The International Journal of Interdisciplinary Organizational Studies
ISSN: 2324-7649 (Print), ISSN: 2324-7657 (Online)
Volume 19, Issue 1, January-June, 2024

# A ROBUST FEATURE SELECTION AND HYBRID DL (DEEP LEARNING) MODEL FOR THE RECOGNITION OF AGRICULTURAL PESTS IN CC (CLOUD COMPUTING) SYSTEM

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

1*Research Scholar, Department of Computer science, Kongunadu Arts and Science College, Coimbatore-641029, Email: bharathipadma19@gmail.com

2Associate professor, Department of Information technology, Kongunadu Arts and Science College, Coimbatore-641029, Email: dhanalakshmik_cs@kongunaducollege.ac.in

**Abstract**: It is well recognized that one of the main factors causing harm to agricultural crops of economic importance is insect pests. The availability of food and a stable agricultural sector depend on accurate agricultural pest prediction, which is largely dependent on the classification of insect pests. Insect pest recognition is time-consuming and expensive since it mostly depends on the special expertise of agricultural specialists due to a wide variety of species of pests and tiny variations between species. This research work, propose a robust FS (Feature Selection) and HDL (Hybrid Deep Learning) model for the efficient recognition of pests using IP102 dataset in CC (Cloud Computing) structure. Initially, the preprocessing is done by Z-score normalization for removing the noises for improving the classifier's execution. Second, the FS is applied by the technique depends on MCSA (Modified Cuckoo Search Algorithm). And finally, the Hybrid DL model is suggested for the efficient recognition of pests. Here the Granular Neural Network (GNN) is hybridized with the Faster R-CNN (Region-Convolutional NN) for the improved execution of the classifier. Thus the result demonstrate that the suggested recognition technique is implemented with the IP102 dataset and which is evaluated with the metrics as accuracy, precision, recall and F-measure. As a result of analysis, it is identified that the suggested technique has high execution results than the existing recognition models.

**Keywords:** Agricultural pests, preprocessing, feature selection, Z-score normalization, Modified Cuckoo Search Algorithm (MCSA), HDL (Hybrid Deep Learning) model and Granular Neural Network (GNN).

## 1. Introduction

One of the crucial issues facing the farming sector is insect pests. The FAO estimates that these crops reduce total food output by 20–40% annually [1]. These pesticides are directed against environment adulteration and high-risk illnesses like cancer, hereditary diseases, major respiratory disorders, and preterm birth. Advanced technical solutions are extensively predicted for agriculture for avoiding unnecessary application of planticides and to classify plant diseases earlier. Harvesting harmed by insects suffer from a variety of illnesses and reduced yields [2]. One of the main causes of yield losses is insect damage to areas used for harvest, including rice, wheat, and beans. To limit crop losses caused by insects, decrease insect numbers, and prevent insect populations from spreading to large areas, biocontrol measures such as pesticides should be employed. Chemicals and pesticides play a major role in pest management. However, they will negatively impact the ecosystem and human health in a number of ways [3]. Consequently, experts studying agriculture across the globe started collaborating to create more effective alternatives to chemical pesticides for controlling pests.

By solving and minimizing these problems, the industry as a whole would expand economically and produce a lot more food using less natural resources [4]. Furthermore, recognizing the insect is crucial since different pest species require different approaches to pest management. The first line of protection against insect pests damaging crops is the ability to identify and categorize insects, to tell the difference between harmful and healthy ones. But classifying insects is a risk work due to its intricate architecture and the similarities among different insect species [5]. Traditionally, entomologists have identified insects by hand, which takes a lot of effort and requires a deep comprehension. The usage of automatic pest classification has grown in popularity recently due to the requirement for continuous and expensive control of this activity. Additionally, experts have utilized a range of CAD classification strategies to resolve those issues [6]. It incorporates some AI, technologies for wireless communication, CC, and the Internet of Things (IoT) into every step of agriculture. Crop health monitoring, that describes the farm's situation in the consideration of crop disease, is the more significant application of smart agriculture.

Furthermore, the usual manual method of identifying insect infestations is time-consuming, costly, and ineffectual [7-8]. Nevertheless, farmers can avert this insect feast by using the appropriate pesticides and paying attention to infected shrub plants early on. Super computer prophesy, IP (Image Processing), and ML (Machine Learning) techniques are employed to shed light on the aforementioned issues facing the agriculture sector. A number of ML-computerized pest classification techniques have recently been introduced [9]. Conversely, conventional machine learning methods have several shortcomings. If the amount of crop pest species is modest, traditional ML algorithms have been demonstrated to perform effectively; but, when different features need to be directly collected, they lose their effectiveness. They require an additional, crucial level of data pre-processing called FE (Feature Engineering). Furthermore, it has a limited ability to generalize between datasets. Additionally, the quality of the data they have access to determines how effective they are [10–11]. For instance, a small dataset leads to low accuracy, but growing the dataset doesn't significantly improve efficiency after an appropriate level of accuracy is reached. Regarding the classification of insects, there are similar problems [12–13]. As the input dataset contains images, DL models in particular, CNN's DL algorithm can resolve these problems. DL is a subset of ML in which deep features are automatically extracted by MLL (Multilayer Neural Networks). In agriculture, DL- algorithms have gained popularity for identifying weeds, recognizing plants, and identifying plant diseases [14–15]. Another field in which the CNN algorithm performed better than conventional ML methods was in insect classification. Recently, a number of DL algorithms have been applied to recognize pests, producing cutting-edge outcomes in a range of insect detection applications.

The remaining study is ordered as: Section 2 examines some of the most modern methods for detecting pests in order to increase agricultural productivity. The suggested methodology's approach is presented in section 3. The test outcomes and the description are given in section 4. Future tasks and the conclusion are covered in section 5.

**Literature Review**

Some of the most recent methods for identifying pests utilizing sophisticated DM (Data Mining) and ML algorithms in this section.

In order to identify agricultural pests in an application that can be easily installed in mobile terminals, Yang et al. [16] created an intelligent mobile-terminal recognition technique depends on DNN (Deep Neural Networks). In this study, more rich semantic data was created, FMs (Feature Maps) that can successfully enhance the SqueezeNet framework at a cheap computational rate by connecting with the extensive features of the low-level network characteristics and the rich semantic data related to the characteristic of high-level networks. In comparison to previous standard CNNs (Convolutional Neural Networks), experiments show that the suggested method offers a significantly greater 91.64% recognition accuracy having low training period, all while staying within a constrained computational limit. In the open insect data IP102, researchers have further confirmed that the enhanced SqueezeNet algorithm has a 2.3% higher performance as compared to the new algorithm. A pest surveillance device for agriculture was created by FAISAL et al. [17] utilizing ambient sensors, a pi camera, and DL. The created system has the ability to track and quantify the quantity of pest insects that adhere to the yellow sticky notes on a continual basis. The primary crop planted at the research's location was fruit crops, so environmental data on humidity and temperature were also collected there. The results of the research indicated the biggest relationship between insect pests and humidity. In summary, the technique that has been built has the ability to efficiently gather insect counts through automatic means. This yields valuable insights for optimizing pest management practices during crop cultivation.

An AI-pest identification method was presented by Chen et al. [18] to address the particular problem of scale pest detection from photos. Scale pests in the image are identified and localized using DL- ODM (Object Detection Models), including You Only Look Once v4 (YOLO v4), SSDs (Single-Shot multibox Detectors), and Faster R-CNNs (Region-CNNs). According to the testing outcomes, YOLO v4 outperformed the other techniques by the classification accuracy, scoring 100% for mealybugs, 89% for Coccidae, and 97% for Diaspididae. In the meantime, YOLO v4's computational efficiency indicates that it is appropriate for actual time execution. Furthermore, the YOLO v4 algorithm's prediction outcomes benefit the user even more. Utilizing the trained scale pest recognition approach, a mobile application has been created to make pest recognizing in farms easier. This helps farmers apply suitable pesticides at the right times to minimize crop damage.

An embedded system with ML enhancements was described by Albanese et al. [19] to guarantee ongoing infestations of pest’s detection within fruit plantations. To demonstrate the possibilities of the system, three distinct ML algorithms have been trained and implemented.

Furthermore, because energy harvesting functions are integrated, the suggested method ensures a longer battery life. The difficulty of automating insect infestation for an infinite amount of time with the lack of farmer's assistance is demonstrated by the outcomes.

For the detection of pests in rice throughout its cultivation in agriculture, Bhoi et al. [20] offer an IoT aided UAV (Unmanned Aerial Vehicle) based rice pest recognition system utilizing Imagga cloud. The AI (Artificial Intelligence) system and Python programming language are the main features of the IoT-assisted UAV. They are used to spread rice pest photographs to the Imagga cloud and pest data was provided. By comparing the trust levels with the tags, the pest was recognized by the Imagga cloud. The thing in that picture is symbolized by the tag. For recognizing the pest, the tag that has the highest probability value and is above the threshold is chosen as the target identifier. If a pest is found, the owner is notified so they can take precautions. The suggested approach is capable to recognize any type of pest that damages rice when it has been produced. As an alternative, this study makes an effort to reduce rice waste throughout manufacturing by routinely checking for pests.

The initial step towards an entirely functional, semantically improved DSS (Decision Support System) for IPM (Integrated Pest Management) was provided by Rodríguez-García et al. [21]. The main purpose is to generate a comprehensive agricultural information base through compiling information from various, various sources and to create a framework that will help farmers make decisions about managing diseases and pests. After evaluation in a simulated setting, the pest classifier framework yielded an overall accuracy of 98.8%.

A lightweight AI powered solution that can be loaded onto a smartphone is offered by Tiwari et al. [22] and lets farmers recognize and categorize target insects on selected crops (soybean) in in real time. The dataset was used to create the DL models VGG16 and GoogLeNet, which both performed admirably (0.0852- validation loss and 97.78% validation accuracy) and (69.5098- validation loss and 99.03% validation accuracy). Ultimately, an app for Android is created to enable actual processing time.

To assist specialists and farmers, Karar et al. [23] unveiled an innovative smartphone application that uses DL to automatically classify pests. Depending on CC, the developed application uses a Faster R-CNN to complete the work of insect pest recognition. In order to support agriculturalists, a database of optional insecticides is also connected to agricultural pests that have been identified. The 5 pest groups included in this study :Aphids, Cicadellidae, Flax Budworm, Flea Beetles, and Red Spiders were efficiently verified. For every evaluated pest image, the suggested Faster R-CNN produced the more precise detection outcomes, achieving 99.0%. Furthermore, the DL approach operates better than other earlier recognition techniques namely, conventional BP (Back Propagation) NNs and SSD MobileNet expertise. The main purpose of this research is to implement our developed online program for agricultural pest identification in open fields, including big farms and greenhouses for particular crops.

With the aid of a cutting-edge mobile application, Deepika et al. [24] presented a DL for specialists and farmers for controlling the insects that harm plants. The IMFR-CNN (Improved

MaskFaster Region-Based Convolutional Neural Network) technique has been presented as a means of creating cutting-edge application exploitation for identifying insect plant exhaust in CC. The SSD DL approach and BP are two current techniques to recognize plant infection at initial phase. The suggested IMFRCNN system is assessed and delivers 99.0% greater accuracy for all verified crop image.

A DL method called You Only Look Once (YOLO) was proposed by Vamsi et al. [25] for assessing crop images and accurately identify pest infestation and harm patterns. This pest identification is possible with this methodology, which also alerts farmers to the need for prompt pest treatment. With an accuracy of 87%, the technique can increase the efficiency of managing and identifying pests while reducing the need for human labor, raising crop yields, and strengthening the availability of food.

In order to classify the existence of various crop pests, Venkatasaichandrakanth et al. [24] developed ML and DL algorithms, each having advantages and disadvantages. When compared to ML, DL is a preferable solution because of the availability of the present crop pest image data sources. It illustrates the potential future extent of work that could be completed in response to the flaws found in the literature review.

## 2. Proposed Methodology

As seen in Fig. 3, a brand-new mobile application has been created to recognize crop pests in the cloud. Utilizing the IP102 dataset in a CC system, a HDL approach with strong FS is used to effectively identify pests. The 3 primary modules that make up the structure of the created DL-CC systems are as follows: they enable the automatic detection and classification of insect infestations. First, ZSN (Z-Score Normalization) is used for pre-processing for reducing noise and develop the accuracy of the classifier. Second, the MCSA (Modified Cuckoo Search Algorithm)-based technique is used to perform FS. Lastly, a HDL system is suggested for effective pest recognition. In this case, the R-CNN is hybridized with the GNN (Granular Neural Network) to enhance the classifier's efficiency.

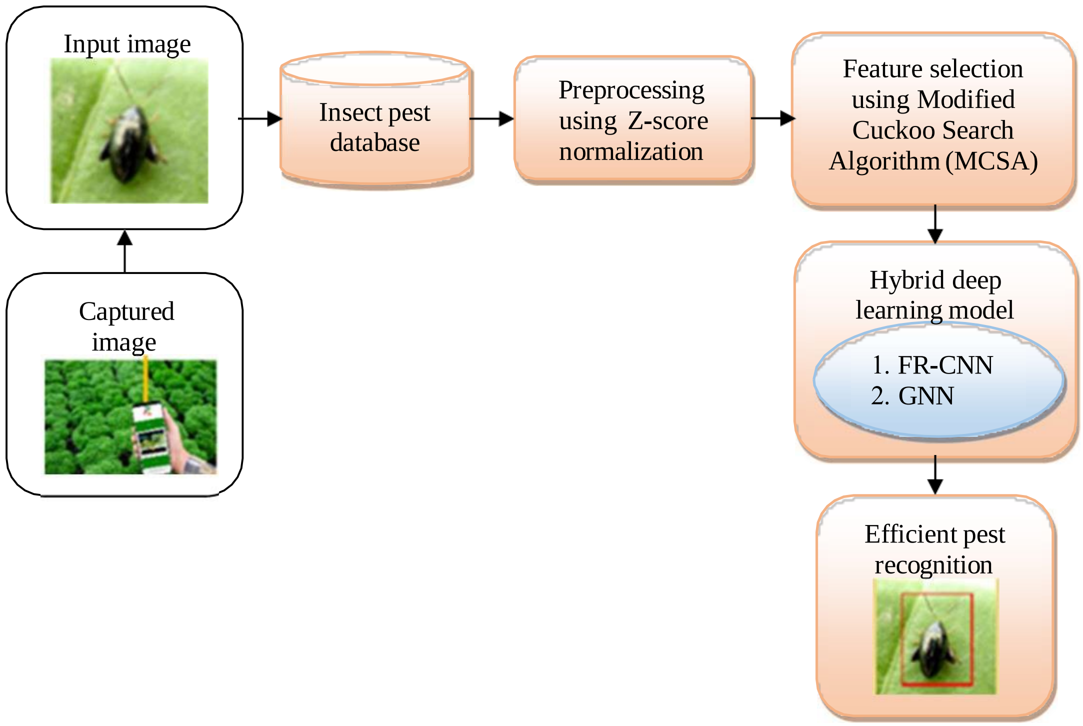


Figure 1. Schematic diagram of the recognizing crop pests using proposed HDL-MCSA.

### 3.1. Insect pests

As seen in Figure. 1, this research covers crop pests into eight types: aphids, flea beetles, Cicadellidae, flax budworm, red spider and Grain spreader thrips, Black cut worm, Nipaecoccusvastalor. They are the most common agricultural pests, particularly in warm climates. A brief definition of each insect pest is as follows:

**Aphids**: In agricultural areas with high temperatures, aphids are among the most severe insect pests on crop development, including potatoes [27]. They have the ability to spread infections from an infested plant to a normal one, causing harm and a notable decrease in crop yield.

**Flea beetles**: Major insect pests that damage harvests in North America and Europe are flea beetles [28]. 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 [29].

**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 greenhouses or open fields, causing major harm [30].

**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 [31].

**Black cut worm:** The host range of black cutworm is broad. This species can eat almost every vegetable, and it also occasionally eats crops and grasses, alfalfa, clover, cotton, rice, sorghum, strawberries, and sugar beets [32].

**Nipaecoccusvastalor:** Another name for Nipaecoccus vastalor is the spiked mealy bug. Although the term refers to the pest's broad host range, the most frequent name for it is coconut mealybug. The coconut mealy insect has the potential to cause palm trees and tropical fruit harvests economic harm [33].

### 3.2. User interface module

Using the support of a smartphone's created (UI) User Interface, an agriculturalist in the wide area or a greenhouse could quickly recognize crop pests on any platform, such as iOS or Android-based strategies. 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.

### 3.3. Preprocessing using Z-Score normalization

The average intensity for each dataset was determined first, followed by the average of the SD (Standard Deviations), in order to normalize the raw intensity information for every study [34]. The estimation of normalization variables, which were subsequently assigned to each test, was based on this total average. The grand average was then equal to an average of all the normalized information. Plotting a ZSN curve is possible. Z-scores are located among 3 SDs (Standard Deviations), meaning that a Z-score would set the individual considerably to the left or considerably to the right of the NDC (Normal Distribution Curve). Users need to have knowledge of the mean (μ) and the population SD (σ) in order to calculate a z-score.

Let xi (i = 1, 2,….,D) represent, in specific terms, the i-th component of every FV (Feature Vector) x ∈ R D. Initially, calculating these D components' means and SDs:

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

Next, Z-score normalization is implemented as

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

These computations indicate that the ZSN process first projections the initial FVs (Feature Vectors) on the 1 vector to the hyperplane 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 feature selection process is carried in which is described in the below section.

### 3.4. Feature selection using Modified Cuckoo Search algorithm (MCSA)

This section proposes a modified CS (Cuckoo Search) algorithm that incorporates opponent learning for enhancing 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.

**Basic CS Algorithm**

The fundamental CS method depends on the fact that certain species of cuckoo parasitize other cuckoos by putting their eggs in their nests. To make the basic CS easier to understand, these 3 optimal criteria are applied: A single egg is laid by a cuckoo at a time [35], which it deposits in a set that is selected arbitrarily; (2) the best nests through superior eggs may be approved to the following groups; (3) the amount of host nests that are obtainable has been set, and the host bird determines the cuckoo's laid egg via probability of $p_a \in [0, 1]$. In this state, the host bird has two options: it neglect the egg and construct a more elaborate one.

The probability represents the impact of each generation's replacement of eggs of cuckoo found via the host bird having fresh eggs. Recall that a solution is symbolized by an egg. As a selection procedure for the technique of optimization, these presumptions guarantee that the best solutions will remain from generation to generation. Therefore, the objective of the CS algorithms is to swap out the low-quality solutions in the nests for higher-quality ones. For cuckoo *i*, a new solution $x_i^{t+1}$ is provided by:

$$x_i^{t+1} = x_i^t + \alpha \otimes L'evy(\lambda) \qquad (3)$$

$$\alpha = \alpha_0 \otimes (x_j^t - x_i^t) \qquad (4)$$

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 random walks can be indicated as *Lévy*(λ ). The value of α0 is set to 0.01 as suggested for improving the search effectiveness. One type of random walk where the step length is determined by the Lévy distribution is called a "Lévy flight." A series of instantaneous jumps produced by a probability density function with a power law tail provided by characterizes this distribution.

$$L'evy(\lambda) \approx u = t^{-\lambda}, (1 < \lambda \leq 3) \qquad (5)$$

A uniform distribution that complies with the Lévy distribution is used to determine the step length *S* of Lévy flights. Additionally, the method employed a balanced blend of global exploratory random walks and local random walks, which were managed by $p_a$ the switching parameter. One way to express the local random walk is as

$$x_i^{t+1} = x_i^t + \alpha\epsilon \otimes H(p_a - s) \otimes (x_j^t - x_k^t) \qquad (6)$$

Here, the Heaviside function can be symbolized as $H$, a arbitrary amount taken from a uniform distribution is s, 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 $\epsilon$.

Conversely, Lévy flights are employed to execute the global random walk:

$$\square_i^{t+1} = \square_i^{t} + \alpha \oplus \square'\square\square\square(\square, \square) \quad (7)$$

□ scaling factor in this case is denoted by α > 0, and the step-lengths are represented by L´evy(□, □), which are distributed in accordance with the probability distribution depicted in (4.17), which has an infinite mean and infinite variance:

$$\square'\square\square\square(\square, \square) = \frac{\lambda\Gamma(\lambda)\sin\left(\frac{\pi\square}{2}\right)}{\pi} \frac{1}{s^{1+\square}} \quad (8)$$

- **Uniform distribution strategies**

Where $\square_i^t$ yet indicates a solution arbitrarily chosen from the existing superior solutions. Hence in the procedure of maintaining equilibrium amongst local and global search process, uniform distribution technique plays a significant part. The uniform distribution strategy establishes the contribution quantity of previous velocity to its novel velocity at the present interval. 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 (9) of velocity update is assessed, it is apparent that this equation is comprised of two segments. The first entity ($\square_i(\square)$) indicates the population and the second entity (($\square_i(\square) - \square\square\square\square\square$)) influences the □th position by the direction of □□□□□,the global best solution. For generating 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:

$$\square_i(\square + 1) = \square(\square_i(\square)) + (\square_i(\square) - \square\square\square\square\square)\square_i\square_1 + (\square_i(\square) - \square_k(\square))\square_i\square_2 \quad (9)$$

$$\square_1 + \square_2 = 1 \quad (10)$$

The uniform distribution factor U stabilizes both the global and the □th solution's local search intensity; the superior solutions arbitrarily selected from the population ($\square \neq \square$) can be represented as $\square_\square$; the self-adaptive learning factor of the global best solution is denoted as $\square_1$, varying between 0 to 1; consequently, the learning factor of the □th solution is $\square_2$, varying between 1 to 0. By using the information from the □th solution to guide the □th solution, the local minima may also be able to prevent in this procedure. The global best solution's influence increases with $\square_1$, surpassing that of the ($\square_\square$) -□th neighbor solution is expressed as follows:

$$\square_1 = 1 + (\square_{init} - 1)\left(\frac{iter_{max} - it}{iter_{max}}\right)^n, \quad (11)$$

Here, the maximum amount of iterations can be denoted as $iter_{max}$, the present amount of iterations can be represented as iter, and the data of a nonlinear modulation can be indicated as □. the initial impact factor of $\square_1$ is $\square_{init}$. As the iteration is raised, $\square_1$will likewise increase nonlinearly from □init to 1, whereas $\square_2$ will decrease proportionally from $(1 - \square_{init})$ to 0. Cuckoos are allowed to fly over the search space having a small $\square_1$ and a large $\square_2$, even if they fly towards it. On the other hand, at the last stages of the search process, a big $\square_1$ and a small $\square_2$ allow the cuckoos to arrive to the global optimum solution. Therefore, at the final stage of optimization, and the suggested technique could efficiently regulate the global search and improve its convergence to achieve the global best solution.

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

$$\square = \square_{\max} * \exp\left(-\square * \left(\frac{\delta}{\delta_{\max}}\right)^{m}\right), \quad (12)$$

Here, $iter_{max}$ denotes the overall iteration amount, iter represents the existing count of iterations, values of maximal distributed value are indicated by $\square_{max}$, and □ is a constant greater than1. The advantage of the proposed MCSA is that it play a role in the solution's dispersion to the search space. Moreover, further precise results can be attained.

Algorithm 1: Modified Cuckoo Search Algorithm (MCSA)

**Input:** pest dataset
**Output:** Optimized Features
(1) Set the parameter values of the algorithm, create the random initial vector values and fix the iteration number □ = 1.
(2) Calculate fitness values of each individual and find the current best individual through the best objective value. Identify whether the stopping criterion is met. If the stopping criterion is met, then output the best solution; else update the iteration number □ = □ + 1 and continue the iteration procedure until the stopping criterion is met.
(3) Retain the best solution of the last iteration, and obtain a set of new solutions □□□□ = $[\square_1^{(t+1)}, \ldots, \square_i^{(t+1)}, \ldots \square_K^{(t+1)}]$ □□ L´evy
(4) Calculate the fitness value $\square_i^{(t+1)}$ of the new solution $\square_i^{(t+1)}$, and compare $\square_i^{(t+1)}$ with $\square_i^{(t)}$ which describes the solution of the □th iteration.
(5) A fraction ($\square_\square$) of worse nests are unrestricted and new ones are constructed.
(6) Retain the best solution.
(7) Search for a new solution by means of uniform distribution approaches.
(8) Retain the best nest with quality solution
(9) Rank the nests and determines the current best one
(10) Pass the current best nest to the subsequent generation
(11) Go to stage (2)
End

### 3.5. Classification using Hybrid deep learning model (HDL)

The HDL based classification technique is proposed for predicting the pests efficiently. Here the granular neural network (GNN) is hybridized with the Faster Region based CNN model for efficient recognition of pests.

#### 3.5.1. Faster R-CNN approach

The latest developments in R-CNN and Fast R-CNN are presented by the Faster R-CNN. In order to anticipate object limits, it combines a Region Proposal Network (RPN) with a Faster R-CNN for target recognition in processing pictures. As seen in Fig. 2, the function of the RPN module is to act as the combined Faster R-CNN's "attention" process. The Faster R-CNN's three fundamental parts can be explained as follows [26]. Initially, the input image is sent into the FE network, which creates FMs (Feature Maps) (see Fig. 2). It could be an CNN structure that has already been trained, like Dense Convolutional Network (Densenet), Residual NN (Resnet), or Inception. In addition, the RPN module suggests where the FMs' objects should be located. Third, to modify these suggested locations and identify multi-object classes or single object class through the relevant bounding box region in the final picture, as seen in Fig. 2, a regressor and classifier is trained by employing the loss function L in (13) for the CNN identification net.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (13)$$

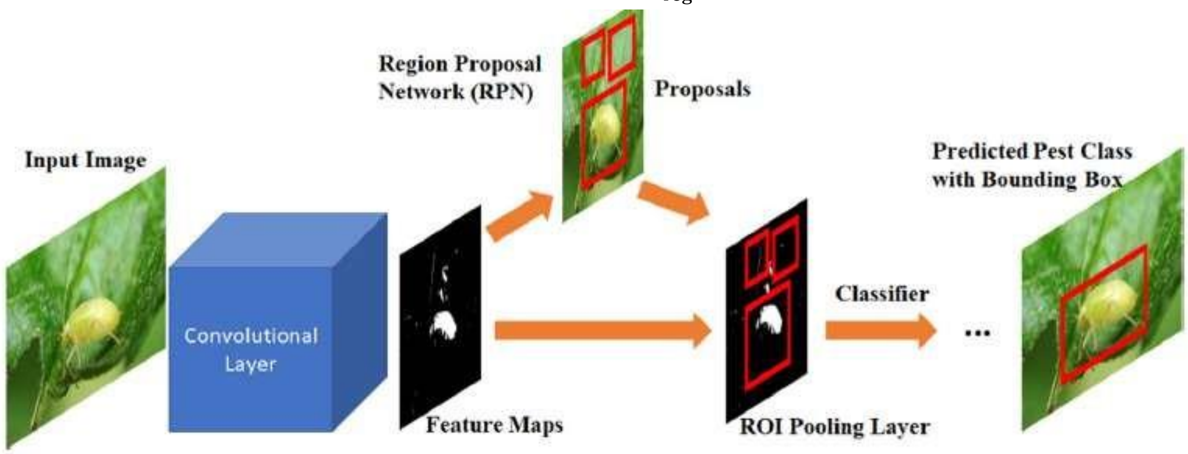


Figure 2. The simple workflow of Faster R-CNN for detection and classification of target object class, i.e., crop pests

The projected chance of all anchor i appearing as object image is denoted by $p_i$ in this case. The actual label for an anchor is denoted by $p_i^*$, where $p_i^* = 1$ for positive anchors and 0 for negative anchors. The projected object's bounding box's parameterized coordinates are represented by the $t_i$ vector, and the bounding box connected to the positive anchors' ground-truth is represented by $t_i^*$. The classifier and regression L are denoted by $L_{cls}$ and $L_{reg}$, accordingly. Only positive anchors cause the regression loss $p_i^*$ Lreg to become active. Considering a balanced parameter $\lambda$, the weighted factors $N_{cls}$ and $N_{reg}$ correspond to the normalized two terms.The outputs of the regression and classification layers are denoted by $\{t_i\}$, $\{p_i\}$, respectively. But because of the size of the dataset, the network was unable to be fully trained. For resolving the issue, a hybrid algorithm was presented in the current study. This paper introduces a technique based on GNN (Granular Neural Networks) for optimizing network parameters and enhancing classifier performance.

### 3.5.2. Granular Neural Network (GNN)

Starting with a specific previously created (numeric) NN, the framework of a GNN and the corresponding learning scheme are formed by adding granular connections that cross over the numeric weights (connections) to the network's architecture [36]. They are focused on an MLP, one of the NN architectures that is most frequently employed and has an extensive range of learning strategies.

### A. Interval Operations

Before delving into the specifics of GNNs , let's review a few interval mathematical concepts results. X = [a, b], Y = [c, d],and so on are samples of intervals that represent the arguments (variables). The mathematical operations are described below:

$$\square\square\square\square\square\square\square\square: \square + \square = [\square + \square, \square + \square] \quad (14)$$

$$\square\square\square\square\square\square\square\square\square\square\square: \square - \square = [\square - \square, \square - \square] \quad (15)$$

$$\square\square\square\square\square\square\square\square: \square \times \square = [\min(\square\square, \square\square, \square\square, \square\square), \max(\square\square, \square\square, \square\square, \square\square)] \quad (16)$$

Dividing (apart as divide by a 0 interval)

$$\frac{X}{F} = \square \times \left(\frac{1}{F}\right)_{\square\square\square h} \qquad \frac{1}{F} = \left[\frac{1}{d}, \frac{1}{c}\right] \quad (17)$$

In addition, there are a number of fundamental results about non decreasing mapping in relation to the mapping (function) of intervals.

$$\square(\square) = \square([\square, \square]) = [\square(\square), \square(\square)] \quad (18)$$

For nonincreasing function,

$$\square(\square) = \square([\square, \square]) = [\square(\square), \square(\square)] \quad (19)$$

**B. Architectures of Interval-Weight NNs**

Figure 3 represents the GNN that is going to be examined. It consists of single output and hidden layer neuron, with n1 neurons in each. The network's features, or inputs, are arranged in the vector form x = [x1, x2,. . ., xn]T. The weight, that joins the the jth hidden layer neuron with the ith input layer neuron is represented by the interval Wji, Wji =[w− ji,w+ji].

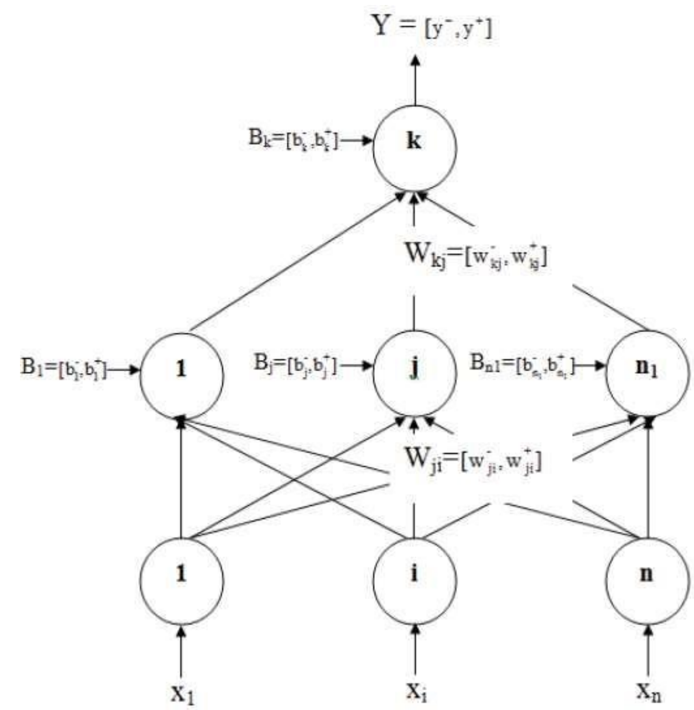


**Fig. 3. Architecture of a granular network.**

Interval value also applies to the weights that separate the output layer from the hidden layer. An interval bias is inherent in every neuron. Because the network's connectivity is interval-feature, any numerical input that is processed results in an interval, such as Y = [y−, y+]. Begin with the previously built MLP (e.g., expressed by the Levenberg–Marquardt BP (BackPropagation) learning technique) as described in the design. It is during this design step that the dimension of the hidden layer is chosen. f1 and f2, accordingly, stand for hidden layer and output layer neurons' activation functions. The following gives a description of the relevant formulas:

hidden layer,

$$h_j = f_1(z_j),\ z_j = \sum_{i=1}^{n} w_{ji}x_i + b_j, \qquad j = 1,2,\ldots, n_1 \tag{20}$$

$$f_1(z) = \frac{2}{1+e^{-2\times z}} - 1 \tag{21}$$

output layer

$$y = f_2(z),\ z = \sum_{j=1}^{n1} v_j h_j + b_j, \qquad f_2(z) = \tag{22}$$

The following hidden layer of the interval-valued NN applies the previously given formulas:

$$H_i = [h_j^-, h_j^+] = [f_1(z_j^-), f_1(z_j^+)] \tag{23}$$

$$z_j^- = \sum_{i=1}^{n} (w_{ji}^- x_i + b_j), \tag{24}$$

$$z_j^+ = \sum_{i=1}^{n} (w_{ji}^+ x_i + b_j), \qquad j = 1,2,\ldots\ldots, n1 \tag{25}$$

output layer

$$Y = [y^-, y^+] = [f_2(z^-), f_2(z^+)] \tag{26}$$

$$z^- = \sum_{i=1}^{n} (\min(v_j^- h_j^-, v_j^+ h_j^-, v_j^- h_j^+, v_j^+ h_j^+) + b_j) \tag{27}$$

$$z^+ = \sum_{j=1}^{n1} (\max(v_j^- h_j^-, v_j^+ h_j^-, v_j^- h_j^+, v_j^+ h_j^+) + b_j) \tag{28}$$

Since the objectives derived from the experimental results are numerical and the outputs of the GNN are intervals, by defining an appropriate performance index, or objective function, the optimization of which (minimization or maximization) is achieved by an appropriate distribution or allocation of the granularity of the data.

**Results and Discussion**

The suggested HDL-MCSA approach is assessed by contrasting its results with those of the current classifiers. In addition to classification accuracy, the classifier is evaluated using the average outcomes for each classifier and the statistical metrics provided in equations (29)–(32).

The ratio of correctly identified positive results to all expected positive information is termed as precision.

Precision = TP/TP+FP (29)

The ratio of correctly identified positive results to all data in the definite class is termed as sensitivity.

Recall = TP/TP+FN (30)

The weighted average of Precision and Recall is known as the F1 score. That requires both outcomes as FN (False Negatives), and FP (False Positives).

F1 Score = 2*(Recall * Precision) / (Recall +

Precision) (31)

The following formula determines accuracy by the positives and negatives:

$$\text{Accuracy} = (TP+FP)/(TP+TN+FP+FN) \quad (32)$$

here TP stands for True Positive, FP stands for False Positive, TN stands for True Negative, and FN stands for False Negative, respectively. Every standard and suggested classifier has these parameter values computed.

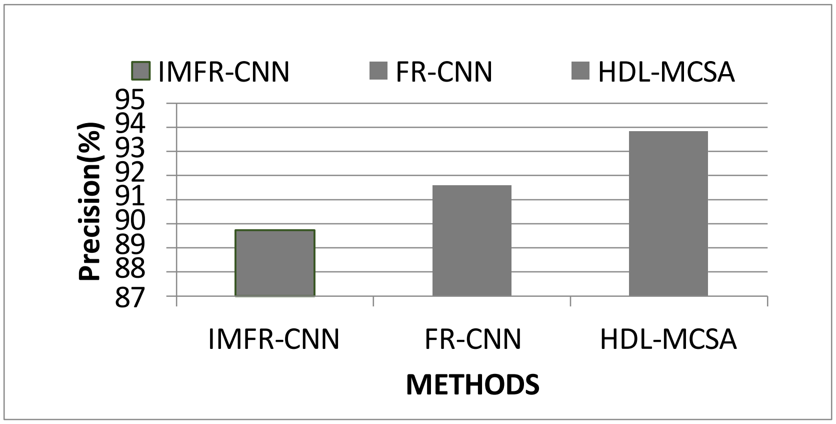


Figure .4. Precision comparison outcomes of the suggested HDL-MCSA with the existing classifiers

Figure 4 indicates the efficiency of the suggested HDL-MCSA's precision comparison results. As an outcome, studies indicate that FS with MCSA may predict pests accurately. Using a similar image dataset, the suggested HDL based classifier clearly obtained the highest performing precision score of 93.83% on crop pest classifications.

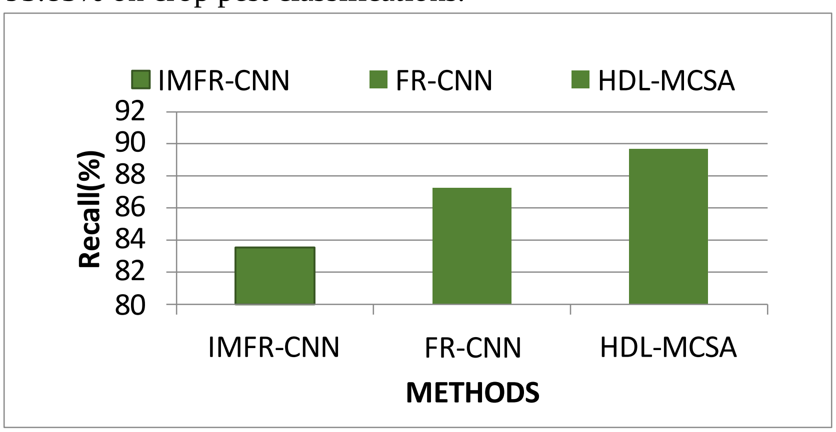


Figure 5. Recall comparison outcomes of the suggested HDL-MCSA with the previous classifiers

The execution outcomes of the suggested HDL-MCSA technique are presented in figure 5. According to the outcomes, the present strategy yields lower recall outcomes (87.25% for the FR-CNN strategy metric and 83.54% for the IMFR-CNN approach metric), while the suggested technique yields high recall rates (91.25%).

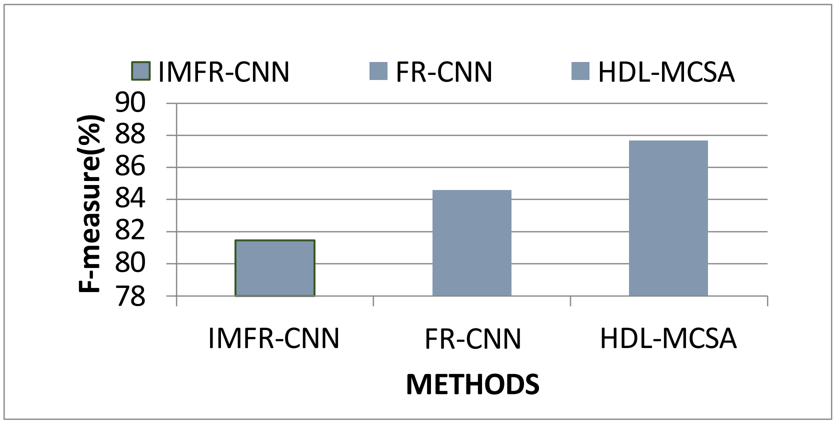


Figure .6. F-measure comparison outcomes of the suggested HDL-MCSA with the present classifiers

The outcomes presented in Figure 6 demonstrate that the suggested MCSA in conjunction using the HDL classifier yields an impressive pest prediction rate, significantly outperforming both FR-CNN and IMFR-CNN. The findings of the HDL research method and the quantitative analysis by F-measure concur. The accuracy of the suggested FR-CNN and GNN classifier is assessed with that of other cutting-edge classification techniques using a pest dataset.

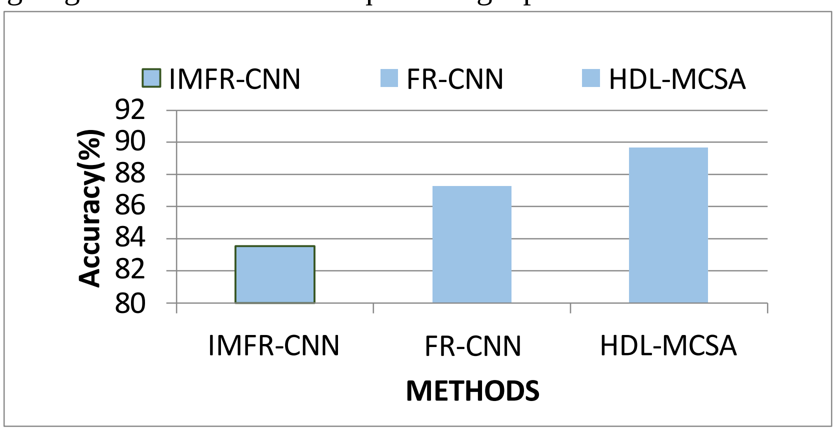


Figure .7. Accuracy comparison outcomes of the suggested HDL-MCSA with the present classifiers

The suggested HDL-MCSA classifiers provide greater accuracy than the current classifier, as demonstrated in Figure.7. However, HDL-MCSA techniques and other classifiers like FR-CNN and IMFR-CNN took into account the entire dataset at once. In comparison, the MABCO classifier outperforms all other classifiers when used in a static setting, demonstrating the efficacy of the approach for recognizing pests. Consequently, the classifiers' accuracy will be higher than that of the current classifier.

## 5. Conclusion

Contamination is the main factor responsible for both financial loss and agricultural loss worldwide. If invasive insects could be automatically recognized, the process of detecting and eliminating foreign insects would be significantly expedited. Utilizing the IP102 dataset in a CC environment, this research effort developed a strong FS and HDL algorithm for the efficient recognition of pests. The assessment outcomes of the suggested HDL-MCSA for insect pest recognition presented better performance than the SOTA techniques, GNNs and Faster R-CNN. The outcomes of the research proved that the apps of mobile they created for automated crop pest control is valid and efficient. In addition, the pest classification outcomes are combined with the application of appropriate pesticides to provide guidance to farmers and specialists. However, the research only looks into significant pests. Insects come in a wide variety of forms, including variations among larvae and adults. For instance, noctuid pests have only been discovered in their larval phase, which is the most hazardous stage of growth. Our goal is to increase the number of pest types that the suggested HDL-MCSA framework can properly identify in the future, hence increasing the classification accuracy. This study may speed up and improve the efficiency with which experts and farmers identify pests, minimizing damage to the economy and agricultural productivity.

## REFERENCES

1. Dalai, R., & Senapati, K. K. (2019, March). An intelligent vision based pest detection system using RCNN based deep learning mechanism. In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)* (pp. 1-4). IEEE.
2. Lima, M. C. F., de Almeida Leandro, M. E. D., Valero, C., Coronel, L. C. P., & Bazzo, C. O. G. (2020). Automatic detection and monitoring of insect pests—A review. *Agriculture*, *10*(5), 161.
3. Cardim Ferreira Lima, M., Damascena de Almeida Leandro, M. E., Valero, C., Pereira Coronel, L. C., & Gonçalves Bazzo, C. O. (2020). Automatic detection and monitoring of insect pests—A review. *Agriculture*, *10*(5), 161.
4. Segalla, A., Fiacco, G., Tramarin, L., Nardello, M., & Brunelli, D. (2020, November). Neural networks for pest detection in precision agriculture. In *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)* (pp. 7-12). IEEE.
5. Nugroho, A. P., Purba, S., Pratomo, Y. B., Hadi, S., & Utami, S. S. (2020, February). Development of cloud-based bioacoustics monitoring system for supporting Integrated Pest Management in agriculture production. In *IOP Conference Series: Earth and Environmental Science* (Vol. 449, No. 1, p. 012032). IOP Publishing.
6. Lina, Y., & Xiuming, Y. (2020, December). Design of Intelligent Pest Monitoring System Based on Image Classification Algorithm. In *2020 3rd International Conference on Control and Robots (ICCR)* (pp. 21-24). IEEE.
7. Li, W., Zheng, T., Yang, Z., Li, M., Sun, C., & Yang, X. (2021). Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. *Ecological Informatics*, *66*, 101460.
8. Sharmila, V. C., Chauhan, N., Kumar, R., & Barwal, S. (2021, June). Design of Intelligent Insect Monitoring System Using Deep Learning Techniques. In *Proceedings of the First*

*International Conference on Computing, Communication and Control System, I3CAC 2021, 7-8 June 2021, Bharath University, Chennai, India*.

9. Malek, M. A., Reya, S. S., Hasan, M. Z., & Hossain, S. (2021, January). A crop pest classification model using deep learning techniques. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (pp. 367-371). IEEE.
10. Karar, M. E., Alsunaydi, F., Albusaymi, S., & Alotaibi, S. (2021). A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*, *60*(5), 4423-4432.
11. Gambhir, J., Patel, N., Patil, S., Takale, P., Chougule, A., Prabhakar, C. S., ... & Sohane, R. K. (2022, February). Deep Learning for Real-Time Diagnosis of Pest and Diseases on Crops. In *Intelligent Data Engineering and Analytics: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)* (pp. 189-197). Singapore: Springer Nature Singapore.
12. Devisurya, V., Vidhyaa, V. P., Shobiya, K., & Suguna, C (2022). Insect and Pest Detection in Stored Grains: Analysis of Environmental Factors and Comparison of Deep Learning Methods.
13. Kathole, A. B., Katti, J., Lonare, S., & Dharmale, G. (2023). Identify and classify pests in the agricultural sector using metaheuristics deep learning approach. *Franklin Open*, 100024.
14. Popescu, D., Dinca, A., Ichim, L., & Angelescu, N. (2023). New trends in detection of harmful insects and pests in modern agriculture using artificial neural networks. a review. *Frontiers in Plant Science*, *14*, 1268167.
15. Peng, H., Xu, H., Gao, Z., Zhou, Z., Tian, X., Deng, Q., ... & Xian, C. (2023). Crop pest image classification based on improved densely connected convolutional network. *Frontiers in Plant Science*, *14*, 1133060.
16. Yang, Z., Yang, X., Li, M., & Li, W. (2021). Automated garden-insect recognition using improved lightweight convolution network. *Information Processing in Agriculture*.
17. FAISAL, M. S. A. B. (2021). A Pest Monitoring System for Agriculture Using Deep Learning. *Research Progress in Mechanical and Manufacturing Engineering*, *2*(2), 1023-1034.
18. Chen, J. W., Lin, W. J., Cheng, H. J., Hung, C. L., Lin, C. Y., & Chen, S. P. (2021). A smartphone-based application for scale pest detection using multiple-object detection methods. *Electronics*, *10*(4), 372.
19. Albanese, A., Nardello, M., & Brunelli, D. (2021). Automated pest detection with DNN on the edge for precision agriculture. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *11*(3), 458-467.
20. Bhoi, S. K., Jena, K. K., Panda, S. K., Long, H. V., Kumar, R., Subbulakshmi, P., & Jebreen, H. B. (2021). An Internet of Things assisted Unmanned Aerial Vehicle based artificial intelligence model for rice pest detection. *Microprocessors and Microsystems*, *80*, 103607.

21. Rodríguez-García, M. Á., García-Sánchez, F., & Valencia-García, R. (2021). Knowledge-based system for crop pests and diseases recognition. *Electronics*, *10*(8), 905.
22. Tiwari, V., Patel, H., Muttreja, R., Goyal, M., Ojha, M., Gupta, S., ... & Jain, S. (2021). Real-time soybean crop insect classification using customized deep learning models. In *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2021, Volume 1* (pp. 143-156). Springer Singapore.
23. Karar, M. E., Alsunaydi, F., Albusaymi, S., & Alotaibi, S. (2021). A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*, *60*(5), 4423-4432.
24. Deepika, P., & Arthi, B. (2022). Prediction of plant pest detection using improved mask FRCNN in cloud environment. *Measurement: Sensors*, *24*, 100549.
25. Vamsi, B. T., Yadla, J., & Kumar, G. V. (2023, May). Application of IP102 Dataset in Pest Detection by using Deep Learning Techniques. In *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
26. Venkatasaichandrakanth, P., & Iyapparaja, M. (2024). A Detailed Study on Deep Learning versus Machine Learning Approaches for Pest Classification in Field Crops. In *Artificial Intelligence and Machine Learning for Smart Community* (pp. 1-25). CRC Press.
27. Yi, X. U. (2020). Aphids and their transmitted potato viruses: A continuous challenges in potato crops. *Journal of Integrative Agriculture*, *19*(2), 367-375.
28. Lundin, O. (2019). No-till protects spring oilseed rape (Brassica napus L.) against crop damage by flea beetles (Phyllotreta spp.). *Agriculture, Ecosystems & Environment*, *278*, 1-5.
29. Seetha, S., Sahu, S., Mahapatra, B. B., & Mishra, M. (2017). Post Embryonic Changes in the Eye of an Economic Mango Plant Pest Amritodus atkinsoni Leth.(Hemiptera: Membracoidea: Cicadellidae). In *3rd International Multidisciplinary Microscopy and Microanalysis Congress (InterM) Proceedings, Oludeniz, Turkey, 19-23 October 2015* (pp. 25-31). Springer International Publishing.
30. Piccinini, E., Ferrari, V., Campanelli, G., Fusari, F., Righetti, L., Matteo, R., & Lazzeri, L. (2015). Effect of two bio-based liquid formulations from Brassica carinata in containing red spider mite (Tetranychus urticae) on eggplant. *Industrial Crops and Products*, *75*, 36-41.
31. Li, Z., Jiang, X., Jia, X., Duan, X., Wang, Y. and Mu, J., 2022. Classification Method of Significant Rice Pests Based on Deep Learning. *Agronomy*, *12*(9), p.2096.
32. Wang, Y., Fang, G., Cao, Y., Wu, N., Cui, Q., Zhu, C., Qian, L., Huang, Y. and Zhan, S., 2021. The genome of the black cutworm Agrotis ipsilon. *Insect Biochemistry and Molecular Biology*, *139*, p.103665.
33. Gomes, J.C. and Borges, D.L., 2022. Insect pest image recognition: A few-shot machine learning approach including maturity stages classification. *Agronomy*, *12*(8), p.1733.

34. Fei, N., Gao, Y., Lu, Z., & Xiang, T. (2021). Z-score normalization, hubness, and few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 142-151).
35. Yang, Q., Huang, H., Zhang, J., Gao, H., & Liu, P. (2023). A collaborative cuckoo search algorithm with modified operation mode. *Engineering Applications of Artificial Intelligence*, *121*, 106006.
36. Tomasiello, S., Loia, V., & Khaliq, A. (2021). A granular recurrent neural network for multiple time series prediction. *Neural Computing and Applications*, *33*(16), 10293-10310.