Computers, Materials & Continua
DOI:10.32604/cmc.2022.022215
Article

Tech Science Press

An Optimized Convolution Neural Network Architecture for Paddy Disease Classification

Muhammad Asif Saleem[1], Muhammad Aamir[1,2,*], Rosziati Ibrahim[1], Norhalina Senan[1] and Tahir Alyas[3]

[1]Faculty of Computer Science, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 54000, Malaysia
[2]School of Electronics, Computing and Mathematics, University of Derby, Derby, DE22 1GB, United Kingdom
[3]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
*Corresponding Author: Muhammad Aamir. Email: m.aamir@derby.ac.uk
Received: 31 July 2021; Accepted: 16 November 2021

Abstract: Plant disease classification based on digital pictures is challenging. Machine learning approaches and plant image categorization technologies such as deep learning have been utilized to recognize, identify, and diagnose plant diseases in the previous decade. Increasing the yield quantity and quality of rice forming is an important cause for the paddy production countries. However, some diseases that are blocking the improvement in paddy production are considered as an ominous threat. Convolution Neural Network (CNN) has shown a remarkable performance in solving the early detection of paddy leaf diseases based on its images in the fast-growing era of science and technology. Nevertheless, the significant CNN architectures construction is dependent on expertise in a neural network and domain knowledge. This approach is time-consuming, and high computational resources are mandatory. In this research, we propose a novel method based on Mutant Particle swarm optimization (MUT-PSO) Algorithms to search for an optimum CNN architecture for Paddy leaf disease classification. Experimentation results show that Mutant Particle swarm optimization Convolution Neural Network (MUTPSO-CNN) can find optimum CNN architecture that offers better performance than existing hand-crafted CNN architectures in terms of accuracy, precision/recall, and execution time.

Keywords: Deep learning; optimum CNN architecture; particle swarm optimization; convolutional neural network; parameter optimization

1 Introduction

Early detection and categorization of diseases at crops is one of the most essential agricultural practices. Every year infectious disease causes a significant economic loss to farmers. The symptoms and indications caused by the pathogens are used to identify and classify rice plant diseases. Multiple countries around the world depend on agriculture for fulfilling their food needs [1]. In these countries, farmers are facing several issues like shortage of water, natural disaster, and disease on plants. According to data, rice plant disease destroys 10–15 percent of yields in Asia, and the source of this destruction is a fungus and viral disease on rice plant [2]. The example of diseases produced by viral and fungus-like rice blast, bacterial leaf blast and sheath blight. Mostly diseases attack on leaves of plants and their on time recognition is most important. However, identifying the diseases on crops manually based on symptoms might be challenging. Manual inspection of crops is a difficult due large area of fields. It's quite impossible for farmers to classify the rice leaves as infected or not by disease [3]. It is also very important for the farmer to know about the type of disease for curing it. Due to lack of knowledge and focus on the application of technology in agriculture. The production of rice is reduced up to 37% overall worldwide [4]. Thus, to remove such issues, technical help from the scientific approach is necessary. Taking early action against disease may result in curing it on time, this will help the farmer to maintain their quality as well as quantity.

Currently, machine learning techniques such as support vector machine (SVM), Gaussian Naïve Bays, Radial basis function network, deep neural network (DNN) are applied for image processing. CNN is latest one and have shown a remarkable [5]. CNN's architecture is inspired by the biological structure of mammal's visual cortexes. CNNs are consist of several types of layers like convolution, pooling, and fully connected. CNN learns the features from input data automatically by stacking several layers [6]. Alex-Net, Lenet are considered as fundamental CNN architecture. These architectures consisted of several hidden layers with millions of parameters and have achieved state of art performance on the ImageNet dataset with an error rate of 15.3% [7]. Due to the impressive performance of these fundamentals' architectures, CNN became much popular in the computer vision field. More reasons behind its success are application on large datasets, fast computation by using GPUs, and strong regularization techniques [8]. The success of CNNs has motivated researchers and practitioners to apply to several area such as engineering, medical and agricultures problems.

Researchers have worked on Alex-Net architecture for improving its accuracy. Performance can be enhanced by adjusting filter size and stride rate in the first convolution layer smaller. Study in [9] improved the accuracy by designing architecture while adding up to sixteen layers. The author highlighted that it is critical to achieve better performance by increasing the number of layers meanwhile study in [10] also showed practically that making architecture deeper can harm the performance. Additionally, it is very difficult to optimize deeper CNN architecture [10]. The performance of CNNs is purely dependent on architectural design. Each dataset needs a different architectural setup. Determining the appropriate architecture for different datasets is quite challenging as each dataset will be different from others in terms of classes in datasets, instances in each class, quality of images in datasets, and other reasons.

While designing the architecture of the CNN, there are multiple structural parameters that need to be considered. Some of these parameters are the number of convolution layers, number of pooling layers, number of filters, filter size, stride rate, and place of pooling layer. Due to the excess number of parameters, it is difficult to find an optimum combination of parameters showing remarkable performances. Researchers are creating architecture with a try and error approach [11]. Some of the researchers use grid search or random search techniques for identifying the optimum combination of parameters. Although, these techniques help in making suitable combination, however it is time consuming and requires high computational powers [11]. Nowadays researchers are considering the suitable combination of hyperparameters as an optimization problem. Automated methods [12] are improved in terms of accuracy, precision, and recall as compared to manually crafted architectures. These automated architectures work by utilizing prior knowledge for making hyperparameter combinations to reduce miss classification rate.

Thus, in this research, the main objective is to handle the problem of creating optimum CNN architecture for image classification in order to produce the best performance on the training and testing accuracy, precision and recall. For that, we are implementing a MUT-PSO algorithm for creating optimum CNN architecture automatically based on input data. For validating the proposed technique, the datasets of paddy leaf disease were used which shows the state-of-the-art performance.

The rest of the paper is organized as follows: Section 2 highlights existing studies and the algorithms used for the image classification, more specifically on paddy diseases. Furthermore, Section 3 explains the standard architecture of CNN and the MUTPSO is discussed in Section 4. Furthermore, Section 5 explains the proposed methodology of our research activity while Section 6 proposes the optimized CNN architecture generator algorithm. The simulation results are discussed in the Section 7 and finally, the paper is concluded in the Section 8.

2 Related Work

Many approaches and procedures have been used in the literature for the identification of rice crop diseases based on the images. Most of these techniques uses image processing for proposing a solution. Most commonly used methods for paddy leaf disease classification are Self Organizing Map, neural networks [12], Gaussian Naïve Bayes [13], SVM [14] and the latest technique is CNN. CNN performs extra ordinary in classification of paddy diseases due its strong convolution and pooling operations [15]. Several CNN architectures are proposed by researchers from 1998 to till date. Choosing and creating CNN architecture for a particular problem is purely manual, consuming lots of time and resources [16].

Researchers have contributed to propose an optimization algorithm. Each optimization algorithm has its own merits and demerits due which these several algorithms are made hybrid with others for optimum solutions. In [17] proposes a hybrid approach based on Genetic Algorithm and firefly algorithms for optimization problems. The simplest technique for choosing the best CNN architecture is cross-validation [18]. In this technique several CNN architectures are executed which are prepared manually by a large number of choices. As per the literature study, the most common strategy for hyperparameter optimization is grid search, which is well known for its drawback to the expensive computation. In the near past, a new technique random search [19] is also proposed which selects the hyperparameters randomly from search space. Its performance is better than grid search as it also requires less computation power. However, neither random nor grid search uses previous evaluations to select the next set of hyperparameters for testing to improve the desired architecture. Meanwhile, Bayesian Optimization (BO) is an optimization algorithm used by several researchers for the parameter's optimization of CNN. It optimizes the continuous hyper-parameter of the DNN [20] Meanwhile, [21] proposes an adaptive technique for updating hyperparameter automatically.

Evolutionary Algorithms (EA) are used in hyper-parameter optimization of learning algorithms. Nowadays EA is also widely used in automating architecture design. Reference [22] uses a genetic algorithm to optimize the number of filter and filter size in the convolution layer. In this research, the architecture was built consisting of three convolution layers and one fully connected layer. Soft computing techniques are applied for solving several real-time problems like rainfall prediction [23]. A Particle Swarm Optimization algorithm (PSO) is used for the optimization of Rainfall-runoff Modeling. Reference [24] uses PSO and extreme machine learning for selecting data-driven input variables. Reference [25] proposes an improved deep learning algorithm based on contractive auto-encoder empowered with restricted Boltzmann machine (RBM) for dimensionality reduction.

Nowadays researchers and practitioners are focusing on architecture design. In [26] author applied reinforcement learning and recurrent neural network for finding architecture. In [27] a genetic algorithm is used for designing complex CNN architecture via mutation operation. In [28], reinforcement learning based on Q-learning for searching best architecture designing is employed where the depth of architecture is decided by the user manually.

3 Convolution Neural Network

CNN have shown outstanding performances on multiple tasks relevant to image classification, segmentation, detection objects, Natural Language Processing, and video processing [29]. The architecture of CNN is based on several layers as shown in Fig. 1. These layers can be divided into two sections, where the first section is based on convolution layers, and the second is the pooling layers section containing fully connected layers [30]. Layer–wise stacking of linear and non-linear processing units allows one to learn it at multiple abstraction levels. Numerous improvements are made in the architecture and methodology of CNN for making it able to apply to large and complex problems [30]. Improvements in CNN are made at different aspects, like optimizing parameters, processing time, and design patterns. CNN performance can be improved by adding image transformation in training data and testing data to generate additional predictions [31].

The convolution layer is an initial layer after the input layer in the architecture. It consists of filters that are used to extract features from an input image. A feature map is calculated for each kernel. Only a tiny portion of the input, known as the receptive field, may be connected by the feature map units. A feature map is generally produced by moving filter on the input image and computing the dot product, followed by a non-linear activation function. Each feature map has the same weights (filters) for all units. The benefit of sharing weights is that it reduces the number of parameters required and allows you to identify the same feature independent of its location.

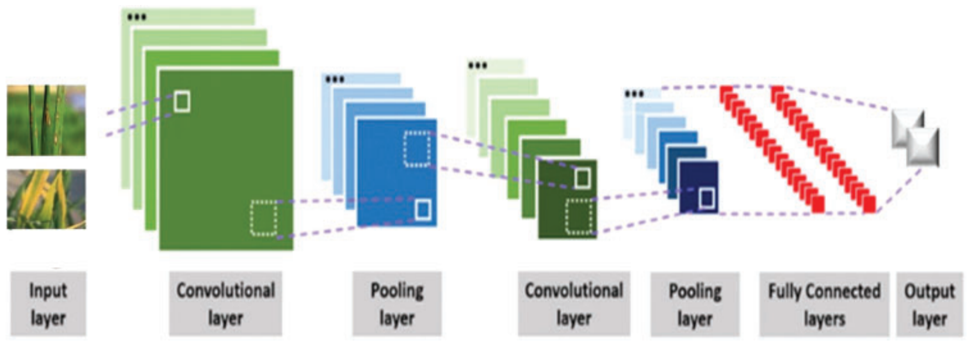

Figure 1: CNN architecture

Relu, Sigmoid, and Tanh are the most common activation function used in the convolution layer to activate the layer for feature extraction. Most of the researchers recommend applying Relu as it works almost perfectly in the training process. The output size is calculated by formula given in Eq. (1).

$$W = \left[\frac{H - F}{s}\right] + 1 \tag{1}$$

The pooling layer reduces previous feature maps' resolution. Pooling results in insensitivity to minor transformations and distortions. Pooling divides the inputs into disjoint areas of (R $*$ R) size and creates one output from each region type of Pooling layers can be either maximum or average. The output size of the pooling layer cab is calculated by Eq. (2).

$$P = \frac{\mathrm{W}}{\mathrm{R}} \tag{2}$$

CNNs' top layers are one or more fully connected layers, comparable to a feed-forward neural network, that seek to extract global characteristics from the inputs. These layers' units are linked to all concealed units in the preceding layer. The SoftMax classifier is the last layer, and it calculates the posterior probability of each class label over K classes, as given in Eq. (3).

$$Yi- = \frac{\mathrm{exp}(-\mathrm{zi})}{\sum_{\mathrm{j}-1}^{\mathrm{k}} \mathrm{Exp}(\mathrm{zj})} \tag{3}$$

4 Mutant Particle Swarm Optimization

The MUT-PSO is a variant of PSO, it focuses on controlled mutation. In MUT-PSO the Mutation operation is executed as described in the Algorithm explained in Tab. 4 after fitness evaluation by a fitness function $\mathrm{F(p)}_{\mathrm{A}\times 1}$. Tab. 1 shows the complete Algorithm of MUTPSO. Initially, the Mutation operation's calculation is done with equation

$$\bar{\mathrm{M}}_{\mathrm{o}}(\mathrm{i}) = \sum_{\mathrm{j}=1}^{\mathrm{A}} \frac{\mathrm{V}_{\mathrm{Ji}}}{\mathrm{A}} \tag{4}$$

Table 1: MUTPSO algorithms

Algorithm 1: MUT PSO algorithm

```
1. Initialize population P (p1, p2 . . . Pn)
2. Initialize velocity V (V1, V2 . . . Vn)
3. Determine fitness function
4. Select Local Best particle (Lbp)
5. Select Global best particle (Gbp)
6. Update velocity of each particle by
        $\mathrm{V_{im}}(n) = V_{im}(n-1) + a_1 * (Local\ Intelligence) + a_2 * (Global\ Intelligence\}$
Where,
$a_1 = rand(),\ a_2 = rand()$
        $LocalIntelligence = p_{im} - L_{bpim}(n-1)$
        $GlobalIntelligence = p_{im} - G_{bpim}(n-1)$
7. Update position of each particle:
8. Calculate Mutant particle
        $\bar{M}_o(i) = \sum_{j=1}^{A} \frac{V_{ji}}{A}$
        $p_{im}(n) = p_{im}(n-1) + \bar{M}_o(i) * rand()$
9. Calculate the fitness of muted particle
10. Update population
11. Check Criteria
12. Terminate
```

In above equation v_{Ji} is the velocity of jth particle at ith cycle and A is the number of particles. The mutation process is further initiated by

$$S_{vim} = 1 + \frac{1}{1 + e^{v_{im}(n)}} \tag{5}$$

It is further checked that

$$\text{if}(\text{rand}() < S_{vim})$$

$$P_M(i) = P_{LB}(i) + \bar{M}_o(i)\text{rand}()$$

5 Proposed Methodology

In this section, the proposed technique is discussed. The proposed techniques are divided into two parts training and testing. The training part is further divided into four layers which are the input layer, preprocessing layer, the prediction layer, and the performance evaluation layer. The proposed Algorithm takes the input of parameters mentioned in Tabs. 3–5. As per the proposed method, data is preprocessed initially, like resizing the images, de-noising, and normalization. Implementing denoising is considered as one of the significant part in this technique. Several denoising technique are proposed and evaluated in [31]. After successful completion of preprocessing, the data is sent to the prediction layer. In this layer, MUTPSO creates CNN architecture as per input data. In this research, we applied it for searching the optimum CNN architecture. The optimum architecture found by MUT-PSO is saved and in the testing phase, it is applied on paddy leaf disease image taken from Kaggle repository. Fig. 2 illustrates the whole process in detail.

Table 2: MUTPSO-CNN algorithms

Algorithm 2: MUTPSO-CNN algorithm

```
13. Input: Parameter
14. Pi.depth = rand(3, depth) ;
15. for j = 1 to Pi.depth do
16. if j == 1 then
17. list layers[j] ←Insert-Convolution layer (kmax, mapsmax) ;
18. else if j == Pi.depth then
19. list layers[j] ← addFully Connected (nout) ;
20. else if list layers[j-1].type == “fully-connected” then
21. list layers[j] ← addFully Connected Layer (nmax) ;
22. Else
23. layer type ← rand(1, 3) ;
24. if layer type == 1 then
25. list layers[j] ←insert-Convolution layer (kmax, mapsmax) ;
26. else if layer type == 2 then
27. list layers[j] ← Insert-Pooling layer() ;
28. Else
29. list layers[j] ← Insert-Pooling layer() ;
30. End
```

Table 3: MUT-PSO parameters

S. NO	MUT-PSO parametres	Values
1	Total MUT-PSO iteration	10
2	Size of swarm	20
3	Probability	0.5

Table 4: CNN architecture parameters

S. no	Parameters	Min. values	Max. values
1	Conv. layer output	3	100
2	Classes at FC	1	300
3	Filter size	3	7
4	Number of layers	3	20

Table 5: Parameters for CNN training

S. No	Parameters	Values
1	Evaluation of particles (epochs)	1
2	Global best (GB) epochs	100
3	Drop out	0.5
4	BN Layer	Y/N

The working structure of MUTPSO is consists of initialize population, initialize velocity, fitness evaluation of population, select Local Best particle (Lbp), select Global best particle (Gbp), update velocity of each particle, calculate the fitness of muted particle, update population and check criteria if it met than terminate the iteration else again evaluation the fitness. Tab. 2 shows complete algorithms along with mathematical equations of each phase.

Several evolutionary algorithms are proposed by previous researchers. Particle Swarm Optimization and Genetic Algorithms are considered as one of the best algorithms for optimization problems. MUT-PSO is a variant of PSO. It performs much better than other PSO variants even as compared to standard PSO. In this research structural parameters like convolutional layers, pooling layers, filter size, and the number of filters of CNN are optimized by MUT-PSO Automatically. Blocks which having excellent performance are kept and forwarded to the next iteration. These blocks are known as global best particles. Particle evaluation needs to restart at each iteration, but the proposed Algorithm keeps good blocks and discards if having poor performance. Tab. 2 shows complete Algorithm of creating optimum CNN architecture.

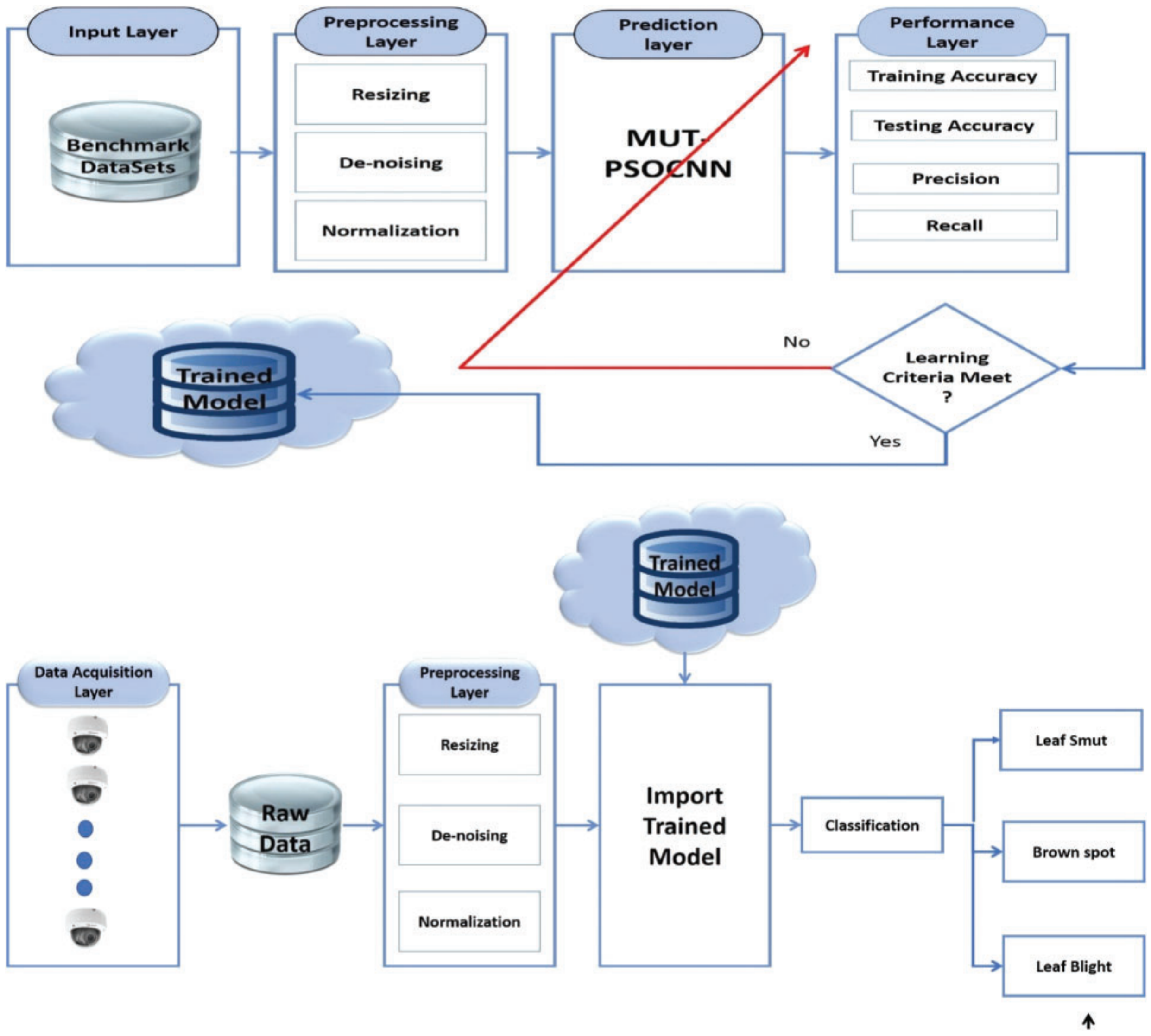

Figure 2: Proposed methodology

6 Proposed Algorithm

This section presents a brief description of the proposed Algorithm. The proposed Algorithm will create CNN architecture automatically with the help of MUT-PSO. This Algorithm is named as MUTPSO-CNN. This Algorithm will include initialization of swarm, fitness evaluation, calculating the difference of particle, calculates the velocity, and update velocity. Fig. 3 illustrates the complete process. Tab. 3 shows Algorithms 2 which brief complete MUTPSO-CNN, and Tab. 4 shows the initialization of MUTPSO-CNN algorithms for searching optimum CNN architecture based on input data.

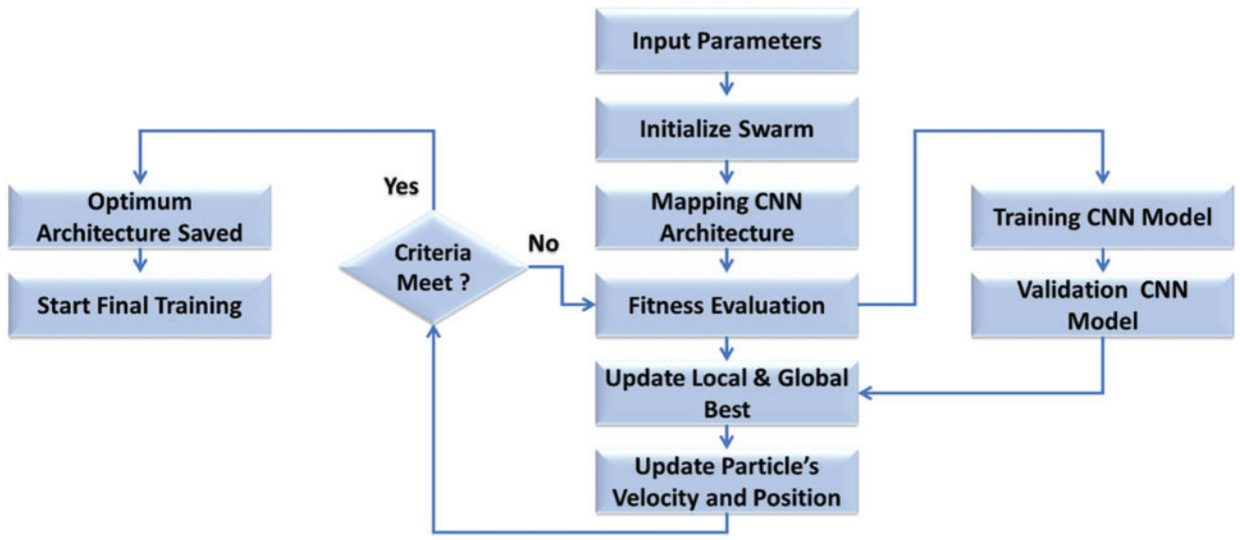

Figure 3: Flow of algorithm process

7 Results and Discussion

The models discovered by MUTPSO-CNN are tested and validated on a dataset of paddy leaf diseases available at Kaggle. The proposed Algorithm can achieve competitive results without the use of any data augmentation techniques or sophisticated architectures. Currently, the suggested MUTPSO-CNN models do not include feedback or parallel connections, and they are much simpler than existing models. Because the swarm was started with small networks, the proposed approach can locate smaller networks than peer competitors' models. Smaller networks also converge faster than larger networks. As a result, most population-based algorithms will be unable to identify good larger networks since they will be eliminated early in the process before they have a chance to outperform smaller networks. Tab. 6 presents suitable architecture created by MUTPSO-CNN.

Table 6: Architecture created by MUTPSO-CNN

S. no	Created architecture	Parameters of architecture	Dataset
1	Conv.layer	Conv.kernel value: 4 ∗ 4 output filter: 74	Paddy leaf diseases
	Conv.layer	Conv.kernel value: 6 ∗ 6 output filter: 183	
	Conv.layer	Conv.kernel value5 ∗ 5 output filter: 93	
	Conv.layer	Conv.kernel value: 5 ∗ 5 output filter: 184	
	Conv.layer	Conv.kernel value: 5 ∗ 6 output filter: 243	
	Average pooling layer	Pooling kernel value: 3 ∗ 3 stride 2 ∗ 2	
	Fully connected layer	Output neuron: 4	
2	Conv.layer	Conv.kernel value: 5 ∗ 5 output filter: 76	Paddy leaf images
	Conv.layer	Conv.kernel value: 4 ∗ 4 output filter: 183	
	Conv.layer	Conv.kernel value6 ∗ 6 output filter: 94	
	Conv.layer	Conv.kernel value: 6 ∗ 6 output filter: 187	
	Max. pooling layer	Pooling kernel value: 3 ∗ 3 stride 2 ∗ 2	
	Fully connected layer	Output neuron: 4	

This section presents the results based on our proposed Algorithm MUTPSO-CNN. The dataset used for this is Paddy leaf diseases. Firstly, the architecture created by our proposed is available in Tab. 7 at serial #1. Below is the performance of our proposed Algorithm on ten iterations. Validation of this research is based on convergence graphs and confusion matrix shown in Figs. 4–7 respectively. It is clear from simulation results that our proposed algorithms created CNN architecture performs better than manually crafted architectures. Tabs. 8 and 9 shows a comparative analysis of several techniques with the proposed technique.

Table 7: Performance of MUTPSO-CNN

S. no	Dataset	Iteration	Accuracy	Execution time
1	Paddy leaf disease	1	0.8928204	3947.3476192951202
		2	0.91474003	1383.8086533546448
		3	0.91583997	1097.3266532421112
		4	0.93842002	2521.9288318157196
		5	0.92763997	2911.4929699897766
		6	0.97202002	2333.1383199691772
		7	0.97241998	1051.578779220581
		8	0.97104001	1258.1493203639984
		9	0.96670002	1851.065199136734
		10	0.92389997	1465.7671558856964
2	Paddy leaf images	1	0.8528204	3647.3476192951202
		2	0.81474003	1583.8086533546448
		3	0.81583997	1297.3266532421112
		4	0.92842002	2421.9288318157196
		5	0.92765997	2611.4929699897766
		6	0.95205002	2533.1383199691772
		7	0.93249998	1551.578779220581
		8	0.91105001	1158.1493203639984
		9	0.79671002	1151.065199136734
		10	0.97359997	1465.7671558856964

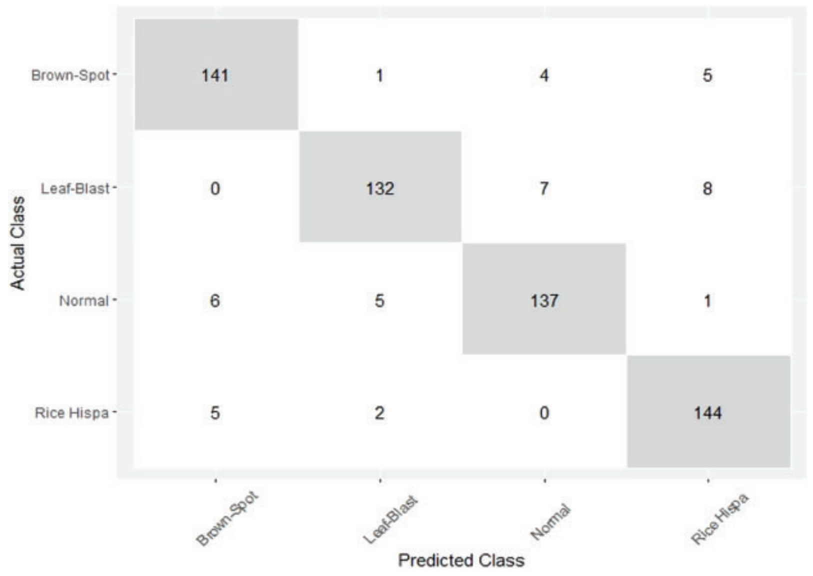

Figure 4: Confusion matrix of MUTPSO-CNN for paddy leaf diseases dataset

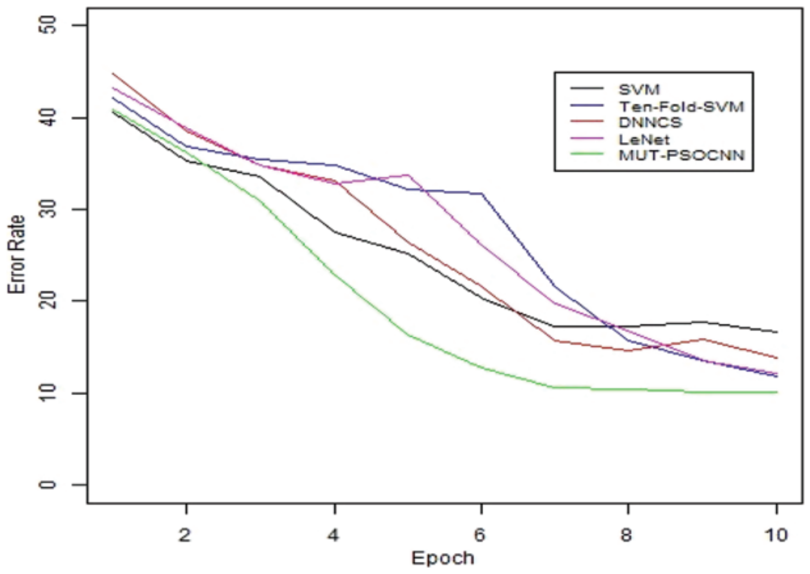

Figure 5: Convergence of MUTPSO-CNN for paddy leaf diseases dataset

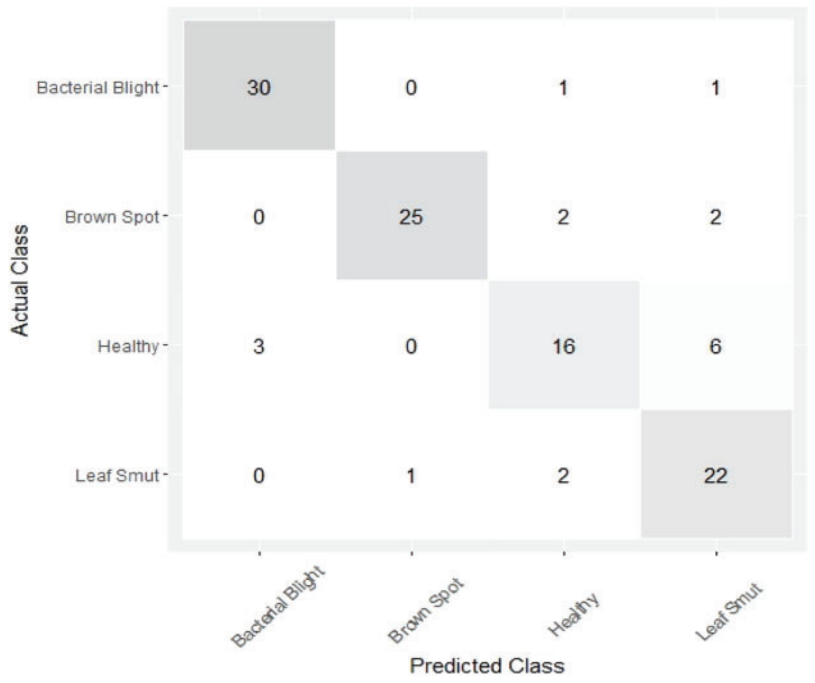

Figure 6: Confusion matrix of MUTPSO-CNN for paddy leaf images dataset

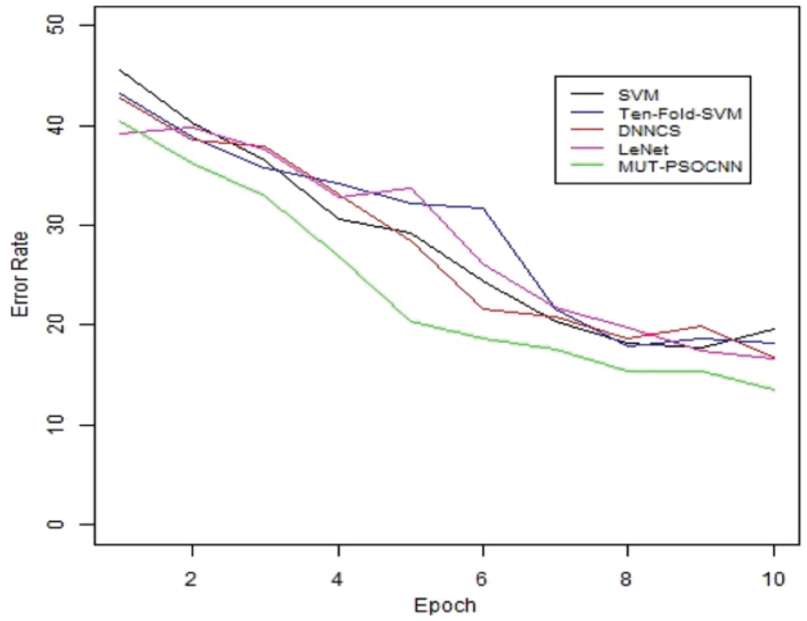

Figure 7: Convergence of MUTPSO-CNN for paddy leaf images dataset

Table 8: Comparison of MUTPSO-CNN with other techniques for paddy leaf diseases dataset

Methods	Training accuracy	Testing accuracy	Precision	Recall
LeNet	95.52	93.33	95.92	91.74
SVM	78.65	73.33	73.10	72.40
DNNCS	88.25	86.96	95.92	96.41
10-Fold SVM	91.20	88.57	87.46	88.27
MUTPSO-CNN	97.12	97.35	97.10	96.41

Table 9: Comparison of MUTPSO-CNN with other techniques for paddy leaf images dataset

Methods	Training accuracy	Testing accuracy	Precision	Recall
LeNet	93.45	91.33	95.92	91.74
SVM	76.58	73.33	73.10	72.40
DNNCS	88.24	86.96	95.92	96.41
10-Fold SVM	89.50	88.57	87.46	88.27
MUTPSO-CNN	96.12	93.35	97.10	96.41

8 Conclusion

Deduction in quality, as well as quantity of rice crop, is mainly due to widespread of diseases. Due to the large area of crops, it's quite difficult for farmers to identify the attacks of disease on crops, but farmers came into knowledge about the attack of diseases, it advances to very severe stage. Researchers have contributed towards the application of deep learning techniques but due to lack of expertise in domain knowledge as well as neural networks, progress was not up to mark. To resolve this limitation, we proposed an optimized CNN architecture generator based on MUT-PSO. The proposed approach generates the most optimum CNN architecture based on the input dataset. For evaluation of the proposed technique in this research activity, we considered two benchmark paddy disease datasets namely Paddy Leaf Diseases and Paddy Leaf Images dataset. Both datasets are publicly available on Kaggle. Experimentation results show that MUTPSO-CNN finds optimum CNN architecture for Paddy leaf classification. CNN architecture created by the proposed Algorithm shows remarkable performance as compared to standard CNN architecture available. We are currently working on integrating multiple optimization approaches to get a more accurate and robust approach with less space and time complexity. In the near future, we plan to extend out proposed approach for many complex datasets and provide more significant and extensible solutions based on our existing methodology.

Acknowledgement: The authors would like to thank the Research Management Office (RMC), Universiti Tun Hussein Onn Malaysia and Ministry of Higher Education, Malaysia for supporting this research under Multi-Disciplinary Research (MDR) Grant Vot H483.

Funding Statement: The authors received funding source for this research activity under Multi-Disciplinary Research (MDR) Grant Vot H483 from Research Management Centre (RMC) office, Universiti Tun Hussein Onn Malaysia (UTHM).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Ramesh and D. Vydeki, "Recognition and classification of paddy leaf diseases using optimized deep neural network with jaya algorithm," *Information Processing in Agriculture*, vol. 7, no. 2, pp. 249–260, 2020.

[2] N. Senan, M. Aamir, R. Ibrahim, N. Taujuddin and W. Muda, "An efficient convolutional neural network for paddy leaf disease and pest classification," *International Journal Advanced. Computer Science Application*, vol. 11, no. 7, pp. 116–122, 2020.

[3] R. Narmadha and G. Arulvadivu, "Detection and measurement of paddy leaf disease symptoms using image processing," in *Proc. 2017 Int. Conf. on Computer Communication and Informatics (ICCCI)*, India, pp. 1–4, 2017.

[4] S. Pavithra, A. Priyadharshini, V. Praveena and T. Monika, "Paddy leaf disease detection using SVM classifier," *International Journal of Communication and Computer Technologies*, vol. 3, no. 1, pp. 16–20, 2015.

[5] M. T. Sadiq, X. Yu, Z. Yuan, F. Zeming, A. U. Rehman, I. Ullah *et al.,* "Motor imagery EEG signals decoding by multivariate empirical wavelet transform-based framework for robust brain–computer interfaces," *IEEE Access*, vol. 7, no. 1, pp. 171431–171451, 2019.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *The 3rd International Conference on Learning Representations (ICLR2015)*, https://arxiv.org/abs/1409.1556. 2015.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, USA, vol. 6, no. 2, pp. 770–778, 2016.

[9] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, USA, pp. 5353–5360, 2015.

[10] A. X. Zheng and M. Bilenko, "Lazy paired hyper-parameter tuning," in *Twenty-Third Int. Joint Conf. on Artificial Intelligence Bejing*, China, pp. 1–8, 2013.

[11] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[12] S. Phadikar and J. Sil, "Rice disease identification using pattern recognition techniques," in *IEEE Proc. of 11th Int. Conf. on Computer and Information Technology*, China, vol. 1, no. 2, pp. 420–423, 2008.

[13] T. Islam, M. Sah, S. Baral and R. R. Choudhury, "A faster technique on rice disease detection using image processing of affected area in agro-field," in *IEEE Explore Compliant of the 2nd Int. Conf. on Inventive Communication and Computational Technologies*, USA, pp. 62–66, 2018.

[14] N. Mangla, P. B. Raj, S. G. Hedge and R. Pooja, "Paddy leaf disease detection using image processing and machine learning," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 7, no. 2, pp. 97–99, 2019.

[15] T. Verma and S. Dubey, "Paddy disease recognition using image processing and radial basis function network," *Indian Journal of Science and Technology*, vol. 10, no. 46, pp. 1–7, 2017.

[16] V. K. Shrivastava, M. K. Pradhan, S. Minz and M. P. Thakur, "Rice plant disease classification using transfer learning of deep convolution neural network," in *The Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Joint Int. Workshop on Earth Observations for Agricultural Monitoring*, China, 2019.

[17] F. Wahid, M. Fayaz, A. Aljarbouh, M. Mir and M. Aamir, “Energy consumption optimization and user comfort maximization in smart buildings using a hybrid of the firefly and genetic algorithms,” *Energies*, vol. 13, no. 17, pp. 43–63, 2020.
[18] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 124–132, 2012.
[19] F. Hutter, H. H. Hoos and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in *Proc. of the Int. conf. on learning and intelligent optimization*, Springer, Kalamata, Greece, pp. 507–523, 2011.
[20] J. Snoek, H. Larochelle and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” *Advances in Neural Information Processing Systems*, vol. 25, no.2, pp. 178–182, 2012.
[21] C. Ma and Y. Li, “Improving forecasting accuracy of annual runoff time series using RBFN based on EEMD decomposition,” *DEStech Transactions on Engineering and Technology Research*, vol. 16, no. 2, pp. 216–221, 2016.
[22] R. Taormina and K. W. Chau, “Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and extreme learning machines,” *Journal of Hydrology*, vol. 529, no. 15, pp. 1617–1632, 2015.
[23] J. Zhang and K. -W. Chau, “Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization,” *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 840–858, 2009.
[24] G. Silva, T. Valente, A. Silva Paiva and M. Gattass, “Convolutional neural network-based PSO for lung nodule false positive reduction on CT images,” *Computer Methods and Programs in Biomedicine*, vol. 162, no. 01, pp. 109–118, 2018.
[25] M. Aamir, N. M. Nawi, F. Wahid, M. S. H. Zada and M. Z. Rehman, “Hybrid contractive autoencoder with restricted Boltzmann machine for multiclass classification,” *Arabian Journal for Science and Engineering*, vol. 1, no. 1, pp. 1–15, 2021.
[26] A. Onat, H. Kita and Y. Nishikawa, “Reinforcement learning of dynamic behavior by using recurrent neural networks,” *Artificial Life and Robotics*, vol. 1, no. 3, pp. 117–121, 1997.
[27] T. Fatyanosa and A. Masayoshi, “Effects of the number of hyperparameters on the performance of GA-CNN,” in *Proc. of the IEEE/ACM Int. Conf. on Big Data Computing, Applications and Technologies (BDCAT)*, New York United States, pp. 144–153, 2020.
[28] B. Baker, O. Gupta, N. Naik and R. Raskar, “Designing neural network architectures using reinforcement learning,” *in Proc. of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
[29] W. Chen and X. Yang, “Convolutional neural networks for image classification,” in *Proc. of the Int. Conf. on Advanced Systems and Electric Technologies IEEE*, USA, pp. 1–7, 2018.
[30] R. A. Naqvi, M. Arsalan, A. Rehman and A. Paul, “Deep learning-based drivers emotion classification system in time series data for remote applications,” *Remote Sensing*, vol. 12, no. 3, pp. 587–619, 2020.
[31] S. Shichijo, S. Nomura, K. Aoyama and Y. Nishikawa, “Application of convolutional neural networks in the diagnosis of helicobacter pylori infection based on endoscopic images,” *EBioMedicine*, vol. 25, no. 1, pp. 106–111, 2017.