Leveraging Deep Learning and Farmland Fertility Algorithm for Automated Rice Pest Detection and Classification Model

Hussain. A[1*], and Balaji Srikaanth. P[1]

[1] Department of Networking and Communications, College of Engineering and Technology (CET),
SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.
[e-mail: ha2514@srmist.edu.in, balajis7@srmist.edu.in]
[*]Corresponding author: Hussain. A

Received November 22, 2023; revised Feburary 24, 2024; accepted March 17, 2024;
published April 30, 2024

Abstract

Rice pest identification is essential in modern agriculture for the health of rice crops. As global rice consumption rises, yields and quality must be maintained. Various methodologies were employed to identify pests, encompassing sensor-based technologies, deep learning, and remote sensing models. Visual inspection by professionals and farmers remains essential, but integrating technology such as satellites, IoT-based sensors, and drones enhances efficiency and accuracy. A computer vision system processes images to detect pests automatically. It gives real-time data for proactive and targeted pest management. With this motive in mind, this research provides a novel farmland fertility algorithm with a deep learning-based automated rice pest detection and classification (FFADL-ARPDC) technique. The FFADL-ARPDC approach classifies rice pests from rice plant images. Before processing, FFADL-ARPDC removes noise and enhances contrast using bilateral filtering (BF). Additionally, rice crop images are processed using the NASNetLarge deep learning architecture to extract image features. The FFA is used for hyperparameter tweaking to optimise the model performance of the NASNetLarge, which aids in enhancing classification performance. Using an Elman recurrent neural network (ERNN), the model accurately categorises 14 types of pests. The FFADL-ARPDC approach is thoroughly evaluated using a benchmark dataset available in the public repository. With an accuracy of 97.58, the FFADL-ARPDC model exceeds existing pest detection methods.

Keywords: Rice plant, Pest detection, Agriculture, Deep learning, Farmland fertility algorithm.

http://doi.org/10.3837/tiis.2024.04.008 **ISSN : 1976-7277**

1. Introduction

Rice is the most important world food crop for human consumption and remains the keystone of global food safety. The growth cycle of rice is followed by the incidence of various pests, which result in severe yield losses [1]. Correct detection of rice pests helps early preventive processes for avoiding financial losses [2]. Earlier rice pest observing techniques mostly utilize trapping lights for capturing pests that are further counted, retrieved and detected the following day. The conventional approach to detecting rice disease includes visual analysis by plant protection experts [3]. But, this technique is time-taking and largely depends on subjective assessment, creating it hard to quickly control and avoid rice diseases. With the fast development of software and hardware technologies, the application of artificial intelligence (AI), image identification, big data, and deep learning (DL) have been more frequent in the domain of agriculture, especially in crop disease detection and identification. If diseases can affect rice plants, their morphological characteristics and physiological structures are modified leading to signs such as decay, deformation, and leaf discoloration [4]. This laborious method is implemented manually in terms of developing the recent pest condition and gives area for misdiagnosis and delays in diagnosing period [5]. With the gradual growth of DL and ML, enhancing researcher counts are an attempt to increase these new methods in the domain of pest detection.

Conventionally, the recognition and classification of rice pests and diseases depend on manual visual analysis that is time taking, inaccuracy, and ineffective [6]. These inefficient standard manual analysis techniques prevent adequate rice production from attaining social requirements, requiring the emergence of new identification methods [7]. Developments in computer technology are provided to increase conventional image-processing approaches, comprising detection, feature extraction, reduced dimensionality, and image pre-processing [8]. These techniques are employed for agricultural images for different resolutions, for instance, classification of crop pest and disease sites and varieties. DL-based target detection and classification algorithms are newly developed in traditional investigation, modifying classical image detection methods [9]. These techniques enable machines to adjustably learn image features with no manual feature extraction, therefore, allowing proficient implementation of identification and classification tasks. Deep convolutional networks can be effectually enhanced migration among rice diseases and pests [10].

This article offers the design of a new farmland fertility algorithm with deep learning-based automated rice pest detection and classification (FFADL-ARPDC) technique. The FFADL-ARPDC technique performs bilateral filtering (BF)-based noise removal and contrast enhancement as a pre-processing stage. In addition, the NASNetLarge-DL architecture is employed to extract meaningful features from rice crop images. To optimize the model performance of the NASNetLarge, the FFA is utilized for the hyperparameter tuning process, which helps in improving the classification performance. Finally, the Elman recurrent neural network (ERNN) is used for pest classification, enabling the model to recognize 14 diverse pest species accurately. The FFADL-ARPDC approach is thoroughly evaluated using IP102-dataset from the Kaggle repository.

2. Related Works

Gong et al. [11] presented a method for automatic rice pest classification and detection that depends on FCNs and chooses 10 rice pests for experiments. Primary, the authors present a novel encoding–decoding from the FCN and a sequence of sub-networks linked by jump paths to integrate long jumps and shortcut connections for correct and fine-grained insect boundary recognition. Secondary, the network also combines a CRF module for insect contour refinement and boundary localization; at last, a new DenseNet design that establishes an attention mechanism (ECA) has been presented to concentrate on extracting insect edge features for effectual rice pest classification. Rahman et al. [12] constructed a DL-based model for identifying pests and diseases in the images of rice plants. Primarily, state-of-the-art large-scale construction like InceptionV3 and VGG16 are fine-tuned and adopted for identifying and recognizing diseases and pests of rice plants. Later, dual-phase small CNN approaches are presented, and related with the modern memory effectual CNN methods like SqueezeNet, MobileNet, and NasNet Mobile.

Thenmozhi and Reddy [13] introduced a DCNN method for classifying insect species. TL is enforced for fine-tuning the pre-trained techniques. The data augmenting methods like rotation, translation, reflection, and scaling are also enforced for preventing the network from the issue of overfitting. The presented method is related and analyzed with pre-trained DL constructions like GoogLeNet, AlexNet, VGGNet, and ResNet for classifying insects. The presented method is examined by employing the NBAIR dataset. Prasath and Akila [14] introduced an approach called Yolov3 for recognizing pests in plants and the conceded neurons are maximized by the method called Adaptive Energy-based Harris Hawks Optimizer (AE-HHO). A deep feature-extracting process is employed with VGG16 and ResNet50 methods. The classifying process is accomplished by the approach called Weight Optimized DNN (WO-DNN). This WO-DNN approach of the weight feature is maximized by employing the AE-HHO method.

V. Malathi and M. P. Gopinath [15] proposed a model for the classification of Ten insect types in the rice crop that were identified using deep convolutional neural networks (DCNN). Different DCNN architectures were used to create the models, with the results being interpreted based on their accuracy and performance. By adjusting the ResNet-50 model's hyperparameters and layers, we successfully applied the transfer learning strategy to the pest dataset. ResNet-50 was fine-tuned to outperform other models, and it now has an accuracy of 95.012%. Data augmentation was discussed in the research to expand the dataset, and pre-trained architecture models for pest recognition were also considered. By comparing the anticipated and observed models, we determined the loss function and applied it using ADAM optimization with fine-tuned parameters. Based on findings from other articles, the authors of this one underlined the usefulness of deep learning models and transfer learning for categorizing insect pests.

In [16], an IoT-based classifying and recognizing pest model is presented. Primarily, the IoT sensors are employed for accomplishing the object recognition which is achieved by the model Yolov3. Additionally, the recognized imageries are given as input to the CNN method, where the deep aspects are taken and sent as input to Convolution Neural LSTM (CNLSTM) method. Few hyperparameters are maximally tuned by Adaptive Honey Badger Algorithm (AHBA) technique. Ayan et al. [17] altered and re-trained 7 diverse pre-training CNN methods such as Xception, VGG-16, SqueezeNet, ResNet50, InceptionV3, MobileNet, and VGG-19 by employing fitting fine-tuning policies and Transfer Learning (TL) models on open dataset D0. Then, the CNN approaches such as MobileNet, Inception-V3, and Xception are collected through the Sum of Maximum probability policy, the method was called the SMPEnsemble. Later, these methods are gathered by implementing weighted voting. H. Peng et al. [18] have created HQIP102, a massive dataset on insect pests that includes 47,393 photos over 102 different pest classes on eight different crop types. The HQIP102 dataset was obtained by meticulously filtering the IP102 dataset to remove incorrect pest classifications and images lacking pest subjects. To enhance DenseNet's representation capacity, the MADN framework was proposed. It presents the Selective Kernel unit to modify the size of the field of reception adaptively, the Representative Batch Normalisation module to ensure the features adhere to a constant distribution throughout, and the ACON activation function to activate neurons systematically. Ensemble learning is what makes up the MADN model. Several metrics assessed MADN's efficacy, including precision, recall, accuracy, F1Score. The MADN model outperformed competing models in accuracy and F1Score, although using fewer parameters than ResNet-101 and using more GPU RAM and a more extended training period. The summary of the related study is shown in **Table 1**.

Table 1. Summary of the related study

Authors	**Methodology Used**	**Advantage**	**Disadvantage**
Gong et al.	FCNs, CRF module, DenseNet with ECA	Effective edge characteristic extraction using DenseNet-ECA, improved contour refinement using CRF, and accurate pest identification using FCNs with novel encoding-decoding.	Complex architecture necessitates plenty of computation, which might lengthen the training process.
Rahman et al.	InceptionV3, VGG16, dual-phase CNN, SqueezeNet, MobileNet, NasNet Mobile	Use of sophisticated models (InceptionV3, VGG16) for pest and disease detection, study of memory-efficient CNN approaches (SqueezeNet, MobileNet, NasNet Mobile).	Difficulty in balancing accuracy and processing economy throughout the model selection and fine-tuning processes.
Thenmozhi and Reddy	TL, Data augmentation, GoogLeNet, AlexNet, VGGNet, ResNet	Evaluation of different pre-trained architectures (GoogLeNet, AlexNet, VGGNet, ResNet), usage of TL for insect classification, and prevention of overfitting by data augmentation.	Large amounts of annotated data are required for TL, and its success is contingent on the availability and quality of transferable characteristics.
Prasath and Akila	Yolov3, AE-HHO, VGG16, ResNet50, WO-DNN	Effective pest detection using Yolov3 optimised with AE-HHO; utilising VGG16 and ResNet50 for deep feature extraction; and using WO-DNN for classification.	Yolov3's complicated design may necessitate a lot of computing time, and fine-tuning the parameters of the AE-HHO may be necessary for optimal performance.
V. Malathi and M. P. Gopinath	DCNN, Transfer Learning, ResNet-50	Transfer learning on ResNet-50 was used to successfully create DCNN models for insect categorization, resulting in a high accuracy of 95.012%.	Dependence on the quality and variety of the dataset, possible bias from the transfer learning process.

Kathole et al.	IoT, Yolov3, CNN, CNLSTM, AHBA	The capacity of AHBA to efficiently adjust constrained hyperparameters, the incorporation of IoT and Yolov3 for pest detection, and the application of the CNN-CNLSTM technique are all noteworthy.	Need for an Internet of Things backbone, the possibility of connectivity problems impacting real-time identification, and the difficulty of fine-tuning AHBA settings.
Ayan et al.	Various pre-trained CNNs, Transfer Learning, SMPEnsemble	Consolidation using the SMPEnsemble method, followed by TL-enhanced retraining of multiple previously-trained CNNs.	Possible difficulty in achieving harmony amongst CNN outputs; additional computing burden introduced by handling numerous models simultaneously during inference.
H. Peng et al.	HQIP102 dataset, MADN architecture, ResNet-101, GoogLeNet, MobileNet V2	The MADN architecture outperforms ResNet-101 with fewer parameters in terms of accuracy (10.48%) and F1Score (10.56%) thanks to the inclusion of the HQIP102 dataset.	Training times and GPU memory requirements are increased by MADN's use of an ensemble learning technique, as well as by the possible difficulty of implementing the ACON activation function and Representative Batch Normalisation.

2.1 Problem Statement

i. Several approaches have large computing needs because their designs are complicated.
ii. Model performance, especially when using transfer learning methods, is highly dependent on the quality and variety of the datasets used in such methods.
iii. Research shows that the employment of a wide variety of models and optimisation approaches presents difficulties in model selection and parameter tweaking.
iv. Higher GPU memory and extended training durations are the consequences of using ensemble learning techniques.
v. Precise parameter adjustment is required when utilising complicated optimisation techniques.

3. The Proposed Model

In this study, we focus on the design and development of automated rice pest detection and classification using the FFADL-ARPDC technique. The major intention of the FFADL-ARPDC model is to exploit the DL model with a hyperparameter tuning strategy for the identification and categorization of rice pests into different classes. To accomplish this, the FFADL-ARPDC method encompasses pre-processing, NASNetLarge feature extraction, FFA-based hyperparameter tuning, and ERNN-based classification.

3.1 Image Pre-processing

For pre-processing the input images, BF-based noise removal and Contrast Limited Adaptive Histogram Equalization (CLAHE) based contrast enhancement are applied. BF is established for smooth images while preserving the fine details and edges in the image [19]. It takes into account both the intensity similarity and spatial proximity between pixels while using the filter, which makes it efficient in decreasing noise without blurring edges. **Fig. 1** depicts the entire flow of the FFADL-ARPDC algorithm.

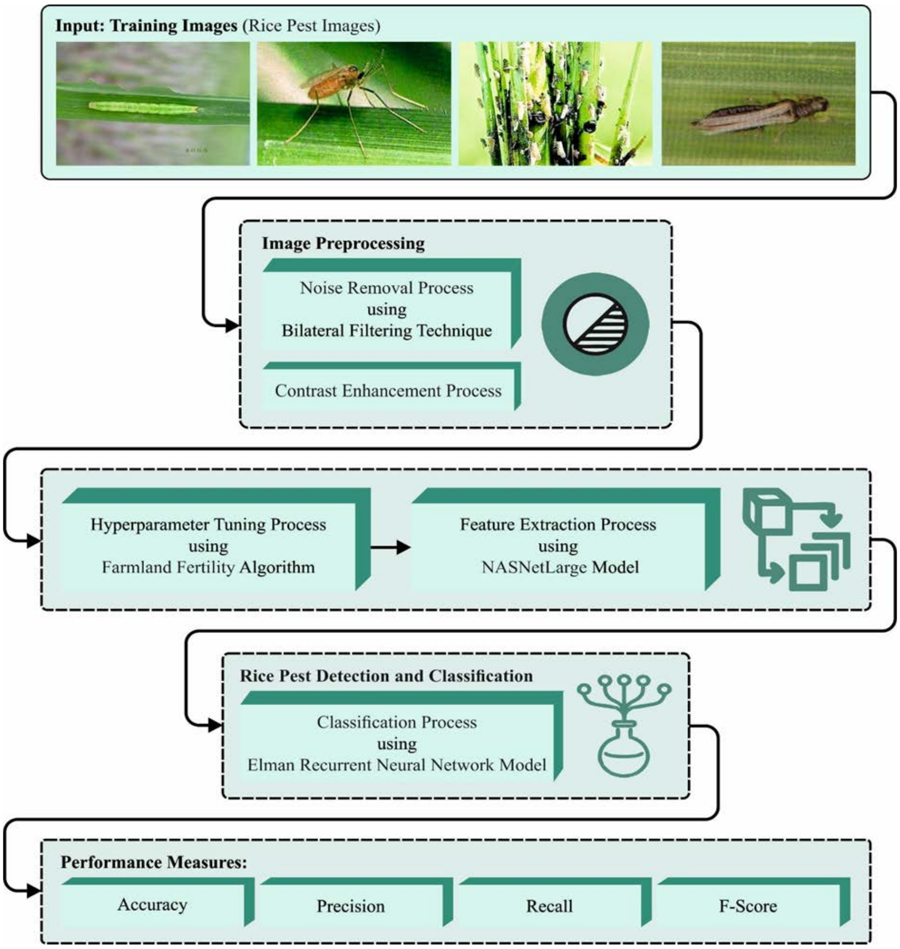
Input: Training Images (Rice Pest Images)
Image Preprocessing
Noise Removal Process using Bilateral Filtering Technique
Contrast Enhancement Process
Hyperparameter Tuning Process using Farmland Fertility Algorithm
Feature Extraction Process using NASNetLarge Model
Rice Pest Detection and Classification
Classification Process using Elman Recurrent Neural Network Model
Performance Measures:
Accuracy
Precision
Recall
F-Score

Fig. 1. Overall flow of FFADL-ARPDC algorithm.

The fundamental notion behind BF is to calculate the weighted average of adjacent pixel value, where the weight is defined by the intensity difference and the spatial distance between its neighbours and the center pixel. Following, CLAHE is an image processing method used for enhancing the image contrast while avoiding the challenges of over-amplifying noise in uniform regions. It is an extension of the classical Histogram Equalization (HE) technique. CLAHE addresses the limitation of classical HE by implementing the equalization process on smaller, overlapping image regions, called blocks or tiles. This local strategy assists in avoiding artifacts excessive and intensification of noise. Every tile's histogram is equalized, but there is a restriction imposed on how much the histogram can be stretched. This constraint is the "contrast limiting" aspect of CLAHE. When the count of the histogram bin exceeds the limit, then the excess values are redistributed to nearby bins, which prevents drastic amplification.

The CLAHE approach functions on a lesser area of images which is called a tile [20]. The contrast restricted in tile can be fine-tuned so that the histogram created from that region matches the given histogram shape. The neighboring tile has been connected using bilinear interpolation. This method is done such that the outcomes of the integration of the tiles look smoother.

$$\beta = \frac{M}{N}\left(1 + \frac{\alpha}{100}(S_{\max} - 1)\right) \tag{1}$$

In Eq. (1), β implied the limit value (clip limit), S_{max} stands for the maximal suitable slope. The clip factor α depicts the addition of histogram limit values from 1 and 100. M variable represents the region size, $and\ N$ denotes the grey level value (256).

3.2 Feature Extraction using NASNetLarge

In this work, the NASNetLarge deep learning architecture is employed to extract meaningful features from rice crop images. Neural Architecture Search Network (NASNet) is a family of NN architecture that was created by an automatic method named called NAS architecture [21]. The objective of NAS is to automatically discover the optimum network architecture for certain tasks, which relieves human engineers from the labour-intensive task of manually developing architecture. NASNetLarge is the neural architecture that was designed by the NAS architecture. It is well-known for its performance on image classification tasks and the ability and complexity to accomplish higher accuracy on different benchmark data. NASNetLarge was discovered by an automatic process that searched through a large space of possible architectures Rather than manually designing the architecture. This process optimizes both model accuracy and computation efficacy. It comprises repeating cells, each having a stack of operations that are connected. They are organized in a hierarchy to generate the full network. Also, it consists of a skip connection, otherwise called a residual connection. This connection helps in gradient flow during training and could facilitate the learning of complicated features. Typically, the last layer of the NASNetLarge includes global average pooling that decreases the spatial dimension of the mapping feature and assists in creating a fixed-size representation for classification. NASNetLarge exploits different pooling functions including max and avg pooling for capturing various stages of data from the mapping feature.

3.3 Hyperparameter Tuning using FFA

The FFA is exploited for the NASNetLarge method to adjust the hyperparameter process. FFA is based on the natural fertility of the farmland and attempts to enhance every solution through the separation of farmland and the desirable productivity of internal and external memory [22]. The FFA method has the best exploration and convergence strategy due to external memory (global solution and) internal memory (local solution) in optimizer issues including data clustering. Furthermore, the FFA technique exploits a global solution that ensures model convergence. Thus, this algorithm doesn't need to be changed dramatically unlike the approaches studied in the background. Also, the outcomes prove that the solution to the problems of data clustering is of improved quality than other techniques. However, the shortcoming of the FFA model in data clustering problems is that similar to metaheuristic systems, by enhancing the dimension of the problems, the algorithm might have lost its efficacy to a certain degree, thus the parameter of the FFA model should be more carefully selected. The step by step and formulation of the FFA model is given below. At first, an initial population was generated using the following expression:

$$N = k * n \tag{2}$$

Where k defines the amount of sections of space or land, and n shows the amount of performances existing in every region of farmland. A single section is regarded to define k's value. The optimum values are considered as k within [2,8] [3]. All the areas of farmland's quality were attained by the average of the solution existing in every region of farmland. At first, Eq. (3) allows solutions to all the areas.

$$Section_s = x(aj), a = n * (s - 1): n * s\ s = 1,2, \dots k, j = 1,2, \dots D \tag{3}$$

Where s denotes the amount of sections and $j = 1,2, \dots, D$ shows the dimension of parameter $x(aj)$ that each solution in the searching space. This formula separates present solutions of every segment in a simple manner to separately evaluate the section average. Eq. (4) defines the quality of every section after segmentation.

$$Fit\_Section_s = Mean(all\ Fit(x_{ji})\ in\ Section_s)\ s = 1.2 \dots k, i = 1.2 \dots n \tag{4}$$

Where $Fit\_Section_s$ defines the quality of outreached section of the solution. The amount of better local memory to Eq. (6) and the amount of optimum memory defines the local as well as global memory of all the sections based on Eq. (5).

$$M\_Global = round(t * N)\ 0.1 < t < 1 \tag{5}$$

$$M\_Local = round(t * n)\ 0.1 < t < 1 \tag{6}$$

Where, M_Global and M_Local indicate the solution counts in global as well as local memory, correspondingly. Furthermore, local and international memories can be upgraded. According to Eqs. (7) and (8), each solution has the worse section of farmland that is hybridized with the global memory solution.

$$h = \alpha * rand(-1,1) \tag{7}$$

$$X_{new} = h * (X_{ij} - X_{M\_Global}) + X_{ij} \tag{8}$$

In Eqs. (7) and (8), α refers to a random integer within [0,1]. X_{M_Global} shows the randomly generated solution amongst the available solutions in the global memory, X_{ij} denotes the selected solution in the worst section for making some modifications, and also h indicates the decimal number. Based on Eqs. (9) and (10), the available solutions in the other sections also change.

$$h = \beta * rand(-1,1) \tag{9}$$

$$X_{new} = h * (X_{ij} - X_{uj}) + X_{ij} \tag{10}$$

In the Equation, X_{uj} denotes the random path selected between each solution in the section, and β indicates the number within [0,1]. X_{ij} shows the performance in the section chosen to execute the changes, and h denotes a decimal number evaluated using Eq. (10). The combination of the solution considered with $Best_{Global}$ or $Best_{Local}$ is defined using Eq. (11) for upgrading outcomes based on local as well as global memory.

$$H = \begin{cases} X_{new} = X_{ij} + \omega_1 * \left(X_{ij} - Best_{Local}(b)\right): & Q > rand(0,1) \\ X_{new} = X_{ij} + rand(0,1) * \left(X_{ij} - Best_{Local}(b)\right): & else \end{cases} \tag{11}$$

Where $the\ Q$ value ranges within [0,1] that defines the amount of combination of the solutions with $Best_{Global}$. ω_1 is a parameter associated with FFA that is an integer. Initially, the algorithm must be indicated, and based on the iteration its amount reduces as follows.

$$\omega_1 = \omega_1 * R_v\ 0 < R_v < 1 \tag{12}$$

At the final stage, the termination condition was explored. If the final condition is present, then the algorithm stops. Or else, the process continued the task until the ending criteria were met. The FFA method derives a FF to obtain better performance of classification. It defines a positive integer to characterize the better solution of the solution candidate. The reduction of the classifier error rate was assumed as the FF.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No. of\ misclassified\ instances}{Total\ No. of\ instances} * 100 \quad (13)$$

3.4 Pest Classification using ERNN Model

In the final stage, the ERNN model can be applied to the recognition and classification of pest classes. Elman in 1990 [23] established the ERNN as a simple RNN. The presented method has the benefits of faster convergence, accurate mapping capability, consuming time series, and nonlinear prediction capabilities. ENN combined with dissimilar areas to its resolve. In this study, the outcome of hidden layers (HL) was allowable to feedback on itself with butter layer is called a recurrent layer (RL). Single RL neurons with the constant weight of one connect each hidden neuron. Consequently, the number of recurrent neurons can be equivalent to hidden neuron counts. All the layers have different neurons that transmit data from one layer to another by calculating a non-linear function of the weighted sum of input. Fig. 2 depicts the infrastructure of ERNN. The multi-input ERNN method was illustrated where the quantity of neurons in the input layers is m and in the HL is n and single outcome unit.

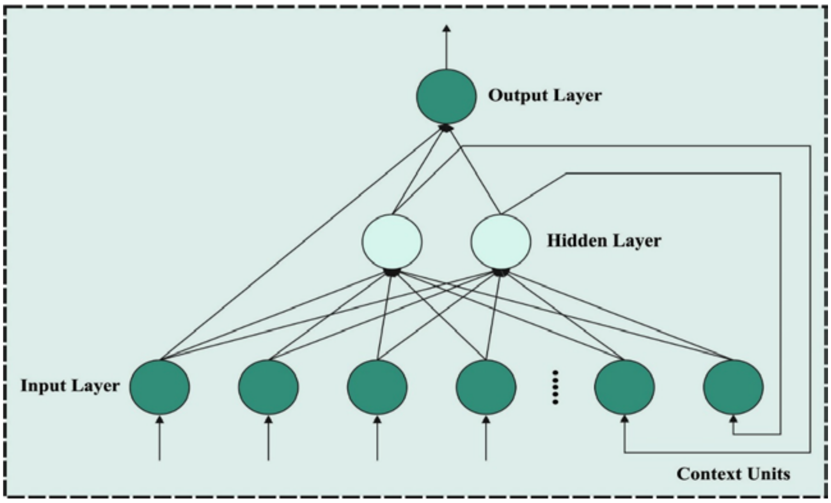

Fig. 2. Infrastructure of ERNN.

Consider $x_{it}(i = 1,2,,m)$ as a collection of input vectors of neurons at t time, y_{t+1} represents the outcomes of the network at $t + 1$time , $z_{jt}(j = 1,2, \ldots , n)$ shows the outcomes of HL neurons at t time, and $(j = 1,2, \ldots , n)$ determines the RL neurons. w_{ij} shows the weighted to link the i^{th} nodes in the input layer neuron to j^{th} nodes from the HLs. c_j, indicates the weight that correspondingly connects the j^{th} nodes in the HL neuron to nodes under the output and RL. The HL phase is given in the following: the input of all the neurons from the HL is shown below:

$$(k) = \sum_{i=1}^{n} w_{ij}\, x(k-1) + \sum_{j=1}^{m} c_j\, u_{jt}(k) \quad (14)$$

$$u_{jt}(k) = z_{jt}(k-1), i = 1,2, \ldots , n, j = 1,2, \ldots , m.$$

The outcome of hidden neuron is given as follows:

$$z_{jt}(k) = f_H\left(net_{jt}(k)\right) = \left(\sum_{i=1}^{n} w_{ij}\,(k) + \sum_{j=1}^{m} c_j\, u_j(k)\right) \tag{15}$$

The sigmoid function in HLs has been selected as an activation function: $f_H(x) = 1/(1 + e^{-\chi})$. The outcomes of HL are shown in the following:

$$y_{t+1}(k) = f_T\left(\sum_{j=1} v_j\,(k)\right) \tag{16}$$

In Eq. (16), (x) denotes the identity map as an activation function.

4. Experimental Validation

The performance validation of the FFADL-ARPDC approach is tested on IP102-dataset from the Kaggle repository [24]. Fig. 3 illustrates the sample images. Table 2 defines a detailed description of the IP102 dataset.

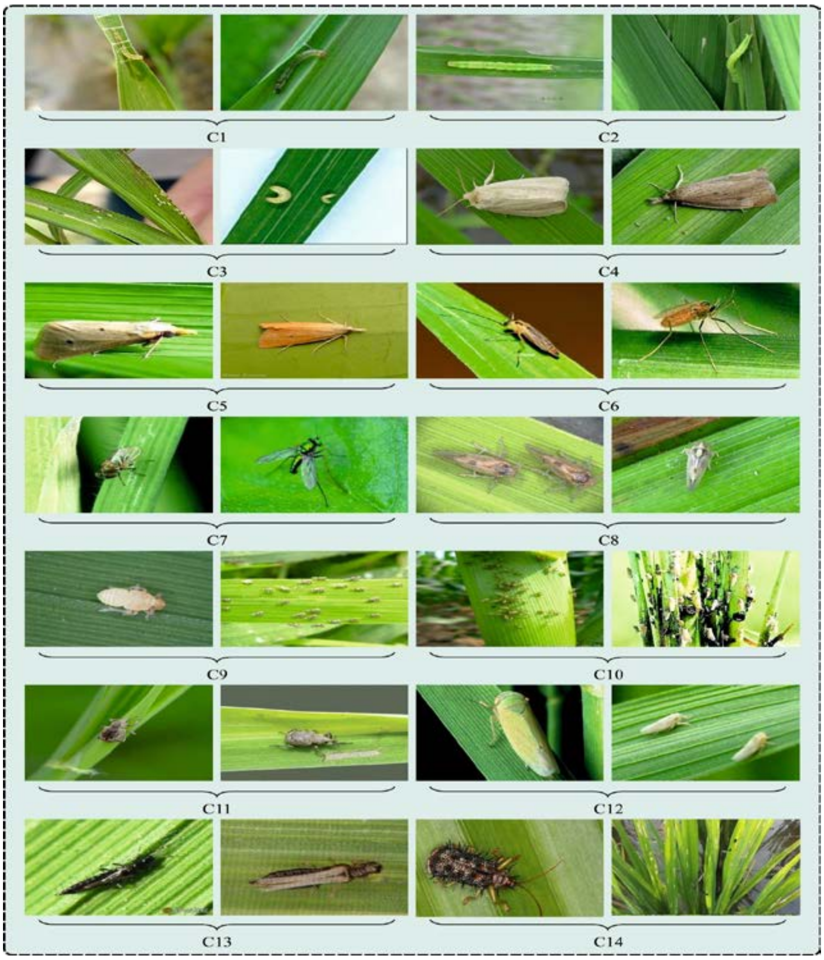

Fig. 3. Sample images.

Table 2. Description of database

S. No	Class	Labels	No. of Samples
1	Rice Leaf Roller	C-1	400
2	Rice Leaf Caterpillar	C-2	150
3	Paddy Stem Maggot	C-3	100
4	Asiatic Rice Borer	C-4	400
5	Yellow Rice Borer	C-5	200
6	Rice Gall Midge	C-6	200
7	Rice Stemfly	C-7	120
8	Brown Plant Hopper	C-8	400
9	White Backed Plant Hopper	C-9	400
10	Small Brown Plant Hopper	C-10	230
11	Rice Water Weevil	C-11	400
12	Rice Leafhopper	C-12	140
13	Grain Spreader Thrips	C-13	100
14	Spiny Beetle	C-14	140

Fig. 4 represents the confusion matrices produced by the FFADL-ARPDC algorithm on varying datasets. The simulation values referred that the FFADL-ARPDC approach has identified and classified all 14 classes accurately.

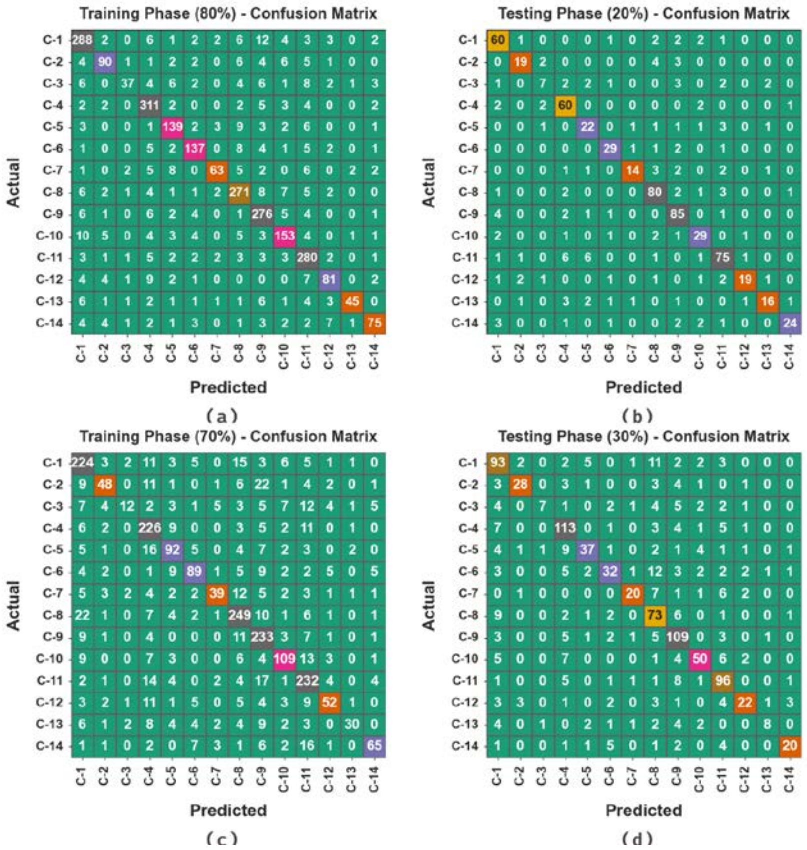

Fig. 4. Confusion matrices of (a-b) 80:20 of TR set/TS set and (c-d) 70:30 of TR set/TS set

Table 3. Rice pest detection and classification outcome of FFADL-ARPDC approach on 80:20 of TR set/TS set

Class Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TR set (80%)				
C-1	96.34	83.72	87.01	85.33
C-2	98.00	80.36	73.77	76.92
C-3	98.11	82.22	46.25	59.20
C-4	97.19	85.21	93.39	89.11
C-5	97.67	80.81	82.25	81.52
C-6	98.04	85.09	82.53	83.79
C-7	98.41	86.30	65.62	74.56
C-8	96.67	84.16	87.42	85.76
C-9	96.71	82.39	90.20	86.12
C-10	97.23	81.38	79.27	80.31
C-11	96.63	81.63	90.91	86.02
C-12	98.08	78.64	72.97	75.70
C-13	98.78	90.00	61.64	73.17
C-14	98.26	82.42	70.75	76.14
Average	**97.58**	**83.17**	**77.43**	**79.55**
TS set (20%)				
C-1	96.45	80.00	86.96	83.33
C-2	97.93	79.17	67.86	73.08
C-3	97.19	53.85	35.00	42.42
C-4	96.30	76.92	89.55	82.76
C-5	96.89	64.71	70.97	67.69
C-6	98.22	80.56	85.29	82.86
C-7	98.08	82.35	58.33	68.29
C-8	96.30	84.21	88.89	86.49
C-9	95.86	81.73	90.43	85.86
C-10	97.34	74.36	78.38	76.32
C-11	95.12	82.42	81.52	81.97
C-12	98.08	86.36	65.52	74.51
C-13	97.63	76.19	59.26	66.67
C-14	98.08	88.89	70.59	78.69
Average	**97.10**	**77.98**	**73.47**	**75.07**

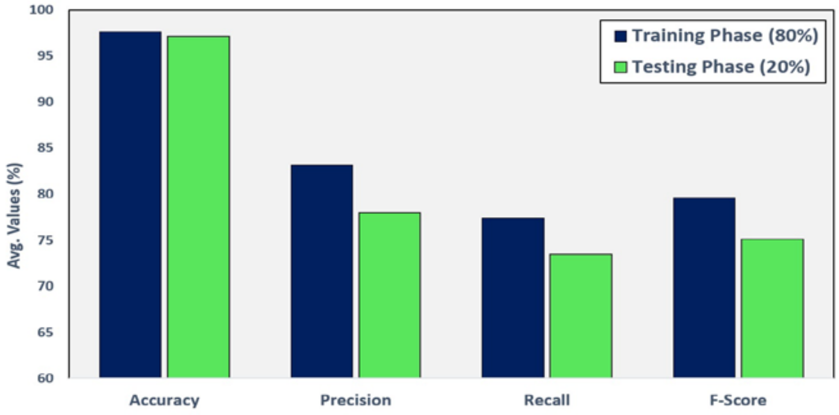

Fig. 5. Average of FFADL-ARPDC system on 80:20 of TR set/TS set

In **Table 3** and **Fig. 5**, the rice pest detection and classification results of the FFADL-ARPDC technique on 80:20 of the TR set/TS set are highlighted. The results imply that the FFADL-ARPDC technique accomplishes effectual results under all classes. On the 80% TR set, the FFADL-ARPDC technique offers an average $accu_y$ of 97.58%, $prec_n$ of 83.17%, $reca_l$ of 77.43%, and F_{score} of 79.55%. At the same time, on the 20% TR set, the FFADL-ARPDC technique offers an average $accu_y$ of 97.10%, $prec_n$ of 77.98%, $reca_l$ of 73.47%, and F_{score} of 75.07%. **Fig. 6** demonstrates the training accuracy TR_accu_y and VL_accu_y of the FFADL-ARPDC technique on 80:20 of the TR set/TS set. The TL_accu_y is defined by the evaluation of the FFADL-ARPDC technique on the TR dataset whereas the VL_accu_y is computed by evaluating the performance on a separate testing dataset. The results exhibit that TR_accu_y and VL_accu_y increase with an upsurge in epochs. Consequently, the performance of the FFADL-ARPDC technique gets improved on the TR and TS dataset with an increase in the number of epochs.

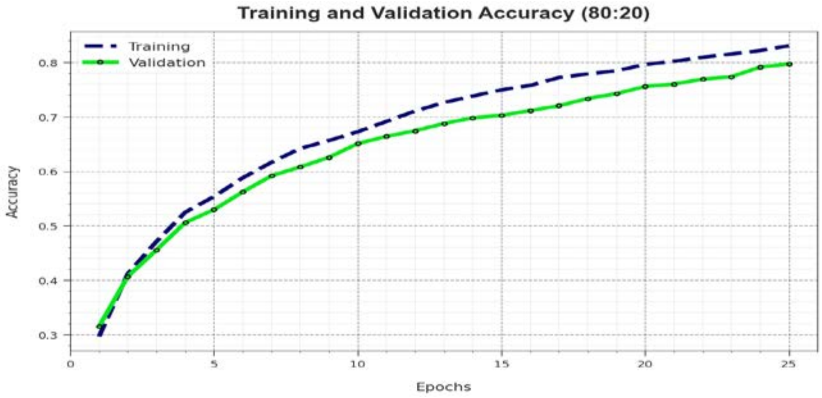

Fig. 6. $Accu_y$ curve of FFADL-ARPDC system on 80:20 of TR set/TS set

In **Fig. 7**, the TR_loss and VR_loss outcomes of the FFADL-ARPDC technique on 80:20 of the TR set/TS set are shown. The TR_loss defines the error among the predictive outcome and original values on the TR data. The VR_loss represents the measure of the performance of the FFADL-ARPDC technique on individual validation data. The results indicate that the TR_loss and VR_loss tend to decrease with rising epochs. It depicted the enhanced performance of the FFADL-ARPDC system and its capability to create an accurate classification.

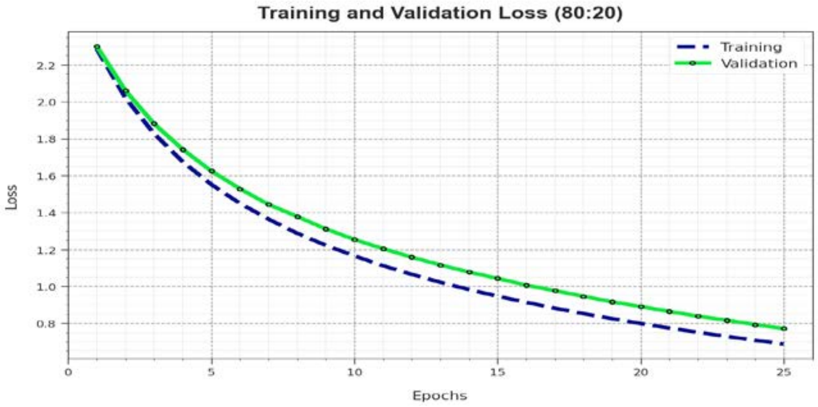

Fig. 7. Loss curve of FFADL-ARPDC system on 80:20 of TR set/TS set

The reduced value of TR_loss and VR_loss demonstrates the enhanced performance of the FFADL-ARPDC methodology in capturing patterns and relationships. A detailed PR outcome of the FFADL-ARPDC algorithm is depicted on 80:20 of the TR set/TS set in **Fig. 8**. The simulation values inferred that the FFADL-ARPDC approach outcomes in greater PR values. Then, it can be noted that the FFADL-ARPDC system attains higher PR values on 14 classes.

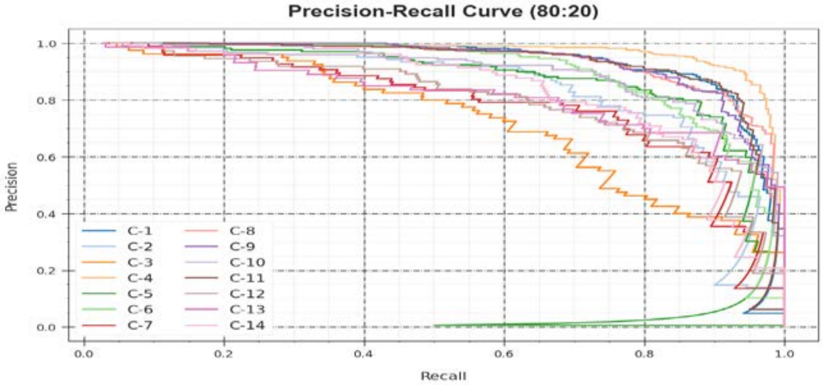

Fig. 8. PR curve of FFADL-ARPDC system on 80:20 of TR set/TS set

In **Fig. 9**, a ROC analysis of the FFADL-ARPDC system is exposed on 80:20 of the TR set/TS set. The figure described that the FFADL-ARPDC methodology led to superior values of ROC. Afterwards, it can be clear that the FFADL-ARPDC approach achieves greater values of ROC on 14 classes.

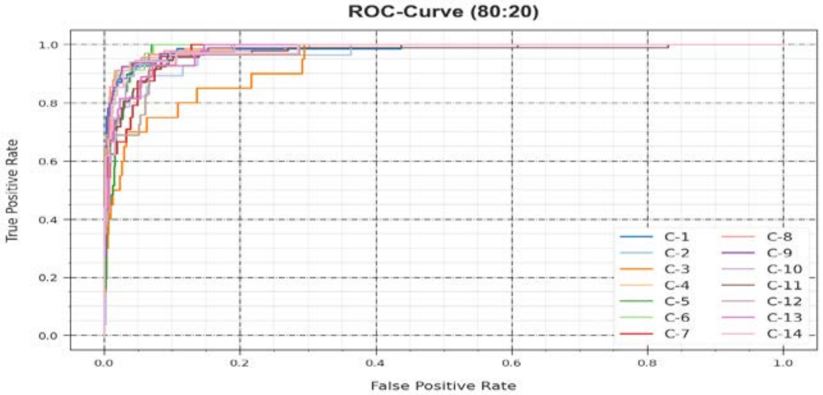

Fig. 9. ROC of FFADL-ARPDC system on 80:20 of TR set/TS set

In **Table 4** and **Fig. 10**, rice pest detection and classification outcomes of the FFADL-ARPDC methodology on 70:30 of the TR set/TS set are highlighted. The results imply that the FFADL-ARPDC system accomplishes effectual outcomes under all classes. On a 70% TR set, the FFADL-ARPDC technique offers an average $accu_y$ of 95.87%, $prec_n$ of 72.33%, $reca_l$ of 62.97%, and F_{score} of 65.87%. Simultaneously, on the 30% TR set, the FFADL-ARPDC method offers an average $accu_y$ of 95.58%, $prec_n$ of 68.21%, $reca_l$ of 60.86%, and F_{score} of 62.92%.

Table 4. Rice pest detection and classification outcome of FFADL-ARPDC method on 70:30 of TR set/TS set

Class Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TR set (70%)				
C-1	94.00	72.43	78.70	75.43
C-2	97.00	78.79	47.71	59.43
C-3	97.63	72.09	41.33	52.54
C-4	93.96	70.83	84.10	76.90
C-5	96.11	69.12	65.28	67.14
C-6	95.94	68.87	53.68	60.33
C-7	97.97	80.36	54.88	65.22
C-8	93.11	67.61	82.93	74.49
C-9	94.38	72.27	86.27	78.65
C-10	95.48	68.09	60.76	64.21
C-11	94.29	71.97	82.78	77.00
C-12	97.04	69.49	44.09	53.95
C-13	97.97	81.25	38.24	52.00
C-14	97.30	69.41	60.82	64.84
Average	**95.87**	**72.33**	**62.97**	**65.87**
TS set (30%)				
C-1	92.70	67.63	76.42	71.76
C-2	96.75	64.29	43.90	52.17
C-3	98.03	66.67	40.00	50.00
C-4	93.49	67.11	85.47	75.19
C-5	95.07	55.36	55.36	55.36
C-6	96.15	73.58	60.94	66.67
C-7	97.24	67.86	50.00	57.58
C-8	93.59	68.46	78.76	73.25
C-9	94.08	70.00	84.48	76.56
C-10	96.35	76.92	69.44	72.99
C-11	93.98	75.00	77.95	76.45
C-12	96.06	66.67	29.79	41.18
C-13	97.44	68.75	34.38	45.83
C-14	97.14	66.67	65.12	65.88
Average	**95.58**	**68.21**	**60.86**	**62.92**

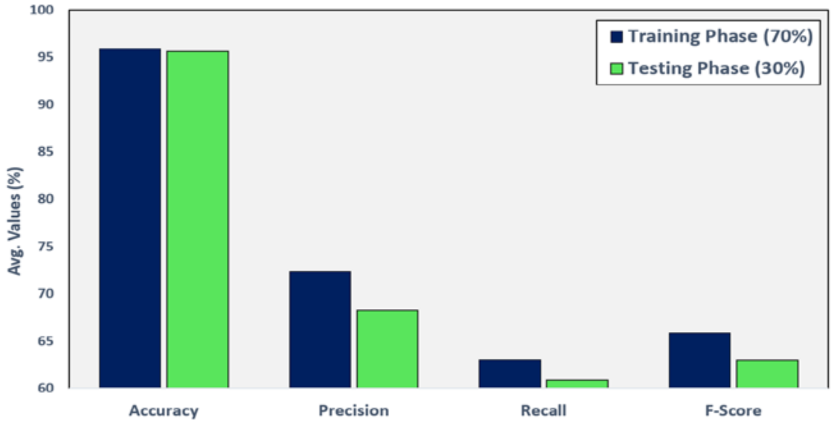

Fig. 10. Average of FFADL-ARPDC method on 70:30 of TR set/TS set

Fig. 11 shows the training accuracy TR_accu_y and VL_accu_y of the FFADL-ARPDC technique at 70:30 of the TR set/TS set. The TL_accu_y is determined by the evaluation of the FFADL-ARPDC technique on the TR dataset whereas the VL_accu_y is computed by evaluating the performance on a separate testing dataset. The outcomes exhibit that TR_accu_y and VL_accu_y increase with an upsurge in epochs. Therefore, the performance of the FFADL-ARPDC technique gets improved on the TR and TS dataset with a rise in the number of epochs.

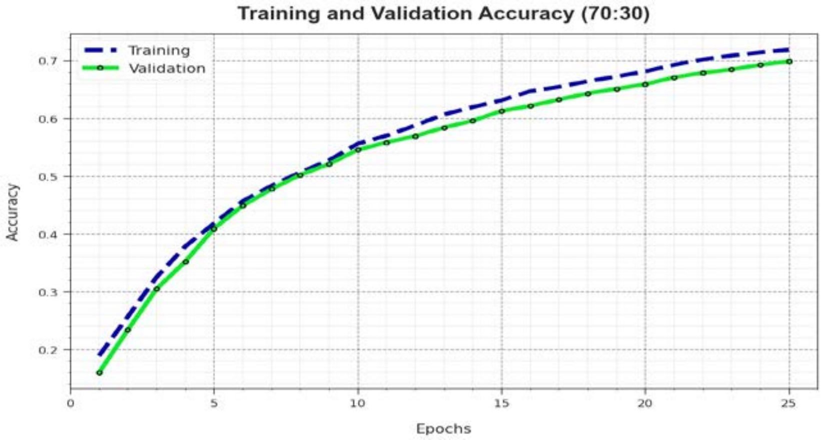

Fig. 11. $Accu_y$ curve of FFADL-ARPDC methodology on 70:30 of TR set/TS set

In Fig. 12, the TR_loss and VR_loss results of the FFADL-ARPDC method on 70:30 of the TR set/TS set are shown. The TR_loss defines the error among the predictive performance and original values on the TR data. The VR_loss represent the measure of the performance of the FFADL-ARPDC system on individual validation data. The results indicate that the TR_loss and VR_loss tend to reduce with rising epochs. It depicted the enhanced performance of the FFADL-ARPDC method and its capability to generate accurate classification. The reduced value of TR_loss and VR_loss demonstrates the improved performance of the FFADL-ARPDC method in capturing patterns and relationships.

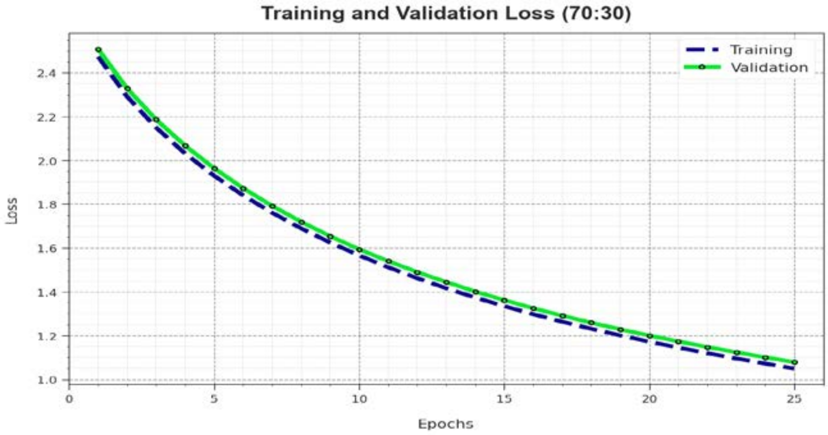

Fig. 12. Loss curve of FFADL-ARPDC system on 70:30 of TR set/TS set

A comprehensive PR curve of the FFADL-ARPDC method is depicted at 70:30 of the TR set/TS set in Fig. 13. The simulation value stated that the FFADL-ARPDC methodology outcomes in higher values of PR. Moreover, the FFADL-ARPDC system can reach maximum PR values on 14 classes.

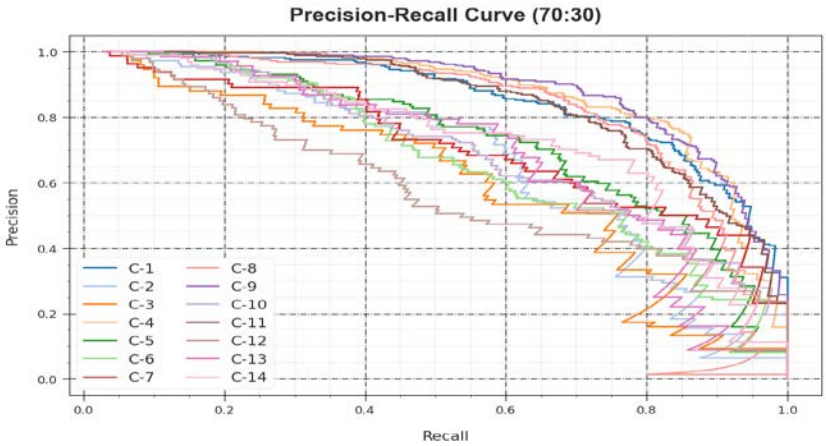

Fig. 13. PR curve of FFADL-ARPDC system on 70:30 of TR set/TS set

In **Fig. 14**, a ROC analysis of the FFADL-ARPDC system is shown at 70:30 of the TR set/TS set. The outcome value defined that the FFADL-ARPDC system led to enhanced ROC outcomes. The FFADL-ARPDC approach shows improved ROC values on 14 classes.

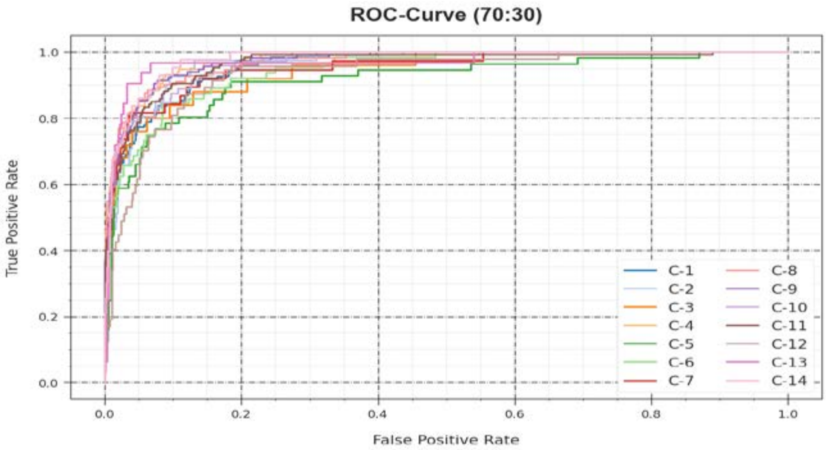

Fig. 14. ROC of FFADL-ARPDC system on 70:30 of TR set/TS set

Table 5. Comparative outcome of FFADL-ARPDC approach with recent approaches

Model	$Prec_n$	$Reca_l$	F_{Score}	$Accu_y$
ResNet50	75.76	72.02	60.67	83.17
VGG16	67.86	65.83	69.45	84.39
MobileNet	66.39	74.73	63.65	82.89
ShuffleNet V2	68.57	71.76	66.81	93.67
CTPDR-ECAAM	67.07	67.77	61.64	87.06
Leaky Relu	72.80	72.92	71.80	90.23
CA+INCV	63.13	75.19	65.74	94.35
FFADL-ARPDC	83.17	77.43	79.55	97.58

The comparative pest detection results of the FFADL-ARPDC technique are demonstrated in **Table 5** and **Fig. 15** [25, 26].

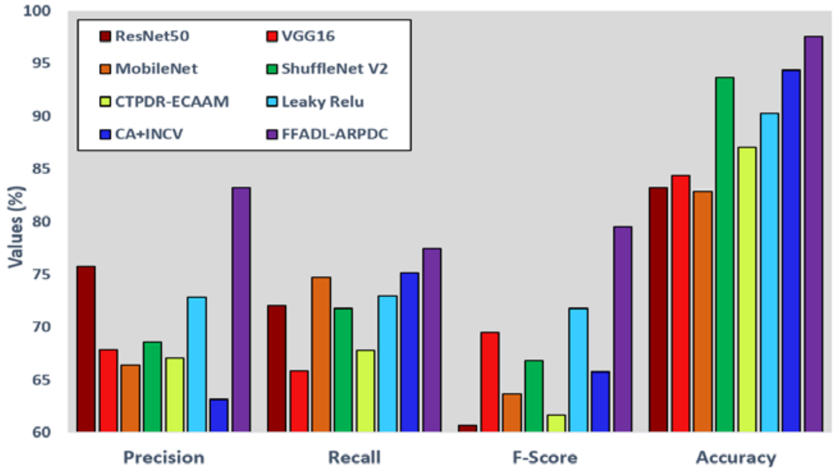

Fig. 15. Comparative outcome of FFADL-ARPDC approach with recent approaches

The outcomes exhibited that the VGG16, MobileNet, ShuffleNet V2, CTPDR-ECAAM, and CA+INCV techniques offered poor performance. Along with that, the ResNet50 and Leaky Relu techniques have provided slightly boosted results. Nevertheless, the FFADL-ARPDC technique illustrated better results with $prec_n$ of 83.17%, $reca_l$ of 77.43%, F_{score} of 79.55%, and $accu_y$ of 97.58%.

Table 6. CT outcome of FFADL-ARPDC approach with recent approaches

Model	**Computational Time (sec)**
ResNet50	2.02
VGG16	2.48
MobileNet	1.08
ShuffleNet V2	2.43
CTPDR-ECAAM	2.53
Leaky Relu	1.07
CA+INCV	1.45
FFADL-ARPDC	0.89

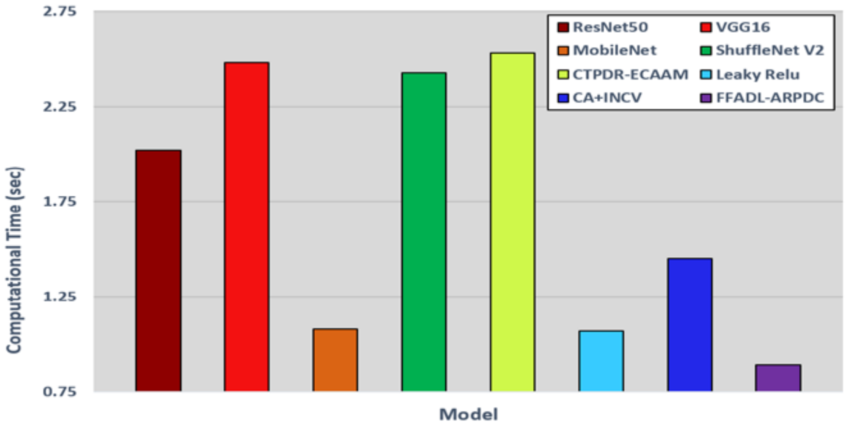

Fig. 16. CT outcome of FFADL-ARPDC approach with recent approaches

In **Table 6** and **Fig. 16**, comparative computation time (CT) results of the FFADL-ARPDC technique are provided. The results indicate that the FFADL-ARPDC technique offers effectual performance with a minimal CT of 0.89s. On the other hand, the existing ResNet50, VGG16, MobileNet, ShuffleNet V2, CTPDR-ECAAM, Leaky Relu, and CA+INCV techniques obtain maximum CT values of 2.02s, 2.48s, 1.08s, 2.43s, 2.53s, 1.07s, and 1.45s respectively.

The FFADL-ARPDC approach exemplifies extensive progress in the domain of agricultural image analysis, especially for the classification of rice pests from plant images. By integrating the NASNetLarge DL structure for feature extractor, the FFA for modifying model parameters, and the ERNN for particular pest identification, this method exposes a unique synergy of cutting-edge methods. The employment of FFA ensures optimum solution by adjusting the model to the intricacies of agricultural analytics, improving its robustness and accuracy. The FFADL-ARPDC approach with its combined components not only addresses the challenges of rice pest classification among them paves the way for developments in tailored performances for agricultural image-based tasks, and presents appreciated insights into the optimizer of DL approaches for real-world agricultural applications. Thus, the FFADL-ARPDC algorithm ensured optimum performance with other approaches.

5. Conclusion

In this study, we focus on the design and development of automated rice pest detection and classification using the FFADL-ARPDC technique. The major intention of the FFADL-ARPDC model is to exploit the DL model with a hyperparameter tuning strategy for the identification and categorization of rice pests into different classes. To accomplish this, the FFADL-ARPDC method encompasses pre-processing, NASNetLarge feature extraction, FFA-based hyperparameter tuning, and ERNN-based classification. To optimize the model performance of the NASNetLarge, the FFA is utilized for the hyperparameter tuning process, which helps in improving the classification performance. The FFADL-ARPDC approach is thoroughly evaluated using IP102-dataset from the Kaggle repository. The FFADL-ARPDC model is compared against traditional detection methods, demonstrating its superior performance in accurately detecting and classifying pests. In future, a feature fusion approach can be derived to improve the performance of the pest detection process.

Acknowledgement

We, Hussain. A and Balaji Srikaanth. P state that the content of this article entitled “Leveraging Deep Learning and Farmland Fertility Algorithm for Automated Rice Pest Detection and Classification Model” does not contain any conflict of interest.

References

[1] S. Jain et al., “Automatic Rice Disease Detection and Assistance Framework Using Deep Learning and a Chatbot,” *Electronics*, vol. 11, no. 14, p. 2110, Jul. 2022. Article (CrossRef Link)

[2] Y. Hu, X. Deng, Y. Lan, X. Chen, Y. Long, and C. Liu, “Detection of Rice Pests Based on Self-Attention Mechanism and Multi-Scale Feature Fusion,” *Insects*, vol. 14, no. 3, p. 280, Mar. 2023. Article (CrossRef Link)

[3] A. A. J. V. Priyangka and I. M. S. Kumara, “Classification Of Rice Plant Diseases Using the Convolutional Neural Network Method,” *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 12, no. 2, pp. 123-129, Aug. 2021. Article (CrossRef Link)

[4] P. Tejaswini, P. Singh, M. Ramchandani, Y. K. Rathore, and R. R. Janghel, “Rice Leaf Disease Classification Using Cnn,” *IOP Conference Series: Earth and Environmental Science*, vol. 1032, no. 1, p. 012017, Jun. 2022. Article (CrossRef Link)

[5] S. I. Hassan, M. M. Alam, U. Illahi, and M. Mohd Suud, “A new deep learning-based technique for rice pest detection using remote sensing,” *PeerJ Computer Science*, vol. 9, p. e1167, Mar. 2023. Article (CrossRef Link)

[6] V. K. Shrivastava, M. K. Pradhan, and M. P. Thakur, “Application of Pre-Trained Deep Convolutional Neural Networks for Rice Plant Disease Classification,” in *Proc. of 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Mar. 2021. Article (CrossRef Link)

[7] K. Abdalgader and J. H. Yousif, "Agricultural Irrigation Control using Sensor-enabled Architecture," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 10, pp. 3275-3298, 2022. Article (CrossRef Link)

[8] R. Sharma, V. Kukreja, and V. Kadyan, “Hispa Rice Disease Classification using Convolutional Neural Network,” in *Proc. of 2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021. Article (CrossRef Link)

[9] H. Kang et al., “A Novel Deep Learning Model for Accurate Pest Detection and Edge Computing Deployment,” *Insects*, vol. 14, no. 7, p. 660, Jul. 2023. Article (CrossRef Link)

[10] H. T. Ung, H. Q. Ung, T. T. Nguyen, and B. T. Nguyen, “An Efficient Insect Pest Classification Using Multiple Convolutional Neural Network Based Models,” *New Trends in Intelligent Software Methodologies, Tools and Techniques*, vol. 355, pp. 584-595, Sep. 2022. Article (CrossRef Link)

[11] H. Gong et al., “Based on FCN and DenseNet Framework for the Research of Rice Pest Identification Methods,” *Agronomy*, vol. 13, no. 2, p. 410, Jan. 2023. Article (CrossRef Link)

[12] C. R. Rahman et al., “Identification and recognition of rice diseases and pests using convolutional neural networks,” *Biosystems Engineering*, vol. 194, pp. 112–120, Jun. 2020. Article (CrossRef Link)

[13] K. Thenmozhi and U. Srinivasulu Reddy, “Crop pest classification based on deep convolutional neural network and transfer learning,” *Computers and Electronics in Agriculture*, vol. 164, p. 104906, Sep. 2019. Article (CrossRef Link)

[14] P. B. and M. Akila, “IoT-based pest detection and classification using deep features with enhanced deep learning strategies,” *Engineering Applications of Artificial Intelligence*, vol. 121, p. 105985, May 2023. Article (CrossRef Link)

[15] V. Malathi and M. P. Gopinath, “Classification of pest detection in paddy crop based on transfer learning approach,” *Acta Agriculturae Scandinavica, Section B — Soil &amp; Plant Science*, vol. 71, no. 7, pp. 552–559, Feb. 2021. Article (CrossRef Link)

[16] A. B. Kathole, K. N. Vhatkar, and S. D. Patil, “IoT-Enabled Pest Identification and Classification with New Meta-Heuristic-Based Deep Learning Framework,” *Cybernetics and Systems*, vol. 55. no. 2. pp. 380-408, 2024. Article (CrossRef Link)

[17] E. Ayan, H. Erbay, and F. Varçın, “Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks,” *Computers and Electronics in Agriculture*, vol. 179, p. 105809, Dec. 2020. Article (CrossRef Link)

[18] H. Peng et al., “Crop pest image classification based on improved densely connected convolutional network,” *Frontiers in Plant Science*, vol. 14, Apr. 2023. Article (CrossRef Link)

[19] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” *Information Fusion*, vol. 55, pp. 220–244, Mar. 2020. Article (CrossRef Link)

[20] M. Hayati et al., “Impact of CLAHE-based image enhancement for diabetic retinopathy classification through deep learning,” *Procedia Computer Science*, vol. 216, pp. 57–66, 2023, Article (CrossRef Link)

[21] M. Mehmood et al., "Improved colorization and classification of intracranial tumor expanse in MRI images via hybrid scheme of Pix2Pix-cGANs and NASNet-large," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4358–4374, Jul. 2022. Article (CrossRef Link)

[22] F. S. Gharehchopogh and H. Shayanfar, "Automatic Data Clustering Using Farmland Fertility Metaheuristic Algorithm," in *Advances in Swarm Intelligence*, Oct. 2022, pp. 199–215. Article (CrossRef Link)

[23] N. P. Kumar, S. Vijayabaskar, L. Murali, and K. Ramaswamy, "Design of optimal Elman Recurrent Neural Network based prediction approach for biofuel production," *Scientific Reports*, vol. 13, no. 1, May 2023. Article (CrossRef Link)

[24] Kaggle Dataset. Article (CrossRef Link)

[25] Z. Li, X. Jiang, X. Jia, X. Duan, Y. Wang, and J. Mu, "Classification Method of Significant Rice Pests Based on Deep Learning," *Agronomy*, vol. 12, no. 9, p. 2096, Sep. 2022. Article (CrossRef Link)

[26] H. Ni et al., "Classification of Typical Pests and Diseases of Rice Based on the ECA Attention Mechanism," *Agriculture*, vol. 13, no. 5, p. 1066, May 2023. Article (CrossRef Link)

Hussain. A is a Research Scholar in the Department of Networking and Communications at the SRM Institute of Science and Technology, Kattankulathur, India. His areas of expertise include Machine Learning, Cloud Computing, and Wireless sensor networks. He has seven years of academic experience. He has published articles in reputable publications and participated in numerous seminars, workshops, national and international conferences.

Balaji Srikaanth. P is working as Assistant Professor in the Department of Networking and Communications, SRM Institute Science and Technology, Kattankulathur, India. He is specialized in the area of Mobile computing, Network security, Wireless Sensor Networks and Ad hoc Networks. He has more than 15 years of academic experience. He has participated in various seminars, workshop, and national conferences and has published papers in reputed journals.