ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2025.1834.892

A DEEP ENSEMBLE LEARNING AND INTELLIGENT HYBRID FEATURE EXTRACTION MODEL FOR THE PEST DETECTION AND CLASSIFICATION IN CLOUD COMPUTING

P. Bharathi[1*], Dr. K. Dhanalakshmi [2]

[1]Research Scholar, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, Tamil Nadu, India. e-mail: bharathipadma19@gmail.com, orcid: https://orcid.org/0009-0009-9517-6773*
[2]Associate Professor, Head, Department of Information Technology, Kongunadu Arts and Science College, Coimbatore, Tamil Nadu, India.
e-mail: kdhanalakshmimca@gmail.com, orcid: https://orcid.org/0009-0002-3609-2288

Received: September 24, 2025; Revised: October 31, 2025; Accepted: December 04, 2025; Published: December 30, 2025

SUMMARY

This study presents an advanced approach for Pest Detection (PD) through automatic identification of invasive insects using Deep Ensemble Learning (DEL) and intelligent Feature Extraction (FE) techniques. The process begins with Z-score normalization (ZSN) to eliminate noise and improve classifier performance. Improved Feature Weighted Fuzzy Clustering (IFWFC) model is presented to achieve effective data grouping and Henon Chaotic Map Encryption (HCME) model to encrypt the image is presented. The extraction of the feature step involves the application of Hybrid Enhanced Wild Horse Optimization (EWHO) and Gray-Level Co-occurrence Matrix (GLCM) to extract the features accurately. The feature selection (FS) is performed on a modified version of Cuckoo Search Algorithm (MCSA) to achieve a precision in the model. The DEL model combines several deep learning models such as ResNet 50, Enhanced DenseNet201 and Granular Neural Network (GNN), which are optimized using the Pelican Optimization Algorithm (POA), to achieve a better classification. The IP102 data set was used to test the system with a better score of 99.90 as accuracy, 99.92 as precision, and 99.91 as a recall. The proposed model is much better than the current pest recognition techniques in classification with respect to accuracy, thus making it useful in real-time pest detection. This study presents the prospect of DEL-based model in precision agriculture and provides an efficient method of pest identification which can lead to better crop production and less dependence on pesticides.

Key words: *pest detection, z-score normalization, c-means clustering, HCME, resnet 50, enhanced densenet201, GNN.*

INTRODUCTION

Insect infestation is one of the crucial concerns that the farming sector experiences. These plants reduce overall food yield by 20–40% annually, according to the FAO [1]. Ecological degradation and the possibility of severe illnesses like cancer, genetic disorders, severe respiratory disorders, and foetal death are the consequences of these pesticides [2]. In agriculture, advanced technology solutions are eagerly waited in order to classify plant diseases early and avoid the needless application of planticides. Recently, smart agriculture has been used to better understand crop conditions, flooding, fertilisation and shrub plants in the grange field [3]. From the Internet

of Things (IoT), Artificial intelligence (AI) and wireless communication technologies are employed throughout the entire agricultural process [4]. Crop health monitoring describes the farm's situation with respect to the consideration of plant disease. In Figure 1, CHM has now been the more important application of smart agriculture.

Figure 1. An illustration of Four different types of insect plants

The basic component of the traditional crop (PMS) pest monitoring system (CPMS) is the observation and analysis of the administrator. Mistakes in assessment are common [5]. For PD, the application of machine learning (ML) approaches has become widespread in automated PMS due to the development of ML technology. A support vector machine (SVM) is used to identify pests by using a drone for capturing images. The histogram of orientated gradient (HOG) features of grayscale images (GSI) is extracted using SVM [6].

HOG effectively extracts attributes with a detection precision (P) of 88.9% on a self-built dataset, outperforming other feature descriptors. Many Pest Monitoring Systems (PMS) employ this technique, which first collects features from images and feeds them into an ML classifier. A sticky trap pest detection device uses k-means segmentation and color space transformation to extract features like mean and standard deviation (SD) [7]. In classification, K-nearest neighbor (KNN) outperforms the decision tree (DT) due to its ability to categorize pests based on previous knowledge [8]. New tools for developing pest detection systems are enabled by color space and image segmentation, with image processing techniques like threshold segmentation from the HALCON library [9]. Notably, Multilayer Perceptron (MLP) achieves higher precision and recall than KNN in pest detection, owing to its deeper structure [10]. Nonetheless, feature extraction procedures in ML may be vulnerable to the problem of generalization as they are sensitive to the presence of morphological and background alterations, which causes the lack of stability in the accuracy of detection.

The conventional object detection (OD) methods are not challenging in agriculture, where lightweight OD methods lead to lower detection accuracy [11]. The research paper is a proposal of a lightweight crop pest detection model based on ensemble deep learning (EDL), where the accuracy of the detection is high with reduced parameters and processing capabilities.The key contributions include: 1) implementing intelligent feature extraction (FE), 2) applying Deep Ensemble Learning (DEL) for improved pest detection, and 3) proposing a lightweight model that meets computational and storage performance criteria for precision agriculture. The remaining study is organised in the following way: Some of the most recent techniques for PD are examined in Section 2. The suggested methodology's steps are outlined in Section 3. Section 4 provides the results and discussion. The conclusion and further work are covered in Section 5.

LITERATURE REVIEW

Some of the most modern methods for detecting pests using ML and DL are reviewed in this section.

The Pest-You Only Look Once (YOLO) model, developed by Tang et al. [12], applies the YOLOv4 and an improved CNN to detect pests in real-time, with an addition of a squeeze-and-excitation attention mechanism to obtain relevant attributes and avoid redundancy to a large degree, being extremely accurate in pest detection. Implemented by Arun et al. [13], SSD, EfficientDet, Faster R-CNN, and CenterNet detect pests with the help of deep learning-based object detection (OD) models using the COCO dataset. Faster R-CNN_ResNet101_V1 outperformed others, achieving a mean average precision (mAP) of 74.77% on pest data. Wang et al. [14] developed detection methods like Faster RCNN (F-RCNN), SSD, YOLOv3, and Cascade R-CNN, which show promise for real-time pest monitoring. Faster-PestNet, suggested by Ali et al. [15], is based on MobileNet to keep the pest classification and demonstrated the accuracy of 82.43 % on the IP102 dataset. Jiao et al. [16] created AF-RCNN with results of 56.4% mAP and 85.1% mRecall on a 24-class pest dataset, which is better than YOLO and Faster R-CNN.

Wang et al. [17] improved pest detection by adding an attention mechanism to a residual network, surpassing state-of-the-art models with an average recall of 89.0% and mAP of 78.7%. YOLOLite-CSG, a lightweight crop pest detection technique by Cheng et al. [18], with a significant lower number of parameters, had 82.9% accuracy, which demonstrates its efficiency in computing. The CLAS strategy proposed by Zhang et al. [19] improves pest feature extraction in complicated backgrounds, using a coordination and local attention (CLA) strategy. MCPD-net, designed by Dong et al. [20], is an enhancement of MFPN and AFRPN which shows 67.3% mAP and 89.3% mean recall on MC pest 2021.

Boissard et al. [21] is a vision system of analyzing rose leaves that was proposed by Boissard et al. [21], in which it demonstrated good automatic processing of low levels of infestation. A method of early pest detection by means of SVM was introduced by Gondal et al. [22], which provides improved results as compared to other automated approaches. Although progressed, issues related to the image classification still exist as irrelevant features and poor feature selection (FS) algorithms. The studies reviewed summarize the critical advancement of deep learning-based pest detection techniques with the demonstration of the efficiency of many models like YOLO, Faster R-CNN, and MobileNet in enhancing accuracy and efficiency in pest detection in agriculture. Nevertheless, there are still problems of computational efficiency, feature extraction and limitations of datasets, indicating that improvements to the lightweight models and feature selection methods should be undertaken to be used in real-time and large-scale pest detection functions.

PROPOSED METHODOLOGY

An intelligent FE and DEL were presented in this research work to detect pests efficiently. At the outset, ZSN performs the preprocessing of removing the noises to enhance the performance of the classifier. Second, the Improved Feature Weighted Fuzzy Clustering based process introduced for the grouping process of the given dataset. Third, an encryption technique-based HCME method are used. Fourth, Hybrid Enhanced (WHO) Wild Horse Optimization (EWHO) with GLCM based feature extraction approach is introduced for extracting the features. Fifth, the feature selection is implemented by the method based on Modified Cuckoo Search Algorithm (MCSA). And finally, the DEL model is proposed for the efficient recognition of pests. Here the Resnet 50, Enhanced densenet201 and GNN for the improved performance of the classifier. The procedure of the suggested methodology is depicted in figure 2.

Insect Pests

As seen in Figure. 2, this research covers crop pests into 5 types: aphids, flea beetles, Cicadellidae, flax budworm, and red spider. They are the widespread agricultural pests especially in warm climates. Each of the insect pests may be defined briefly as follows:

Aphids: The aphids are one of the worst insect pests on crop development in the agricultural areas with high temperatures, including potatoes [23]. They can transmit disease of an infected plant to a healthy one, which leads to its damage and significant reduction in crop production.

Flea Beetles The flea beetles are the major insect pests that destroy harvests in North America and Europe [24]. Their feeding finally leads to the loss of the crop and severely damages new plants.

Cicadellidae A serious infestation of the Cicadellidae pest family, which includes the mango hopper, can result in a 50% reduction of agricultural yield [25].

Flax budworm In temperature-sensitive areas, infections of flax budworm have the potential to damage yields by 40–90%

Red Spider Mites A common pest in all farming areas are red spider mites. They attack a wide variety of host plants, including tomatoes, eggplants, beans, melons, and melons in GH (Greenhouses) or open fields, causing major harm.

Grain spreader thrips: A significant cowpea pest that significantly reduces grain yield is the grain spreader thrips. Plant to plant, it can spread quickly and readily

Black cut worm: The host black cutworm has a wide host range. It is able to feed on virtually all vegetables and also at times on crops and grasses, alfalfa, clover, cotton, rice, sorghum, strawberries, and sugar beets.

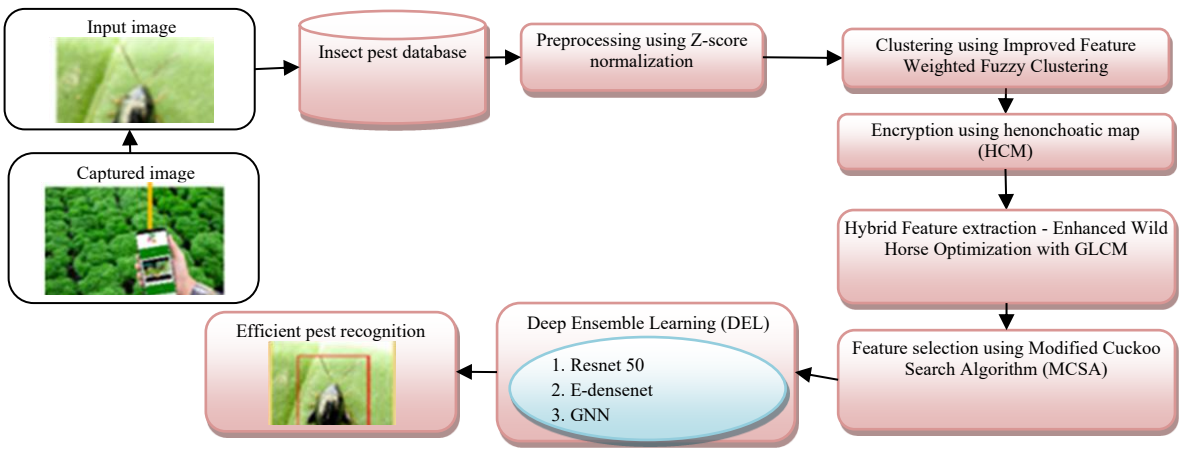

Figure 2. Schematic diagram of the recognizing crop pests using proposed model

Nipaecoccusvastalor: Nipaecoccusvastalor which is also known as the spiked mealy bug. Though the name is given to the wide range of host of the pest, the name most commonly used to refer to it is coconut mealybug. The palm trees and the tropical fruit harvests can be economically damaged by the coconut mealy insect.

User Interface (UI) Module

With the assistance of the developed UI of a smartphone, an agriculturalist in the expansive region or a GH could easily identify the crop pests on any platform, including iOS or Android-based techniques. The UI transmits an HTTP POST application with the picture of the unidentified insect pest that was taken by the user. All queries are handled by the flask web, which stores the input pest image in the cloud and sends it to the DL component for further procedure. The designed UI presents the pest categorization outcomes and associated insecticides at the conclusion of the image-recognition process, providing farmers appropriate conventional agricultural guidance.

Preprocessing Using Z-Score Normalization

Each dataset was first averaged to obtain the average intensity and then the average of averages in the search of a way to normalize the raw intensity data of each study [26]. This total mean was used to estimate the normalization variables which were then allocated to every test. The grand average was then equivalent to an average of all the normalized information. A plot of a Z-score on a normal distribution curve (NDC) is possible. The range of Z-scores is -3 SD, which would place them at the extreme left of the NDC, to +3 SD, which would place them at the extreme right. Users need to have knowledge of the mean (μ) and the population SD (σ) in order to calculate a z-score.

Here, xi (i = 1, 2,..,D) represent the *i*th element of every FV (Feature Vector) x ∈ R D. Initially, calculating these D components' mean and SD equation 1:

$$\mu_x = \frac{1}{D}\sum_{i=1}^{D} x_i, \sigma_x = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(x_i - \mu_x)^2} \quad (1)$$

Implementing ZSN is expressed as equation 2

$$x^{(zn)} = ZN(x) = \frac{x-\mu_x 1}{\sigma_x} \quad (2)$$

These computations indicate that the ZSN process first projections the initial FV on the 1 vector to the HP that is at right angles to √ 1 and covers the origin. Subsequently, these vectors are reduced to equal length D, meaning that the final normalized vectors are located on a hypersphere that is equal to √ D. After pre-processing the given data and then FS process is carried in which is described in the below section.

Improved Feature Weighted Fuzzy Clustering

FCM is an objective function (OF) optimization-based unsupervised FC technique. There are n pixels in the data X = {x1, x2, … , xn}. Here, V = {v1, v2, … , vc} is the cluster centre collection, the eigenvalue (EV) of the provided data is denoted as xj, and c is the number of categories. Assume that OF Jm meets the restriction.

$$\sum_{i=1}^{C} u_{ij} = 1(u_{ij} \in [0,1]) \quad (3)$$

$$J_m = \sum_{i=1}^{C}\sum_{j=1}^{n} u_{ij}^m d^2(x_i, v_j) \quad (4)$$

The membership of the j[th] sample and the *i*[th] category is denoted by u_{ij} as equation 3 and 4 . The weighted index, m, is typically equal to 2. The Euclidean distance (ED) between the cluster centre v_j and the sample point x_i is $d2(x_i, v_j) = \|x_i, v_j\|2$. The weighted distances from the data points (DP) to the corresponding v_j are added up to form the OF J [27]. The value decreases as the DP approaches v_j, indicating a stronger clustering effect. The measure of similarity of the intensity feature with the is only reported by the membership function (MF) in FCM; the spatial feature between the DP does not feature. However, as per the contribution, the contribution of the symmetric distribution on this study assists in overcoming the limitation through the designation of particular weight values of both spatial and attribute characteristics, and then integrated in relation to particular rules.

Symmetric Distribution factor

Assume n features are independent of each other. Following normalisation, the observed probability values of each feature are then multiplied by a symmetric distribution. The equation for the detailed computation is expressed as follows as equation 5:

$$p(z^1 \ldots z^n|x) = \prod_1^n \left(\frac{p(z^i|x)+\beta^i U(x)}{1+\beta^i}\right) \quad (5)$$

The uncertainty of *i* is represented by β^i. The discrete Steady state distribution is denoted by $U(x)$. The feature count is n. The probability values between the n features are denoted by p.

The Symmetric Distribution factor δ_{DF} as equation 6 calculation algorithm is then used to assign the correct weight values to various features.

$$\delta_{DF} = b.\left(-k.rand(\cdot) + \tan\left(\frac{\pi}{4} - \frac{\pi l}{4.Max_{iteration}}\right)\right) \quad (6)$$

Here, the current iteration number is denoted by l. The iterations maximum count is $Max_{iteration}$. The proportion coefficient is denoted by b. This b is employed to prevent the proportional imbalance. Together with the $rand$ function, the lessening δ_{DF} value is concerned, and k is the disturbance deviation factor. The modified leaders count every iteration is δ_{DF}·N, and the total amount of followers is $1 - \delta_{DF}$·N.

The segmentation process for the provided data is effectively improved using the model mentioned above. The FE procedure is explained in the section below.

Encryption using Henon Chaotic Map (HCM)

Henon (CE) chaotic encryption (HCE) is presented in this work to encrypt the image of the insect. The secret image is encrypted by taking advantage of this mechanism Equations (7) and (8). The selected pixel points of the cover image include this secret image embedded in them. Chaotic IE is an encryption (Enc) technique that follows a chaotic sequence produced by the chaotic scheme, as opposed to the traditional (IE) image encryption approach. The method that is being given encrypts a hidden image and then uses the extraction method to fully encrypt and decrypt the image. The Enc algorithm's strength is crucial. An Enc technique-based HCE approach can be moderate for employing the CE method since it has outstanding sensitivity (S) and can effectively withstand various types of attacks.

A two-dimensional, iterated, discrete-time dynamical system with a chaotic attractor is referred to the Henon map [28].

$$X_{n+1} = 1 + y_n - \alpha x_n^2 \quad (7)$$

$$Y_{n+1} = \beta x_n \quad (8)$$

(x, y) represents the system's present state, with x0, y0 serving as a starting point. The area S is circumscribed by four points (−1.33, 0.42), (1.32, 0.133), (1.245, −0.14), and (−1.06, −0.5), as demonstrated by Henon. (xi, yi) for i > 1 also exist in S if S is the area enclosed by four points and the initial point is located there

Using a HCM and an externally given 128-bit secret key, a permutation matrix for the image's pixel shuffling (confusion phase) and a cipher image for the encryption of the shuffled image (diffusion phase) was created by the suggested method. Being a private key algorithm, it is reasonable to assume that both the sender and the recipient have access to the same key.

Feature Extraction Using Hybrid Enhanced Wild Horse Optimization With GLCM

Enhanced WHO

The social behaviour of Wild Horse (WH) served as the model for the WHO algorithm. This WH also referred to as non-terrestrial horses. These horses live in two separate groups: a family group consisting of female horses (mares) and an individual group composed of male horses (stallions). Mating typically occurs between these two groups]. During early life, young horses, known as foals, focus primarily on grazing. As they grow, when they reach adulthood, male foals are recognized as stallions, while female foals depart from their original family group to join new ones. The "single group" is reinforced by stallions as an additional measure, aimed at preventing inbreeding by separating them from their birth groups. During dry seasons, competition for water highlights the dominance hierarchy, where leading stallions gain priority access to water sources, while less dominant members must wait. Although mares lead their respective family groups, they remain subordinate and are required to select a leader among the stallions and follow them. The following is an overview of the WHO algorithm's fundamental stages.

A. Population Initialization and Leadership Selection

A vector is produced by computing each population OF. $(\vec{x}) = \{\vec{x}_1, \vec{x}_2, \ldots \ldots \ldots \vec{x}_n\}$ is the result of a randomisation of the original population $(\vec{x})$, which consists of N members as equation 9.

$$(\vec{O}) = \{\vec{O}_1, \vec{O}_2, \ldots \ldots \ldots \vec{O}_n\} \quad (9)$$

G = NXPS, Here PS is the population-wide proportion of stallions, is used to divide the population into groups. An initial leader is selected at random from each group's stallions; however, as the algorithm progresses, the selection of leaders is determined by the greatest fitness value.

1) Grazing Behavior

Equation 10 depicts the grazing behaviour.

$$\bar{X}_{i,G}^{j} = 2Zcos(2\pi RZ) \times \left(Stallion^{j} - X_{i,G}^{j}\right) + Stallion^{j} \quad (10)$$

Here, member group's present location is denoted as $X_{i,G}^{j}$ equation 10. The location of leader group is denoted as $Stallion^{j}$, with the Z parameter defined as per Equation 12. Because of R, horses may eat from 360 degrees of angles relative to the group leader. A random value, R in the interval [−2, 2] is distributed uniformly. Movement across various radii is caused by the cosine function of R and π, and the value of π is roughly 3.14. Consequently, the member's ultimate location is denoted as $\bar{X}_{i,G}^{j}$, reflects the modified location of that member.

$$P = \vec{R}_1 < TDR: \qquad IDX = (P == 0); \quad (11)$$

$$Z = R_2 \ominus IDX + \vec{R}_3 \ominus (\sim IDX) \quad (12)$$

Here, R_2 specifies a random number ∈ [0, 1], P is a vector ∈ [0, 1], random vectors are $\vec{R}_1$ and $\vec{R}_3$, Z ∈ [0, 1] equation 11 and 12. When P == 0, the IDX indexes of the $\vec{R}_1$ returns meet the conditions. Equation 13 shows that the TDR decreases from 1 to 0 equation 13 .

$$TDR = 1 - iter \times \left(\frac{1}{maxiter}\right) \quad (13)$$

2) Horse Mating Behavior

Equations 14, 15, and 16 present decency and mating behaviour.

$$X_{G,K}^{P} = Crossover\left(X_{G,i}^{q}, X_{G,j}^{z}\right) \quad (14)$$

$$i \neq j \neq k \; p = q = end, \quad (15)$$

$$Crossover = Mean \quad (16)$$

Here, a horse that has puberty and whose parents leave groups i and j takes its place, and $X_{G,K}^{P}$ indicates the location of horse p as it exits group k. They have mated and produced $X_{G,i}^{q}$, although they are unrelated to each other. The foal q, who is in the i group, mated with the horse z, which is in the $X_{G,j}^{z}$ position, after reaching adolescence, leaving the j group.

B. Group leadership

They must be led by the group leader to the proper location near the water. The dominating group struggles for this water, and until the group leaves the others are not allowed to use it. As in (17), Equation 17 illustrates this behaviour, the current leader's location of group i as Stallion G_i, WH representing the water position, and Stallion G_i representing the leader's next location of the i group.

$$\overline{Stallion}_{G_i} = \begin{Bmatrix} 2Ccos(2\pi RZ) \times \left(WH - Stallion_{G_i}\right) + WH & if\ R_3 > 0.5 \\ 2Ccos(2\pi RZ) \times \left(WH - Stallion_{G_i}\right) - WH & if\ R_3 \leq 0.5 \end{Bmatrix} \quad (17)$$

1) Exchange and Leadership Selection

The first step is the random selection of the leaders. Another stage in the algorithm, however, selects the population that is the most suitable to be the leader. The positions of the chosen member and the leader in (18) are expressed as.

$$Stallion_{G_i} = \begin{Bmatrix} X_{G,i} & if\ \cos\ t\left(X_{G,i}\right) < \cos t\ (Stallion_{G_i}) \\ Stallion_{G_i} & if\ \cos\ t\left(X_{G,i}\right) > \cos t\ (Stallion_{G_i}) \end{Bmatrix} \quad (18)$$

The convergence rate of the method is influenced by blindness and ineffective transmission of information among groups that happens at the exploration stage of the WHO as it learns about random individuals. It is therefore an issue that the conventional WHO has slow convergence rate. MWHO can be proposed as a solution to the described issue to accelerate convergence.

Probability Function Approach

This design incorporates a probability function technique for the penalty function in addition to a new initial population strategy. Equations (19), (20), (21) and (22) represent a modest modification of the probability function technique for the penalty function.

$$\Phi(X) = F(X)(1 + penalty) \quad (19)$$

$$penalty = g_{ave}\left(\frac{(g_{max} + g(i))}{(g_{max} - g_{ave})}\right) g(i) \geq g_{ave} \quad (20)$$

$$penalty = g_{ave}\left(\frac{(g_{ave}+g(i))}{(g_{ave}-g_{min})}\right) g(i) < g_{ave} \quad (21)$$

$$penalty = 0 \qquad g(i) = 0 \quad i = 1, \ldots\ldots, m \quad (22)$$

Here, the vector for the design variables is denoted as X. The OF for minimum volume is denoted by F(X). The total number of constraints is n. The maximum, minimum, and average violation values of the generation are denoted by g_{max}, g_{min}, and g_{ave}. The modified OF is U(X). The total violation value of normalised displacement, g_{dj} (x), and stress, g_{sr} (x), is denoted by g(i). The following equations expressed the constraints of the ith individual and g_{dj} (x) and g_{sr} (x) as equation 23 and 24

$$g_{dj} = d_j/d_{uj} - 1 \qquad j = 1 \ldots\ldots m \quad (23)$$

$$g_{sr} = g_r/g_{ar} - 1 \qquad r = 1 \ldots\ldots m \quad (24)$$

Here, the displacement at the jth node's upper bound is denoted by d_{uj}. The permitted stress value in the rth element is g_{ar}. The displacement constraint's count is denoted as m. The count of stress constraints is represented by l. The violations of normalised constraints serve as the foundation for the formulation of the unconstrained optimisation problem. Since the likelihood of survival is not eliminated for designs with minor violations and lesser objective value, g_{ave} is added as a penalty parameter in the prior version of these formulations. The algorithm is compelled to remove designs with significant violations and higher objective values from the generation. The following is the MWOA probability function equation 25,26,27,28:

$$p_m = 0.5(f_{max} - f)/(f_{max} - f_{ave})\ f \geq f_{ave} \quad (25)$$

$$p_m = (f_{ave} - f)/(f_{ave} - f_{min})\ f < f_{ave} \quad (26)$$

$$p_c = (f_{max} - f')/(f_{max} - f_{ave})\ f' \geq f_{ave} \quad (27)$$

$$p_c = 1.0 f' < f_{ave} \quad (28)$$

In this case, this is the fitness of an individual (f). The mean population fitness is represented as. The upper and lower fitness values of the population are represented by and . The f0 represents the less fit of the solutions to be crossed. The amount of probability function violation is used to group design variables in an individual. The probability that the fitness value of an individual will affect the number of design variables perturbed to him is variable. Consequently, under the process of evolution, the probability function method can be modified automatically. Consequently, the algorithm does not need any pre-determined parameters.

GLCM

FE and FS are two types of dimensionality reduction (DR) techniques for categorisation in IP and computer vision. In order to develop new features that include the most valuable information for the target concept, FE completely transforms the initial attributes. Then, FS preserves all of the traits while eliminating those that are either essential or irrelevant to the target concept.

By considering the spatial relationship of pixels, the texture can be analysed by using a statistical technique named gray-level (GL) spatial dependence matrix, or GLCM. In order to extract texture information, GLCM examines the spatial relationships between pixels. In particular, it creates a matrix using the frequency with which pixel pairings with particular GL values appear at particular spatial relationships within an image. Texture analysis techniques such as "Calculate Statistical Measures of Texture" do not directly reveal the shapes or spatial connections of objects inside an image; instead, they concentrate on examining the spatial distribution of pixel intensities. Here, the co-occurrence matrix for small areas of the image is calculated. The Statistical values, such as entropy, probability, homogeneity, contrast, correlation, and homogeneity, are calculated using this matrix. The technique may effectively compute this matrix by converting distance and angle offsets to horizontal and vertical pixel offsets using a provided list.

By creating fourteen textural features based on the co-occurrence matrix, which is calculated from the probability matrix, Haralick is able to extract the texture statistics of images. The Angular Second Moment (ASM)(energy), Inertia Moment, Correlation, Entropy, and the Inverse Difference Moment (IDM) are the four primary features chosen for execution in this study.

a) ASM

Energy or Uniformity are other names for the ASM. It is the sum of squares of the entries in the GLCM. Image homogeneity is measured by the ASM. When pixels are very similar or the image is highly homogeneous, the ASM is high equation 29.

$$ASM = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} P_{ij}^2 \quad (29)$$

In a normalised GLCM, the (i,j)th entry is represented by the symbol p(i,j).

b) IDM

Local homogeneity is known as IDM. When the inverse GLCM is high and the local grey level is uniform, IDM is high.

$$IDM = \frac{\sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} P_{ij}}{1+(i-j)^2} \quad (30)$$

The inverse of the contrast weight is the IDM weight value. In a normalised GLCM, (i,j)th entry is represented by p(i,j) equation 30.

a) Entropy

In image compression, the measure of data regarding image is assessed by entropy. The image information and data loss or message in a transmitted signal are also measured by this entropy.

$$Entropy = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} -P_{ij} * logP_{ij} \quad (31)$$

In a normalised GLCM, (i,j)th entry is represented by p(i,j) equation 31.

b) Correlation

This correlation is employed to assess the linear relationship among the GL of nearby pixels. Here, the tracking and image registration techniques are used by an optical method named Digital Image Correlation (DIC). This will result in precise 2D and 3D assessments of image changes. For assessing deformation, displacement, strain, and optical flow, this DIC is mostly utilized. This DIC is used widely among several domains of science and engineering. Its well-known application is tracking motion of an optical mouse.

$$Correlation = \frac{\sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i,j)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (32)$$

In a normalised GLCM, the (i,j)th entry is represented by p(i,j). The means and standard deviations are denoted by $\mu_x, \mu_y, \sigma_x and\ \sigma_y$ equation 32.

Since the image can be used as an input to implement using MATlab, the four specified image attributes are formulated and obtained using MATlab for GLCM calculation.

FS using Modified Cuckoo Search algorithm (MCSA)

This section proposes a modified CS (Cuckoo Search) algorithm that incorporates opponent learning for improving the exploitation search ability and optimizes the execution of the fundamental CS technique. In order to initialize the population and generate new solutions for candidate in evolving generations, the uniform distribution method is integrated into CS. This technique can direct the population toward greater potential regions and disperse it as widely as possible throughout the searching space (SS).

Basic CS Algorithm

To simplify the basic CS, the following 3 optimal criteria are adopted: Cuckoos lay one egg at a time [30], and place it in a randomly chosen set; (2) the subsequent groups can be allowed to use the best nests made out of high-quality eggs; (3) the quantity of host nests available has been fixed, and the laid egg of the cuckoo is determined by the host bird by a process of probability, $pa \in$ [0,1]. In this condition, the host bird can have two alternatives: they can ignore the egg and create a more elaborate one.

The probability shows the effect of each generation's replacement of cuckoo eggs discovered through the host bird's new eggs. An egg symbolizes a solution. These premises guarantee that the most effective solutions are preserved through generations. This acts as a selection mechanism for the optimization method. Therefore, replacing the low-quality solutions in the nests with higher-quality ones is the aim of the CS algorithms. A novel solution X_i^{t+1} for cuckoo i is given by equation 33,34:

$$X_i^{t+1} = X_i^t + \alpha \otimes L'evy(\lambda) \quad (33)$$

$$\alpha = \alpha_0 \otimes (X_j^t - X_i^t) \quad (34)$$

Here the randomly chosen solution can be represented as X_j^t; α is the step size (α > 0) with dimension equal to the problem's dimension; entry-wise multiplications are symbolized by the product ⊗; and the Lévy flights (LF) random walks can be indicated as Lévy (λ). The value of α0 is set to 0.01 as suggested for improving the search effectiveness. The Lévydistribution (LD) determines the step length of one kind of random walk, known as a "LF." A probability density function (PDF) with a power-law tail is used to derive instantaneous jumps equation 36.

$$L'evy(\lambda) \approx S = t^{-\lambda}, (1 < \lambda \leq 3) \quad (35)$$

The step length S of LF is determined by a uniform distribution which satisfies the LD. Also the method used a balanced combination of global exploratory random walks and local random walks that were controlled using the switching parameter. One way to express the local random walk is given below

$$X_i^{t+1} = X_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (X_j^t - X_k^t) \quad (36)$$

Here, the Heaviside function can be symbolized as H, an arbitrary amount taken from a uniform distribution is ε, the two distinct solutions chosen at random-by-random permutation can be represented as $X_j^t$and X_k^t, and the step size is s.

Conversely, LF are employed to execute the global random walk equation 37:

$$X_i^{t+1} = X_i^t + \alpha \oplus L'evy(s, \lambda) \quad (37)$$

s scaling factor is denoted by $\alpha > 0$, and the step-lengths are represented by L´evy(s, λ), which are distributed in accordance with the probability distribution, that has an infinite mean and infinite variance equation 38 :

$$L'evy(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}} \quad (38)$$

Uniform distribution strategies

Here, X_i^tyet indicates a solution arbitrarily chosen from the existing superior solutions. Hence in the procedure of maintaining equilibrium amongst local (LS) and global search (GS) process, uniform distributiontechnique plays a significant part. The uniform distribution strategy establishes the contribution quantity of old velocity to its new velocity at the present time period. Consequently, uniform distribution strategies that observe the search condition and regulate the value of weight as per one or more feedback attributes.

When the equation (35) of velocity update is assessed, it is apparent that this Eqn is comprised of two segments. The first entity ($X_i(t)$) indicates the population. Through the direction of $Gbest$, the 2nd entity (($X_i(t) - Gbest$))affects the ith position. In order to generate better solutions, the guidance of the neighbor cuckoo is used. The position update equation of the actual CSA is altered for this reason as follows equation 39,40:

$$X_i(t+1) = U\big(V_i(t)\big) + (X_i(t) - Gbest)F_i\delta_1 + (X_i(t) - X_k(t))F_i\delta_2 \quad (39)$$

$$\delta_1 + \delta_2 = 1 \quad (40)$$

The uniform distribution factor U stabilizes both the GS and the ith solution's LS intensity; the superior solutions arbitrarily selected from the population ($i \neq k$) can be represented as Xk. The $\delta 1$ is the self-adaptive learning factor of the $Gbest$, which ranges from 0 to 1. Consequently, the learning factor of the ith solution is $\delta 2$, varying between 1 to 0. By using the information from the kth solution to guide the ith solution, the local minima may also be able to prevent in this procedure. The global best solution's influence increases with $\delta 1$, surpassing that of the (Xk) - kth neighbor solution is expressed as follows equation 41:

$$\delta_1 = 1 + (\delta_{init} - 1)\left(\frac{iter_{max} - iter}{iter_{max}}\right)^n, \quad (41)$$

Here, the maximum number of iterations can be denoted as itermax, the present number of iterations can be represented as iter, and the data of a nonlinear modulation can be indicated as n. the initial impact factor of $\delta 1$ is δinit. As the iteration is raised, $\delta 1$will likewise increase nonlinearly from δinit to 1, whereas $\delta 2$ will decrease proportionally from (1 − δinit) to 0. Cuckoos are allowed to fly over the SS having a small $\delta 1$ and a large $\delta 2$, even if they fly towards it. On the other hand, at the last stages of the search process, a big $\delta 1$ and a small $\delta 2$ permit the cuckoos to arrive to the global optimum solution. Therefore, at the final stage of optimization, and the suggested technique could efficiently regulate the GS and improve its convergence to achieve the $Gbest$.

To govern the magnitude of the velocity, the uniform distribution methodology is applied. This strategy is demonstrated as follows equation 42:

$$U = U_{max} * \exp\left(-m * \left(\frac{\delta}{\delta_{max}}\right)^m\right), \quad (42)$$

Here, the values of maximal distributed value are indicated by U_{max}, and m is a constant greater than 1. The advantage of the suggested MCSA is that it plays a role in the solution's dispersion to the search space. Moreover, further precise results can be attained.

Algorithm 1: MCSA

Input: pest dataset

Output: Optimized Features

(1) Set the iteration number t = 1, initialise the algorithm's parameter values, and generate random initial vector values.

(2) Calculate each person's fitness value (FV) to determine which individual now has the best objective value. Make sure the stopping condition is satisfied by checking. Update $t = t + 1$ and keep going until the stopping condition is satisfied if the stopping requirement is satisfied. Otherwise, output the most optimal solution.

(3) Get a set of new solutions $Xnew = \left[x_1^{(t+1)}, \ldots., x_i^{(t+1)}, \ldots x_K^{(t+1)}\right]$ by LF while keeping the best one from the last iteration.

(4) Assess the FV $F_i^{(t+1)}$ of the novel solution $x_i^{(t+1)}$, and contrast $F_i^{(t+1)}$ with $F_i^{(t)}$that denotes the solution of the tth iteration.

(5) A fraction (pa) of worse nests are abandoned and new ones are built.

(6) Keep the best solution.

(7) Search for a new solution using uniform distribution strategies.

(8) Keep the best nest with quality solution

(9) Rank the nests and find the current best one

(10) Pass the current best nest to the next generation

(11) Go to step (2)

Algorithm 1 applies a Cuckoo Search Optimization method to optimize pest detection features. It starts with the setting of parameters and random solutions. Fitnesses of solutions are checked and the solution with the best fitness is chosen. Levy Flights (LF) are used to come up with new solutions and retain the best solution of the previous iteration. Solutions are evaluated and those that are not good are dropped with the best maintained. New solutions are explored by uniform distribution strategies and the best solutions are ranked and carried over to the next stage. This is done until the stopping condition is achieved to get the best set of features to use when detecting pests.

Classification using DEL

In this study, the DEL framework is proposed for the efficient recognition of pests. Here the Resnet50, enhanced densenet201 and Granular Neural Network (GNN) for the improved performance of the classifier.

Resnet50

The (GBP) Gradient (BP) Back Propagation becomes unstable, when layer count increases in the network. This instability arises from the repeated multiplication of gradients during BP. It may be very little. The first issue resulting with increasing depth is gradient dissipation (GD)/explosion. One of the common problems is the gradient dissipation. There have been many techniques such as employing Xaiver initialization, replacing ReLU as the AF (Activation Function), and using BN (Batch Normalization) that have been found to fight GD. Gradient dissipation is also solved successfully, it can be concluded [31]. Other issues with network deepening are degradation. The performance of the network is getting worse with the depth; this is known as degradation.

According to earlier studies, the model's performance is significantly impacted by the network's depth. Deeper models may theoretically produce better results because the network can perform more complex FE patterns as the number of layers increases. The investigation found that the deep network was still degenerating. As network depth increases, the network's accuracy tends to become saturated or even decrease. The training set's accuracy is decreasing. It's clear that overfitting isn't the reason behind this. The training set should be highly accurate as a result of overfitting. ResNet solves this problem, and after it is solved, the network's depth is increased by several orders of magnitude.

The ultimate output is y = F(x)+x. The (IM) Identity Mapping and RM (Residual Mapping) are 2 types of mapping in ResNet that are shown in Figure 3. The term "curved curve" is used in IM. The term "RM" describes the area that is not the "curved curve."

If IM implies a "difference," or y − x, then RM relates to F(x), as the term suggests. In contrast, IM refers to itself, or x in the equation. ResNet-50 used four residual blocks (RB), fully connected (FC) layers, and convolution (Conv) operations on the input to finish classification tasks. Figure 4 shows the network framework of ResNet-50 with 50 Conv2D operations.

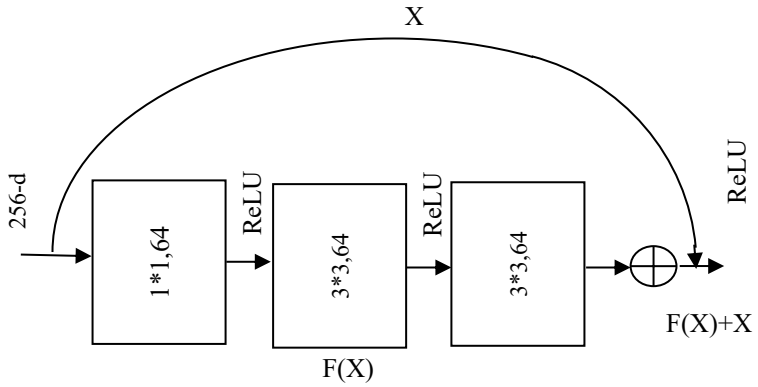

Figure 3. Residual block

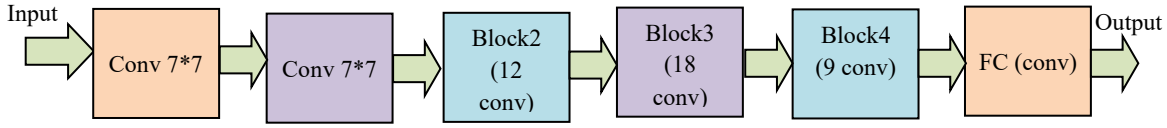

Figure 4. ResNet-50

To summarise the characteristics of the layers that came before it, the FC layer frequently performs at the end of the CNN stage. The feature engineering, local amplification, and local FE processes were performed by the prior Conv and pooling layers, and the final FC layer act as feature weighing. The nonlinear (NL) combinations of complex data that are extracted by the Conv layers are effectively learned via the FC layer. The basic steps in the learning procedure are given below. After being converted into a format that works with an MLP, the image is first compressed into column vectors and then reinserted into the Feed Forward NN (FNN). The compressed data is then used in all subsequent training procedures. The algorithm is capable of efficiently differentiating between the main aspects of the image and a few low-level features by employing classification techniques such as Softmax.

Densenet201

Each layer is feed-forwarded to every other layer using DenseNet, which maximises information flow between layers. By using feature reuse and dense connectivity patterns, it improves gradient flow, reduces the vanishing-gradient problem (VGP), and increases parameter efficiency in contrast to standard systems [32]. With corresponding feature-map (FM) sizes, every layer is linked to every other layer. The layer transmits its FM in subsequent layers after ears were fed by all the layers that proceeded it. The number of connections in an L-layer network can be L(L+1)/2 as the features are concatenated not summed.

To establish thick interactions among layers and give each layer direct access to rich set of FM of prior layers, DenseNets use a growth rate that is greater than 1. Figure 5 illustrates the pattern of connection of the 5-layer dense block (DB) at a growth rate of four (k=4). All the FM which has been generated by all the layers which come before it is inputted into each of the layers of the block.. The following layers (H2, H3, H4) generate their own set of FM by using all of the previous FM (x0, x1, x2) as inputs.

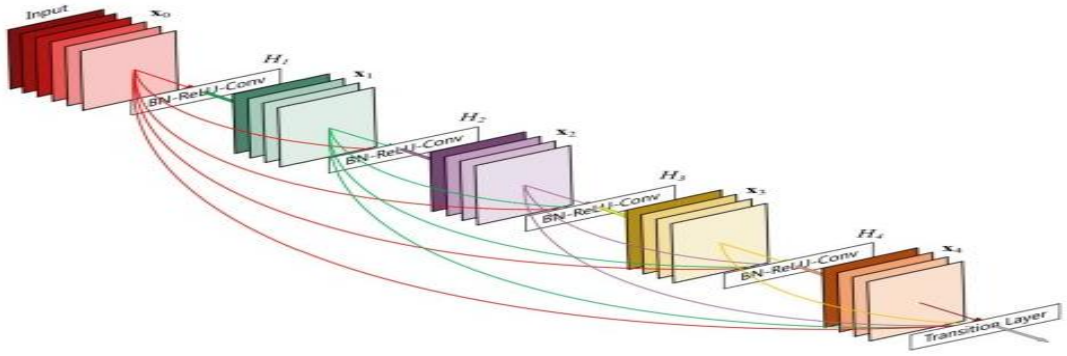

Figure 5. A DB with a growth rate of 4 (k=4) is shown in the 5-layer

BN, ReLU activation, and a single CL are all part of the dense_layer function in the original DenseNet201 architecture. To capture a variety of intricate visual features, the redesigned architecture's dense_layer has two Conv2D layers with varying filter sizes. To enhance performance and prevent overfitting, a dropout layer is used. Images with intricate pixel patterns can be handled efficiently by using the "he_normal" kernel initialiser. This initialiser overcomes the disadvantage of using a traditional normal distribution (ND) for weight initialisation when

combined with the ReLU AF. The "he_normal" initialiser chooses samples from a truncated ND with a centre of 0 and an SD, and the weight tensor's input unit count is denoted by n. An improved DenseNet201 architecture attempts to alleviate overfitting problems and enhance the framework's capacity for recording complex image attributes by implementing these changes.

Pelican Optimization Algorithm (POA)

POA is a novel stochastic nature-inspired optimisation method (NIOA). This POA is believed to be better in exploration and exploitation in the quest to attain the global optimum [33]. Swarm-inspired (SI) algorithms have been the most popular algorithms in recent times. Pelican hunting strategies and habits can act as a source of inspiration to POA. Finding their food source (prey), the pelicans dive, then drive the prey into shallow water by using their wings. The POA's primary steps are given below:

Initialization: All of the pelican members are regarded as candidate solutions in the population-based algorithm known as POA. Using the following equation, each member of the population is initialised at random to begin the optimisation process equation 43:

$$x_i = LB + rand * (UB - LB) i = 1,2, \ldots \ldots N \quad (43)$$

Here, the count of population members is denoted by N. The value of the candidate solution is denoted by xi. An arbitrary vector in the interval [0, 1] is called a rand. The solutions from those first candidates are then used to evaluate the OF for the given problem. After that, the OF vector is computed. To update the candidate solution, the hunting technique of pelicans attacking their food supply is simulated. The two stages of this process simulation are: Moving towards food source (Exploration stage), Winging on the water surface (exploitation stage)

Stage 1: Exploration stage:

The method used by pelicans to search the SS for food sources is simulated in this phase. The Pelicans begin to approach the prey location after identifying their prey. POA's ability to randomly generate the pray location, which boosts exploring power, is one of its key features. Its current state is expressed as equation 44,

$$x_i^{new\_1} = \begin{cases} x_i(t) + rand.\left(x_p - Ix_i\right), if F(x_p) \leq F(x_i) \\ x_i(t) - rand.\left(x_p - Ix_i\right), \quad else \end{cases} \quad (44)$$

Here, x_p is the randomly produced prey location, F(x_p) is the value of its OF. The vector I is produced at random and has a value of either 1 or 2. Next, the following update is made to the solution to reflect the updated location equation 45:

$$x_i(t+1) = \begin{cases} x_i^{new\_1}, if\ F(x_i^{new_1}) \leq F(x_i) \\ x_i(t), \quad else \end{cases} \quad (45)$$

Stage 2: Exploitation stage

The ability of LS is increased during this phase when pelicans begin to extend their wings on the water's surface that allow prey to move upward. In this stage, its current state is expressed as equation 46,

$$x_i^{new\_2} = x_i(t) + R.\left(1 - \frac{t}{T}\right).(2rand - 1)x_i(t) \quad (46)$$

Here, the current iteration is denoted by t. The maximum iteration count is T. and 0.2 is the value of the constant R. The solution is then updated in accordance with the new location as equation 47

$$x_i(t+1) = \begin{cases} x_i^{new\_2}, if\ F(x_i^{new\_^2}) \leq F(x_i) \\ x_i(t), \quad else \end{cases} \quad (47)$$

The OF value of the candidate solution $x_i^{new\_^2}$ is denoted as F($x_i^{new\_^2}$) .

GNN

Artificial NN (ANNs) that can analyse data that was initially quantitative or granular are the focus of the GNN concept [29]. The GNN approach focusses on online incremental learning from data streams. As illustrated in Figure 6, learning in GNN and GNN adhere to a single premise that consists of two stages. Initially, information granules (IG), which are intervals or, more broadly, fuzzy sets, are created using original numerical representation. The IG is used as the basis of the learning adaptation and refining of the NN as opposed to the original input. It is not essential to reveal to the NN all the data points since these are much more than IG. Examples are eliminated in the absence of new information.

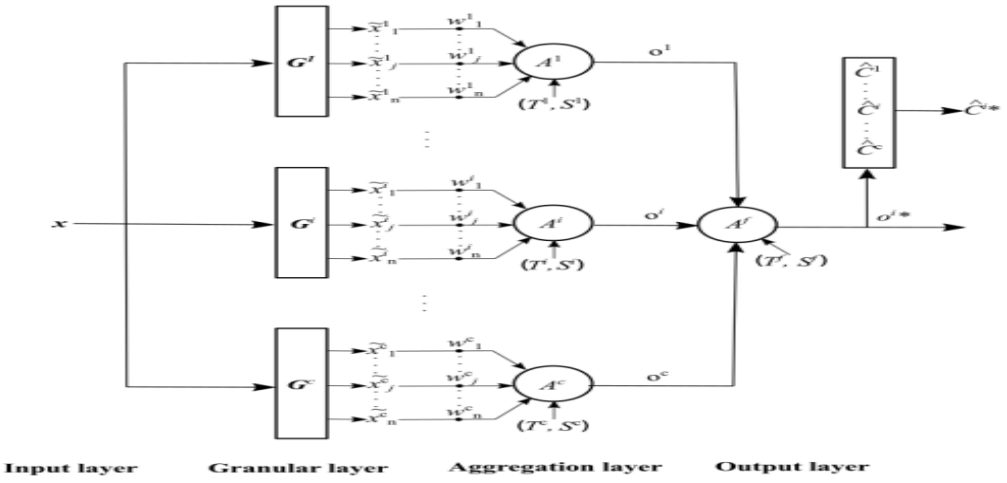

Figure 6. The structure of the classifying developing GNN

GNN processing of stream of data is based on the structure of the classifying developing GNN A quick incremental one-pass-through-the-data learning method. The GNN may have no idea about the statistical properties of the input and classes when it first starts learning. Fuzzy hyperboxes are used to granulate the feature space to set the boundaries of decisions in terms of classes.

An overview of GNN main features is given below

- forgets what is no longer relevant and modifies its parameters and structure to learn new ideas.
- addresses both labelled and unlabelled instances at the same time.
- Handle data uncertainty and identify drifts
- demonstrates the capacity for nonlinear separation
- uses both destructive top-down and constructive bottom-up techniques to foster lifelong learning.

a) GNN Structure and Processing

A stream of data $x[h], h = 1, 2, \ldots$ is used for learning GNN. A class label $C[h]$ may or may not be present with training examples. A class Ck of the finite collection of classes $C = \{C1, \ldots, Cm\}$ in the output space $Y \subseteq N$ is linked to each IG γ_i of a granules finite set $\gamma = \{\gamma 1, \ldots, \gamma c\}$ in the feature space $X \subseteq Rn$. From the data stream, and a layer of T-S neurons, granules are extracted, and these granules in the GNN link the feature and output spaces.

In essence, the input layer fans feature vectors into the network $x[h] = (x1, \ldots, xj, \ldots, xn)[h], h = 1, \ldots$ The collection of information granules $\gamma i \; \forall i$ that are created inside a feature space scope makes up the granular layer. Partially overlapped granules are permitted. Null neurons $TSni \; \forall i$ are included in the aggregation layer. To produce values $oi \; \forall i$ that show class compatibility between examples and granules, they aggregate membership values. The class $\bar{C}_k$ linked to the granule γ_i with the highest compatibility value is output after the decision layer compares the compatibility values o_i. The class label indicators are the output layer. Every layer changes as $x[h], h = 1, \ldots,$ is input, with the exception of the input layer.

Depending on the application scenario, the GNN classifier can be adapted structurally and parametrically in a number of ways. For instance, it is possible to automatically manage the number of classes when it is known beforehand.

If processing time and memory are limitations, the number of granules in the model structure can also be limited.

RESULTS AND DISCUSSION

Pest detection system processes images using MATLAB, that is, it calculates Gray-Level Co-occurrence Matrix (GLCM) feature, namely, ASM, IDM, Entropy, and Correlation. Deep learning architectures such as ResNet 50, Enhanced DenseNet201 and Granular Neural Networks (GNN) are incorporated in MATLAB to extract features and carry out a classification to effectively identify pests.

The Database that will be utilized in this survey is the IP102 dataset which is a collection of images of the commonly found pests in the agricultural environment. It consists of 102 classes of pests, and there is a wide range of images of pests of various stages and in different environmental conditions. This data is used in the training and testing of the proposed pest detection and classification models, where detailed performance appraisal is performed

on the various types of pests. The Adam optimizer was applied to the experiments with a learning rate of 0.001 to ResNet 50 and DenseNet 201, and 0.0005 to GNN. The batch size was 32 with ResNet 50 and DenseNet 201 as well as 64 with GNN. The maximum number of iterations was 1000, and the validation loss was used to check early stopping of 50 iterations with no improvement. The models were trained for 50 epochs (ResNet 50, DenseNet 201) and 40 epochs (GNN), with dropout rates of 0.5 for ResNet 50, 0.4 for DenseNet 201, and 0.3 for GNN.

The suggested HDL-MCSA method's performance is assessed by contrasting it with the current classifiers. Equations (48) – (51) provide certain statistical measurements that are used to evaluate the classifier in addition to classification ACC. The average results for the classifiers are also calculated.

P is defined as the proportion of correctly identified positive observations to all expected positive instances equation 48.

$$P = TP/TP+FP \quad (48)$$

R is defined as the proportion of correctly detected positive instances to all instances in the actual class equation 49.

$$R = TP/TP+FN \quad (49)$$

The F1 score is the weighted average of P and R. FN and FP are considered equation 50.

$$F1\ Score = 2*(R * P) / (R + P) \quad (50)$$

The positives and negatives of ACC are calculated as follows equation 51:

$$ACC = (TP+FP)/(TP+TN+FP+FN) \quad (51)$$

True Positive, False Positive, True Negative, and False Negative are the definitions of TP, FP, TN, and FN, respectively. For each of the suggested and standard classifiers, these parameter values are determined.

Table 1. Performance comparison outcomes among the suggested and current classifiers

Performance Metrics	**IMFR-CNN**	**FR-CNN**	**HDL-MCSA**	**HDL-MCSAWFC**	**DEL**
Accuracy	88	98.90	99.77	99.80	99.90
Precision	87.56	98	99.67	99.84	99.92
Recall	87.12	97.12	99.78	99.82	99.91
F-Measure	88.10	97.23	99.76	99.83	99.91

Table 1 summarizes the performance metrics of five models—IMFR-CNN, FR-CNN, HDL-MCSA, HDL-MCSAWFC, and DEL-based on Accuracy, Precision, Recall, and F-Measure. DEL is the best in all measures, and the highest Accuracy (99.90%), Precision (99.92%), Recall (99.91%), and F-Measure (99.91), which was most effective in making correct predictions, in identifying relevant instances, and balances between Precision and Recall. Comparatively, IMFR-CNN has the lowest metrics of all metrics especially on Accuracy (88%) and Recall (87.12%), whereas HDL-MCSAWFC and HDL-MCSA have quite high metrics with scores of about 99%. This highlights DEL as the most efficient model among those evaluated.

Figure 7 displays the performance of the suggested DEL's precision comparison findings. Therefore, the results confirm that the classification of rice diseases may be accurately predicted by FS utilising Hybrid EWHO. As a result, the performance of the jointly learnt linear transformation is not significantly impacted by the number of helpful features in the suggested DEL. The ability to avoid having to agonisingly adjust the regularisation parameter in the classifier makes it an enticing feature. The suggested DEL solves the categorisation problem in a very efficient manner.

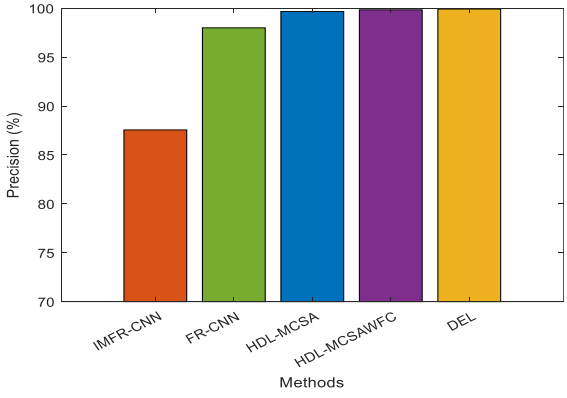

Figure .7. Precision comparison outcomes of the suggested model with the current classifiers

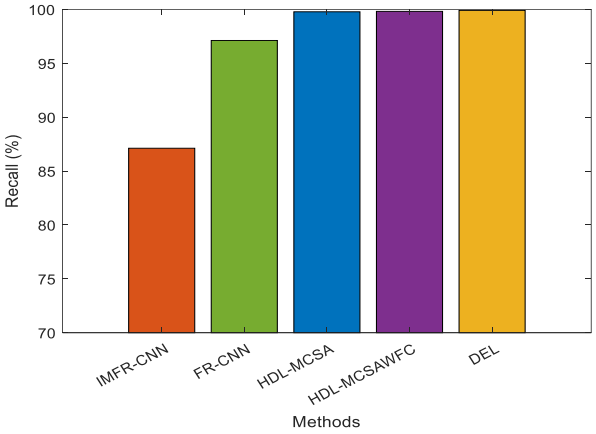

Figure 8. Recall comparison outcomes of the suggested HDL-MCSA with the current classifiers

The performance outcomes of the suggested algorithm are shown in figure 8. Unlike the existing technique (which produces lower results of R e.g., HDL-MCSAWFC method metric of 99.82%, HDL-MCSA method metric of 99.78%, FR-CNN method metric of 97.12% and IMFR-CNN method metric of 87.12%). The new DEL approach produces high results of R of 99.91%.

In comparison to the current approaches, the suggested DEL classifier performs remarkably well in terms of the disease prediction rate, as seen in figure 9. The qualitative analysis utilising EL and the quantitative analysis results in terms of F-measure concur. The ACC of the proposed DEL to the pest dataset is evaluated against the state-of-the-art classification methods. The proposed framework has better f-measure than the existing ones and it was illustrated in graph.

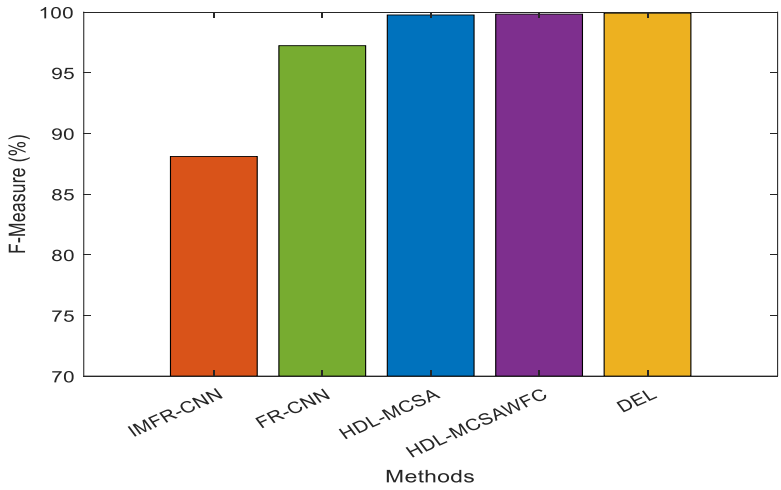

Figure 9. F-measure comparison outcomes of the suggested DEL with the current classifiers

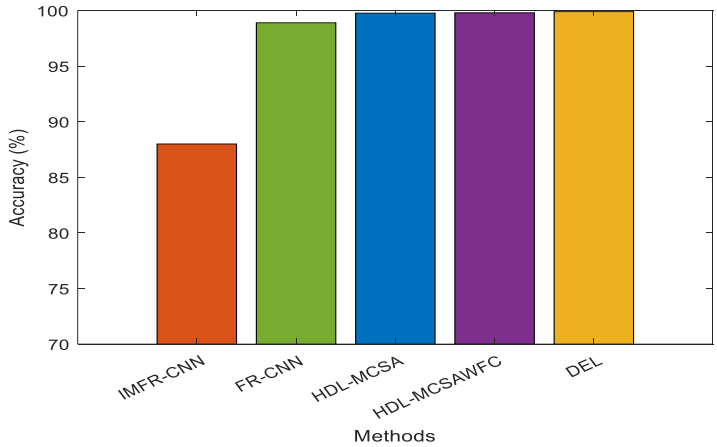

Figure 10. ACC comparison outcomes of the suggested model with the current classifiers

The proposed DEL-based ensemble classifiers are more effective in comparison with the existing classifier in terms of ACC, which can be observed in Figure 10. Like the DEL classifier that illustrates the effectiveness of the

method in dynamic environments to classify pest diseases, all the above classifiers do not work well when applied in the static environment. Hence, the accuracy of the classifiers will be superior as opposed to ensemble classifiers that are built on an existing model.

CONCLUSION

Contamination due to pest infestations is a leading cause of agricultural and financial losses worldwide. Automated identification of invasive insects could significantly accelerate pest detection and control. This research introduces an intelligent Feature Extraction (FE) and Deep Ensemble Learning (DEL) model for efficient pest detection. Initially, preprocessing using Z-score normalization (ZSN) removes noise, improving classifier performance. The Improved Feature Weighted Fuzzy Clustering (IFWFC) method is then applied for effective dataset grouping, followed by the Henon Chaotic Map Encryption (HCME) method for image encryption. The Hybrid Enhanced Wild Horse Optimization (EWHO) with Gray-Level Co-occurrence Matrix (GLCM) approach is employed for feature extraction, and Modified Cuckoo Search Algorithm (MCSA) is used for feature selection. Lastly, DEL model incorporates the ResNet50, Enhanced DenseNet201 and Granular Neural Networks (GNN) to boost the performance of the classifier. The model is tested on the IP102 dataset with excellent results: an accuracy (ACC) of 99.90, precision of 99.92 and recall (R) of 99.91. Such results are better than the existing Ensemble Deep Learning (EDL) models, which show the efficiency of the proposed detection model. The accuracy and consistency are explained by the fact that the model produces non-redundant rule sets that improve its capacity to identify the various types of pests. To increase the usefulness of the model in precision agriculture, it may be beneficial to expand the classification size in the future to add additional categories of pests in the model. This study has a great potential in reducing financial and crop yield losses as the experts and farmers can quickly and effectively detect pests in order to enhance the pest control measures.

REFERENCES

[1] Chithambarathanu M, Jeyakumar MK. Survey on crop pest detection using deep learning and machine learning approaches. Multimedia Tools and Applications. 2023 Nov;82(27):42277-310. https://doi.org/10.1007/s11042-023-15221-3

[2] Domingues T, Brandão T, Ferreira JC. Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey. Agriculture. 2022 Sep 1;12(9):1350. https://doi.org/10.3390/agriculture12091350

[3] Attri I, Awasthi LK, Sharma TP. Machine learning in agriculture: a review of crop management applications. Multimedia Tools and Applications. 2024 Feb;83(5):12875-915. https://doi.org/10.1007/s11042-023-16105-2

[4] Liu J, Wang X. Plant diseases and pests detection based on deep learning: a review. Plant methods. 2021 Feb 24;17(1):22. https://doi.org/10.1186/s13007-021-00722-9

[5] Lin TL, Chang HY, Chen KH. The pest and disease identification in the growth of sweet peppers using faster R-CNN and mask R-CNN. Journal of Internet Technology. 2020 Mar 1;21(2):605-14.

[6] Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors. 2017 Sep 4;17(9):2022. https://doi.org/10.3390/s17092022

[7] Rahman CR, Arko PS, Ali ME, Khan MA, Apon SH, Nowrin F, Wasif A. Identification and recognition of rice diseases and pests using convolutional neural networks. Biosystems Engineering. 2020 Jun 1;194:112-20. https://doi.org/10.1016/j.biosystemseng.2020.03.020

[8] Deepika P, Arthi BJ. Prediction of plant pest detection using improved mask FRCNN in cloud environment. Measurement: Sensors. 2022 Dec 1; 24:100549. https://doi.org/10.1016/j.measen.2022.100549

[9] Saleem MH, Potgieter J, Arif KM. Plant disease detection and classification by deep learning. Plants. 2019 Oct 31;8(11):468. https://doi.org/10.3390/plants8110468

[10] Demilie WB. Plant disease detection and classification techniques: a comparative study of the performances. Journal of Big Data. 2024 Jan 2;11(1):5. https://doi.org/10.1186/s40537-023-00863-9

[11] Türkoğlu M, Hanbay D. Plant disease and pest detection using deep learning-based features. Turkish Journal of Electrical Engineering and Computer Sciences. 2019;27(3):1636-51. https://doi.org/10.3906/elk-1809-181

[12] Tang Z, Chen Z, Qi F, Zhang L, Chen S. Pest-YOLO: Deep image mining and multi-feature fusion for real-time agriculture pest detection. In2021 IEEE International Conference on Data Mining (ICDM) 2021 Dec 7 (pp. 1348-1353). IEEE. https://doi.org/10.1109/ICDM51629.2021.00169

[13] Arun RA, Umamaheswari S. Effective and efficient multi-crop pest detection based on deep learning object detection models. Journal of Intelligent & Fuzzy Systems. 2022 Aug 10;43(4):5185-203.https://doi.org/10.3233/JIFS-220595

[14] Wang QJ, Zhang SY, Dong SF, Zhang GC, Yang J, Li R, Wang HQ. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. Computers and electronics in agriculture. 2020 Aug 1; 175:105585. https://doi.org/10.1016/j.compag.2020.105585

[15] Ali F, Qayyum H, Iqbal MJ. Faster-PestNet: A Lightweight deep learning framework for crop pest detection and classification. IEEE Access. 2023 Sep 20; 11:104016-27. https://doi.org/10.1109/ACCESS.2023.3317506

[16] Jiao L, Dong S, Zhang S, Xie C, Wang H. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. Computers and Electronics in Agriculture. 2020 Jul 1;174:105522. https://doi.org/10.1016/j.compag.2020.105522

[17] Wang R, Jiao L, Xie C, Chen P, Du J, Li R. S-RPN: Sampling-balanced region proposal network for small crop pest detection. Computers and Electronics in Agriculture. 2021 Aug 1; 187:106290.

[18] Cheng Z, Huang R, Qian R, Dong W, Zhu J, Liu M. A lightweight crop pest detection method based on convolutional neural networks. Applied sciences. 2022 Jul 22;12(15):7378. https://doi.org/10.3390/app12157378

[19] Zhang W, Huang H, Sun Y, Wu X. AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning. Frontiers in Plant Science. 2022 Dec 16; 13:1079384. https://doi.org/10.3389/fpls.2022.1079384

[20] Caner A, Ali M, Yıldız A, Hanım E. Improvements in environmental monitoring in IoT networks through sensor fusion techniques. Journal of Wireless Sensor Networks and IoT. 2025;2(2):38-44.

[21] Dong S, Du J, Jiao L, Wang F, Liu K, Teng Y, Wang R. Automatic crop pest detection oriented multiscale feature fusion approach. Insects. 2022 Jun 18;13(6):554. https://doi.org/10.3390/insects13060554

[22] Boissard P, Martin V, Moisan S. A cognitive vision approach to early pest detection in greenhouse crops. computers and electronics in agriculture. 2008 Jul 1;62(2):81-93. https://doi.org/10.1016/j.compag.2007.11.009

[23] Kumar TS. Context-Aware Learning Mechanisms for Self-Organizing Large-Scale Sensing Environments. National Journal of Ubiquitous Computing and Intelligent Environments. 2025 Sep 25:8-13.

[24] Gondal MD, Khan YN. Early pest detection from crop using image processing and computational intelligence. FAST-NU Research Journal. 2015 Jan;1(1):59-68.

[25] Fei N, Gao Y, Lu Z, Xiang T. Z-score normalization, hubness, and few-shot learning. InProceedings of the IEEE/CVF International Conference on Computer Vision 2021 (pp. 142-151).

[26] Yang Q, Huang H, Zhang J, Gao H, Liu P. A collaborative cuckoo search algorithm with modified operation mode. Engineering Applications of Artificial Intelligence. 2023 May 1; 121:106006. https://doi.org/10.1016/j.engappai.2023.106006

[27] Alshammari A. Generation forecasting employing deep recurrent neural network with metaheruistic feature selection methodology for renewable energy power plants. Sustainable Energy Technologies and Assessments. 2023 Feb 1; 55:102968. https://doi.org/10.1016/j.seta.2022.102968

[28] Arvinth N. Trust-Preserving Distributed Learning Control Architecture for Safety-Constrained Cyber-Physical Systems. Transactions on Internet Security, Cloud Services, and Distributed Applications. 2025 Jun 20:38-45.

[29] Naruei I, Keynia F. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. Engineering with computers. 2022 Oct;38(Suppl 4):3025-56.

[30] Hashemzadeh M, Oskouei AG, Farajzadeh N. New fuzzy C-means clustering method based on feature-weight and cluster-weight learning. Applied Soft Computing. 2019 May 1; 78:324-45. https://doi.org/10.1016/j.asoc.2019.02.038

[31] Mishra K, Saharan R. A fast image encryption technique using Henon chaotic map. InProgress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2017, Volume 1 2018 Dec 18 (pp. 329-339). Singapore: Springer Singapore.

[32] Uvarajan KP. Physics-Constrained Uncertainty-Aware Learning Architectures for Resilient Context Inference in Pervasive Intelligent Systems. National Journal of Ubiquitous Computing and Intelligent Environments. 2025 Sep 25:36-44.

[33] Uvarajan KP, Karthika J. AI-Assisted Phenotyping of Leaf Morphology for Early Disease Detection in Grapevines. National Journal of Plant Sciences and Smart Horticulture. 2023 Dec 24:17-24.