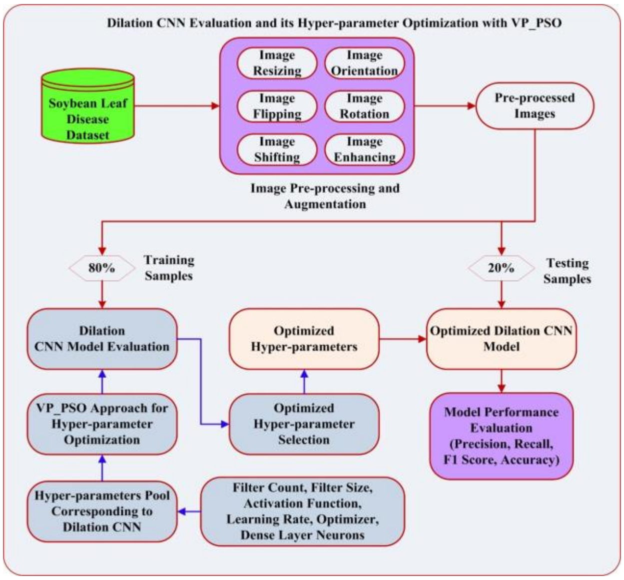


Figure 3: Architectural diagram of proposed VP_PSO with dilation CNN

5 DATASET OVERVIEW AND SIMULATION ENVIRONMENT

This section presents details regarding the specifications of the soybean leaf disease dataset and the simulation environment adopted to implement the suggested framework for enhanced performance.

5.1. Dataset Overview

The soybean leaf dataset comprised six types of diseased leaf images acquired from Xiangyang Farm, Jiusan Farm and Nengjiang Farm of Northeast Agricultural University in Heilongjiang Province from June to late September 2019 and is publicly available in IEEE Dataport [53], which is utilized here for the evaluation of the performance of the proposed model. This soybean leaf disease dataset contains 1620 diseased leaf images of six different classes. Six diseased classes include the name of the disease as a bacterial disease, downy mildew, pesticide, spider mite, viral disease and worm eye. The collected classes of diseased soybean leaf images are illustrated in Figure 4.

Figure 4. Six diseases leave samples of the soybean dataset

All the soybean leaf images are classified into two parts i.e. 1) training set and 2) testing set in a proportion of 80% and 20% described in Table 4. These two kinds of data have been implemented in this proposed model for learning and testing of this model.

Table 4. Number of soybean leaf images used for training and testing

Disease Type in Soybean Dataset	Total Samples	Training Samples (80%)	Testing Samples (20%)
Bacterial disease	270	216	54
Downy mildew	270	216	54
Pesticide	270	216	54
Spider Mite	270	216	54
Viral disease	270	216	54
Worm eye	270	216	54

5.2. Simulation Environment

The experimental setup deployed an adequate system configuration including an 11th Generation Intel(R) Core (TM) i7-11300H processor working at a clock speed ranging from 3.10GHz to 3.11GHz. It is accompanied by 32GB of RAM and runs on the Windows 11 operating system. By using the computational capabilities of Google Colab Pro's Jupyter Notebook environment, the experimenting process has been made convenient. The effective application of Python libraries such as TensorFlow and Keras facilitated the exploitation of the functionalities of DL frameworks, allowing for effective building, training and assessment of models. The scikit-image (skimage) and io libraries were implemented to assist image pre-processing, guaranteeing efficient data preparation. Matplotlib has been employed for visualizing the findings and conducting data analysis.

6 RESULT ANALYSIS

This section demonstrates an explanation of the suggested VP_PSO _ dilation CNN model along with other experimental models that have been employed in diagnosing soybean leaf diseases. The experimental results of these models are also shown.

6.1 Performance Measures

Table 5. Details of performance measures with their formulas

Performance Metric	Usage of the metric	The formula used for this metric
Accuracy	Accuracy is one approach to determine how frequently our proposed	$$\frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (10)$$

	classification models accurately classify a leaf image and also offers the percentage of actual results.	
Recall	The total true positive and false negative of every class are added to evaluate the efficacy of the classifier in identifying class labels.	$\frac{T_P}{T_P + F_N}$ (11)
Precision	Agreement exists between the classifiers and true class labels. Classifiers are determined by combining all true positive and false positive of all the system for every class.	$\frac{T_P}{T_P + F_P}$ (12)
F1 Score	It evaluates the effectiveness of the identification of leaf images by proving equal importance to recall as well as precision. Specifically, it is the harmonic mean of recall and precision.	$2 \times \frac{Precision \times Recall}{Precision + Recall}$ (13)

where, T_P : *true positive*, T_N : *true negative*,

F_P : *false positive and* F_N : *false negative*

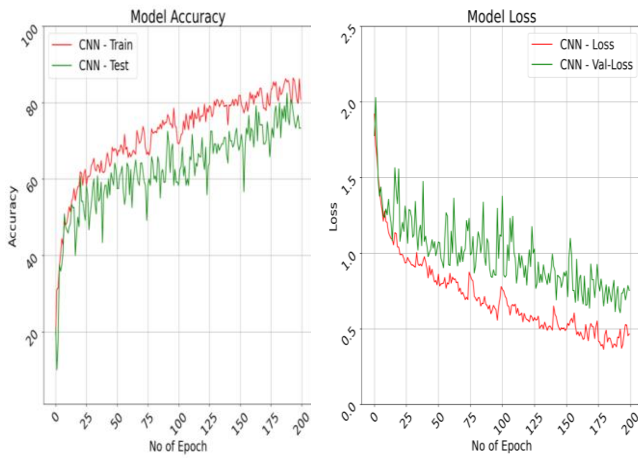
6.2 Result Analysis

The results of the proposed VP_PSO_Dilation CNN architecture and various other classifiers like PSO_Dilation CNN, Dilation CNN, ResNet101, MobiNetV2, VGG16 and classical CNN are shown in below Table 6. The highest accuracy achieved by the proposed VP_PSO_Dilation CNN is 95.32%. According to other performance metrics calculated, the value of recall is 0.9537, the value of precision is 0.9538 and the value of F1 Score is 0.9536. The classifier PSO_Dilation CNN has obtained the value of precision is 0.9474, the value of recall is 0.9472and the value of F1 Score is 0.9473. This PSO_Dilation CNN classifier has achieved an accuracy of 94.72% which remains the second-highest accurate classifier in this paper. In comparison to these above-mentioned architectures, Dilation CNN, ResNet101, MobiNetV2, VGG16 and classical CNN have achieved the accuracy of 74.16%, 41.66%, 72.50%, 55.83% and 73.33% respectively. The F1 score of Dilation CNN, ResNet101, MobiNetV2, VGG16 and classical CNN are 0.7512, 0.3648, 0.6900, 0.5560 and 0.7498 respectively. Based on the experimental analysis, VP_PSO_Dilation CNN architecture has maintained its supremacy based on accuracy and F1 score.

Table 6. Performance analysis of the proposed method with other DL approaches

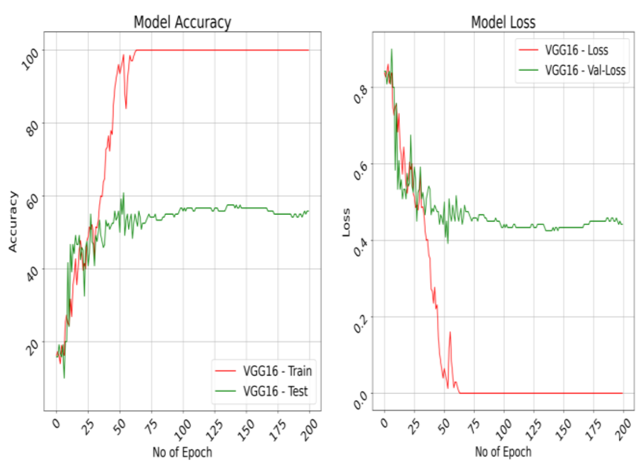
Model	Precision	Recall	F1 Score	Accuracy (%)
CNN	0.7614	0.7558	0.7498	73.33
VGG16	0.5720	0.5681	0.5560	55.83
MobiNetV2	0.8238	0.7128	0.6900	72.50
ResNet101	0.3391	0.4677	0.3648	41.66
Dilation CNN	0.7652	0.7423	0.7512	74.16
PSO_Dilation CNN	0.9474	0.9472	0.9473	94.72
Proposed VP_PSO_Dilation CNN	0.9538	0.9537	0.9536	95.32

The training and testing accuracy of classical CNN, VGG16, MobileNetV2, ResNet101 and dilation CNN are shown in Figure 5(a), 6(a), 7(a), 8(a) and 9(a) respectively. As per the outcomes of these experimental results, dilation CNN achieves an adequate convergence level at the starting level. In this article, all the experiments are done with the epoch value 200. The testing accuracy of VGG16 and ResNet101 is too low, so they do not satisfy the objectives. In comparison to CNN and MobileNetV2, dilation CNN is performing better in connection with training and testing accuracy. Furthermore, it offers an adequate growth rate after the epoch of 110. The effectiveness of dilation CNN increased gradually during both the training and testing phases. The training and validation loss of classical CNN, VGG16, MobileNetV2, ResNet101 and dilation CNN are depicted in Figure 5(b), 6(b), 7(b), 8(b) and 9(b). The experiment observes that CNN, VGG16, MobileNetV2 and ResNet101 have obtained an inappropriate loss while reaching the epoch at 200. In comparison to CNN, VGG16, MobileNetV2 and ResNet101, the dilation CNN model performs well, demonstrating a steady reduction in train and validation loss after epoch 150. However, as per the loss analysis, the effectiveness of this classifier is declining during the training and testing phase.



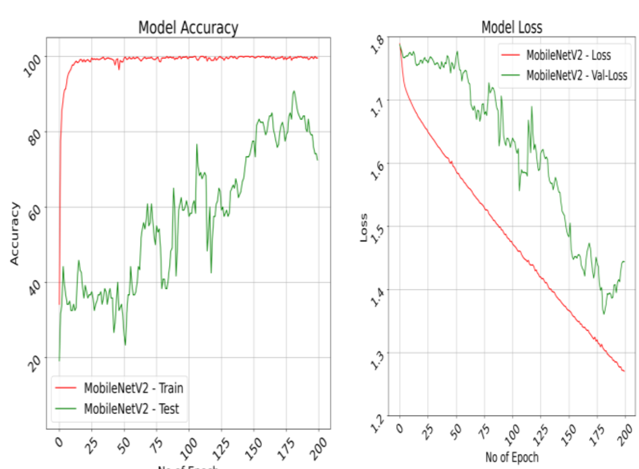
(a) CNN_Accuracy (b) CNN_Loss

Figure 5. Accuracy and loss analysis of the CNN model



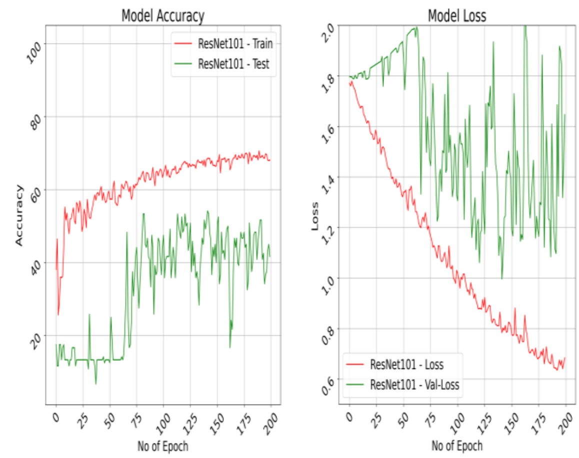
(a) VGG16_Accuracy (b) VGG16_Loss

Figure 6. Accuracy and loss analysis of the VGG16 model



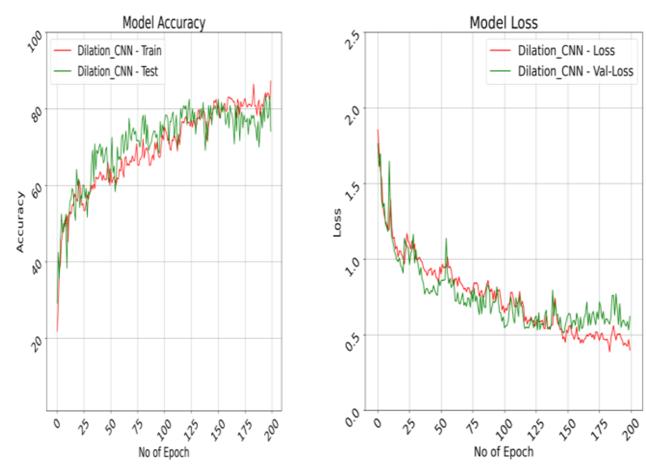
(a) MobileNetV2_Accuracy (b) MobileNetV2_Loss

Figure 7. Accuracy and loss analysis of MobileNetV2 model



ResNet101_Accuracy ResNet101_Loss

Figure 8. Accuracy and loss analysis of the ResNet101 model



(a) dilation CNN _Accuracy (b) dilation CNN _Loss

Figure 9. Accuracy and loss analysis of the ResNet101 model

Figure 10 displays the testing accuracy of VP_PSO_Dilation CNN and PSO_Dilation CNN. The testing accuracy of VP_PSO_Dilation CNN and PSO_Dilation CNN shows strong efficiency in convergence at the start of the epoch, ranging from 4 to 10. In the recent study, the maximum range of epochs taken in this investigated architecture with all optimized techniques is 50. However, the efficacy of the PSO_Dilation CNN model is compared with the VP_PSO_Dilation CNN model to achieve the purpose. VP_PSO_Dilation CNN has a sustainable increase in consistency after epoch 38. The VP_PSO_Dilation CNN continuously augmented the testing accuracy of 95.32% and surpassed comparable methodologies such as CNN, VGG16, MobileNetV2, ResNet101, dilation CNN and PSO_Dilation CNN.

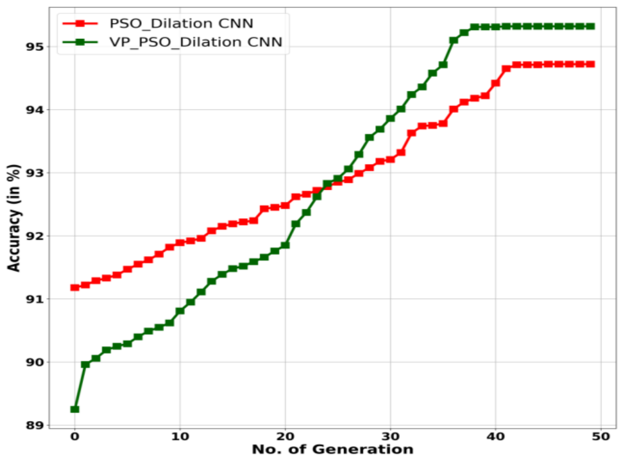


Figure 10. Accuracy analyses of VP_PSO_Dilation CNN and PSO_Dilation CNN models

Figure 11 shows the testing loss for VP_PSO_Dilation CNN and PSO_Dilation CNN. The loss incurred during the testing phase of VP_PSO_Dilation CNN and PSO_Dilation CNN signifies a sustainable convergence while reaching the epoch value at 37. This optimized framework is experimented with a maximum epoch value of 50. The effectiveness of the PSO_Dilation CNN model is compared with the VP_PSO_Dilation CNN model to obtain a significant minimization in the loss. The VP_PSO_Dilation CNN continuously enhanced its testing accuracy and outperformed models like CNN, VGG16, MobileNetV2, ResNet101, dilation CNN and PSO_Dilation CNN. From Figure 10, it is obvious that VP_PSO_Dilation CNN has a very trivial probability of under-fitting as well as over-fitting.

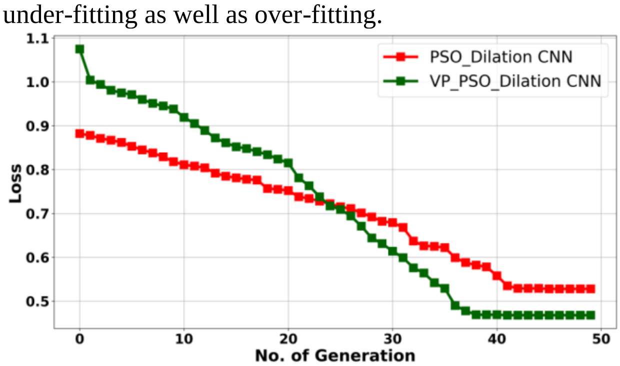


Figure 11. Loss analyses of VP_PSO_Dilation CNN PSO_Dilation CNN models

The proposed VP_PSO_Dilation CNN is analyzed with another optimized technology-based model such as PSO_Dilation CNN, as well as compared with the most emerging techniques like CNN, VGG16, MobileNetV2, ResNet101 and dilation CNN. The outcomes of comparative analysis for Precision, Recall and F1 score are shown in Figures 12, 13 and 14. The proposed VP_PSO_Dilation CNN approach attains a precision of 0.9538, recall of 0.9537 and a best possible F1 score of 0.9536 in comparison with other empirical outcomes. Figure 15 presents an overall comparison of all the experimented models that were investigated altogether and the proposed VP_PSO_Dilation CNN model is found to be a superior model across all other models with respect to all the performance metrics.

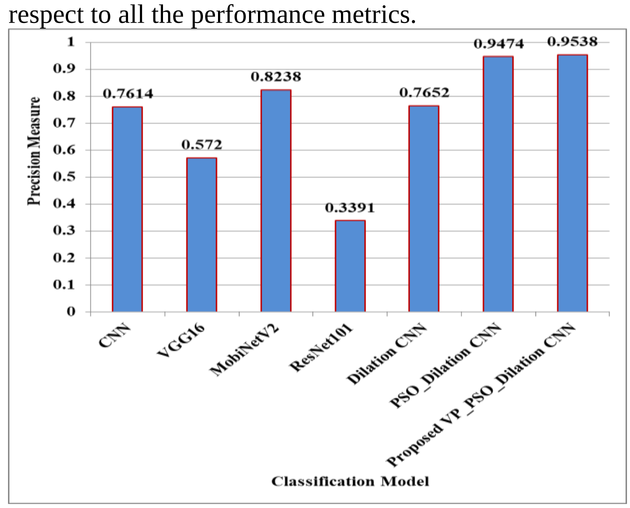


Figure 12. Precision analyses of all comparison models with the proposed model

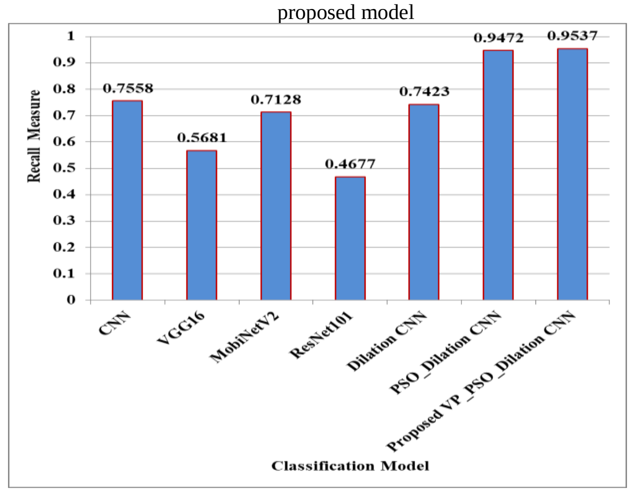


Figure 13. Recall analyses of all comparison models with the proposed model

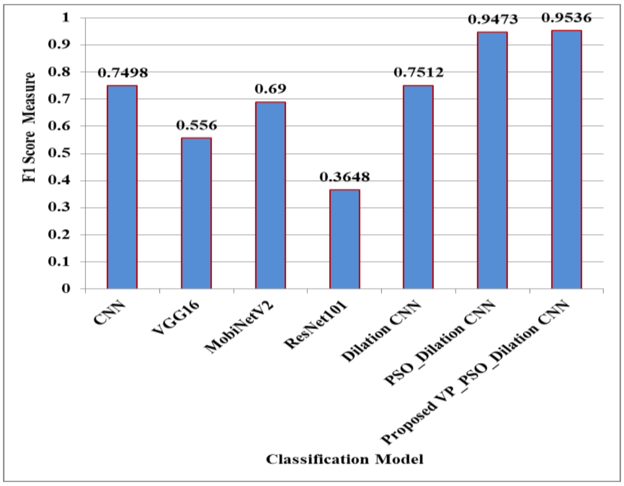


Figure 14. F1 score analyses of all comparison models with the proposed model

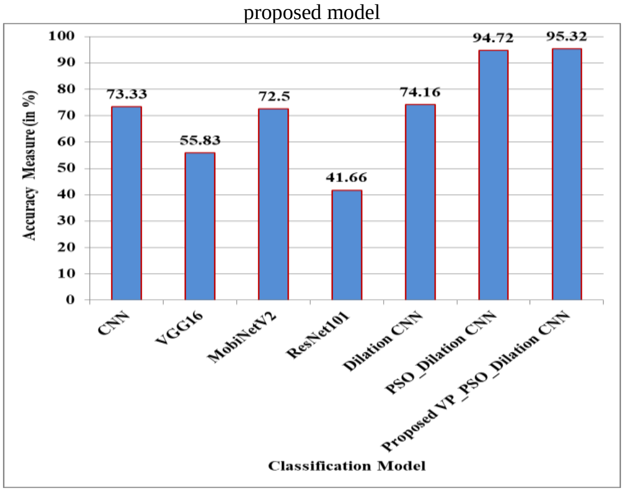


Figure 15. Overall performance analysis of VP_PSO_Dilation CNN with all comparison models

7 Conclusion and Future Scope

The goal of this research is to develop a meta-heuristic strategy that implements the VP_PSO methodology to design a classification system for infectious diseases of soybean leaves. Moreover, the VP_PSO is accountable for automatically optimizing the dilation CNN architecture's hyper-parameter values. A detailed experimental analysis on the proposed VP_PSO_dilation CNN models with the PSO_dilation CNN algorithms implemented for hyper-parameter optimization has been performed. Comparing the simulation results with those of other comparable methods, like PSO_dilation CNN exposes that the method proposed here is quite promising and outperforms those other approaches. The future scope of this research will involve an expanded investigation that uses more sophisticated swarm intelligence-based algorithms and applies them to automatically create CNN architecture on bigger datasets of soybean leaf disease.

References

[1] Pal, Arunangshu, and Vinay Kumar. "AgriDet: Plant Leaf Disease severity classification using agriculture detection framework." Engineering Applications of Artificial Intelligence 119 (2023): 105754.

[2] Shewale, Mitali V., and Rohin D. Daruwala. "High performance deep learning architecture for early detection and classification of plant leaf disease." *Journal of Agriculture and Food Research* 14 (2023): 100675.

[3] Agarwal, Mohit, et al. "ToLeD: Tomato leaf disease detection using convolution neural network." *Procedia Computer Science* 167 (2020): 293-301.

[4] Bangal, Anushka, et al. "Potato Leaf Disease Detection and Classification Using CNN." *Int. J. Res. Publ. Rev. J. homepage www. ijrpr. com* 3.5 (2022): 1510-1515.

[5] Ning, Hongwei, et al. "Convolutional neural network in rice disease recognition: accuracy, speed and lightweight." *Frontiers in Plant Science* 14 (2023): 1269371.

[6] K. Nageswararao , A. S. L. "Apple Disease Detection Using Convolutional Neural Networks". International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 21s, Mar. 2024, pp. 466-70, https://ijisae.org/index.php/IJISAE/article/view/5442.

[7] Wallelign, Serawork, Mihai Polceanu, and Cédric Buche. "Soybean plant disease identification using convolutional neural network." *The thirty-first international flairs conference*. 2018.

[8] Bevers, Noah, Edward J. Sikora, and Nate B. Hardy. "Soybean disease identification using original field images and transfer learning with convolutional neural networks." *Computers and Electronics in Agriculture* 203 (2022): 107449.

[9] Agarwal, Dinesh K., et al. "Soybean: introduction, improvement, and utilization in India—problems and prospects." *Agricultural Research* 2.4 (2013): 293-300.

[10] Hartman, G. L., West, E. D., & Herman, T. K. (2011). Crops that feed the World 2. Soybean-worldwide production, use, and constraints caused by pathogens and pests. Food Security, 3(1), 5-17. httpsdoi.org10.1007s12571-010-0108-x

[11] Kalaitzandonakes, Nicholas, James Kaufman, and Kenneth Zahringer. The economics of soybean disease control. CABI, 2019.

[12] Ramesh, Shima, et al. "Plant disease detection using machine learning." *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*. IEEE, 2018.

[13] Singh, Jaskaran, and Harpreet Kaur. "Plant disease detection based on region-based segmentation and KNN classifier." *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*. Springer International Publishing, 2019.

[14] Chaudhary, Archana, Savita Kolhe, and Raj Kamal. "An improved random forest classifier for multi-class classification." *Information Processing in Agriculture* 3.4 (2016): 215-222.

[15] Das, Debasish, et al. "Leaf disease detection using support vector machine." *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020.

[16] Nithyashree, D., et al. "Plant disease detection using decision tree algorithm and automated disease cure." *Int. J. Eng. Technol* 7.3 (2020): 1834-1838.

[17] Li, Yang, Jing Nie, and Xuewei Chao. "Do we really need deep CNN for plant diseases identification?." *Computers and Electronics in Agriculture* 178 (2020): 105803.

[18] J. Kolli, D. M. Vamsi and V. M. Manikandan, "Plant Disease Detection using Convolutional Neural Network," 2021 IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, 2021, pp. 1-6, doi: 10.1109/IBSSC53889.2021.9673493.

[19] Sun, Xuewei, et al. "Research on plant disease identification based on CNN." *Cognitive Robotics* 2 (2022): 155-163.

[20] Zhang, Shanwen, et al. "Cucumber leaf disease identification with global pooling dilated convolutional neural network." *Computers and Electronics in Agriculture* 162 (2019): 422-430.

[21] Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." *arXiv preprint arXiv:1207.0580* (2012).

[22] Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." *arXiv preprint arXiv:1511.07122* (2015).

[23] Renton, Guillaume, et al. "Fully convolutional network with dilated convolutions for handwritten text line segmentation." *International Journal on Document Analysis and Recognition (IJDAR)* 21 (2018): 177-186.

[24] Lestari, Cynthia Ayu Dwi, Syaiful Anam, and Umu Sa'adah. "Tomato Leaf Disease Classification with Optimized Hyperparameter: A DenseNet-PSO Approach." *First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*. Atlantis Press, 2024.

[25] Singh, Pratibha, Santanu Chaudhury, and Bijaya Ketan Panigrahi. "Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network." *Swarm and Evolutionary Computation* 63 (2021): 100863.

[26] Darwish, Ashraf, Dalia Ezzat, and Aboul Ella 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 (2020): 100616.

[27] Albelwi, Saleh, and Ausif Mahmood. "A framework for designing the architectures of deep convolutional neural networks." *Entropy* 19.6 (2017): 242.

[28] Yeh, Wei-Chang, et al. "Simplified swarm optimization for hyperparameters of convolutional neural networks." *Computers & Industrial Engineering* 177 (2023): 109076.

[29] Sun, Yanan, et al. "Evolving deep convolutional neural networks for image classification." *IEEE Transactions on Evolutionary Computation* 24.2 (2019): 394-407.

[30] Suganuma, Masanori, Shinichi Shirakawa, and Tomoharu Nagao. "A genetic programming approach to designing convolutional neural network architectures." *Proceedings of the genetic and evolutionary computation conference*. 2017.

[31] Bin Wang , Yanan Sun , Bing Xue , Zhang Mengjie , in: Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classifi- cation, IEEE, 2018, pp. 1–8 .

[32] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. ieee, 1995.

[33] A. Reiling, W. Mitchell, S. Westberg, E. Balster and T. Taha, "CNN Optimization with a Genetic Algorithm," 2019 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 2019, pp. 340-344, doi: 10.1109/NAECON46414.2019.9058307.

[34] M. Ünal, M. Onat and A. Bal, "Cellular Neural Network training by ant colony optimization algorithm," 2010 IEEE 18th Signal Processing and Communications Applications Conference, Diyarbakir, Turkey, 2010, pp. 471-474, doi: 10.1109/SIU.2010.5653917.

[35] Yang, Xin-She, and Xingshi He. "Firefly algorithm: recent advances and applications." *International journal of swarm intelligence* 1.1 (2013): 36-50.

[36] Z. Cui , B. Sun , G. Wang , et al. , A novel oriented cuckoo search algorithm to im- prove DV-Hop performance for cyber-physical systems, J. Parallel Distr. Comput. 103 (2017) 42–52 .

[37] Karlekar, Aditya, and Ayan Seal. "SoyNet: Soybean leaf diseases classification." *Computers and Electronics in Agriculture* 172 (2020): 105342.

[38] Bevers, Noah, Edward J. Sikora, and Nate B. Hardy. "Soybean disease identification using original field images and transfer learning with convolutional neural networks." *Computers and Electronics in Agriculture* 203 (2022): 107449.

[39] Pan, Renyong, et al. "A two-stage feature aggregation network for multi-category soybean leaf disease identification." *Journal of King Saud University-Computer and Information Sciences* 35.8 (2023): 101669.

[40] Wu, Qinghai, et al. "A classification method for soybean leaf diseases based on an improved ConvNeXt model." *Scientific Reports* 13.1 (2023): 19141.

[41] Yu, Miao, et al. "A diagnosis model of soybean leaf diseases based on improved residual neural network." *Chemometrics and Intelligent Laboratory Systems* 237 (2023): 104824.

[42] Yu, Miao, et al. "A recognition method of soybean leaf diseases based on an improved deep learning model." *Frontiers in plant science* 13 (2022): 878834.

[43] Fang, Wang, Liu Yue, and Cui Dandan. "Classification system study of soybean leaf disease based on deep learning." *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*. IEEE, 2020.

[44] Abade, André, et al. "NemaNet: A convolutional neural network model for identification of soybean nematodes." *Biosystems Engineering* 213 (2022): 39-62.

[45] Gui, Jiangsheng, and Mor Mbaye. "Identification of Soybean Leaf Spot Diseases using Deep Convolutional Neural Networks." *International Journal of Engineering Research and Technology (IJERT)* 8.10 (2019): 289-294.

[46] Goshika, Sandeep, et al. "Deep Learning Model for Classifying and Evaluating Soybean Leaf Disease

Damage." *International Journal of Molecular Sciences* 25.1 (2023): 106.

[47] J. Yao, W. Xing, D. Wang, J. Xing, L. Wang, Active dropblock: Method to enhance deep model accuracy and robustness, Neurocomputing 454 (2021) 189–200.

[48] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Advances in neural information processing systems, 2016, pp. 4898–4906.

[49] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: 4th International Conference on Learning Representations, ICLR, 2016.

[50] F. Yu, V. Koltun, T. Funkhouser, Dilated residual networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 472–480.

[51] J. Xu, W. Wang, H. Wang, J. Guo, Multi-model ensemble with rich spatial information for object detection, Pattern Recognition 99 (2020) 107098.

[52] Shami, Tareq M., et al. "Velocity pausing particle swarm optimization: a novel variant for global optimization." Neural Computing and Applications 35.12 (2023): 9193-9223. Doi: 10.1007/s00521-022-08179-0.

[53] Yan Hang. (2024). Soybean disease data set in natural scenes. IEEE Dataport. https://dx.doi.org/10.21227/tc50-b707

BHABANISANKAR JENA Bhabanisankar Jena is a dedicated research scholar at the Postgraduate Department of Computer Science at Fakir Mohan University. He holds MCA degree from Fakir Mohan University. He is engaged in advanced research, focusing on areas such as machine learning, deep learning, and artificial intelligence and also has published papers in this domain. His work aims to develop innovative computational models and algorithms to solve complex problems in various domains. He is passionate about leveraging technology to create impactful solutions and continuously seeks to expand his knowledge and expertise in the evolving landscape of computer science.

Dr. ASHANTA RANJAN ROUTRAY Dr. Ashanta Ranjan Routray is the Head of the Department of Computer Science at Fakir Mohan University. He holds a Master of Engineering in Computer Science and Engineering and a Ph.D. in Computer Science. With 24 years of experience in the field, Dr. Routray has an extensive publication record in various reputable journals. His research interests span across multiple areas within computer science, reflecting his deep expertise and commitment to advancing the discipline.

Dr. Janmenjoy Nayak is working as an Assistant Professor, P.G. Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo(MSCB) University, Baripada, Mayurbhanj, Odisha, India. Being two times Gold Medalist in Computer Science in his career, he has been awarded with INSPIRE Research Fellowship from Department of Science & Technology, Govt. of India (both as JRF and SRF level) and Best researcher award from Jawaharlal Nehru University of Technology, Kakinada, AP for the AY: 2018-19 and many other awards from national and international academic agencies. He has edited 22 books and 10 Special Issues on the applications of Computational Intelligence, Soft Computing, and pattern recognition, published by reputed International publications. He has published more than 200+ referred articles in various book chapters, conferences and International repute peer reviewed journals of Elsevier, Inderscience, Springer, IEEE etc. He has a total twelve years of experience in both teaching and research. He is the Senior member of IEEE and life member of some of the reputed societies like CSI India, IAENG (Hongkong) etc. He has successfully conducted and is being associated with 15 International repute series conferences like ICCIDM, HIS, ARIAM, CIPR, SCDA etc. His area of interest includes data mining, nature inspired algorithms and soft computing.