

Article

Improved Artificial Ecosystem Optimizer with Deep-Learning-Based Insect Detection and Classification for Agricultural Sector

Mohammed Aljebreen ¹, Hanan Abdullah Mengash ² , Fadoua Kouki ³ and Abdelwahed Motwakel ^{4,*} 

¹ Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia

² Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ Department of Financial and Banking Sciences, Applied College at Muhail Aseer, King Khalid University, Abha 62529, Saudi Arabia

⁴ Department of Information Systems, College of Business Administration in Hawtat Bani Tamim, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia

* Correspondence: am.ismaeil@psau.edu.sa

Abstract: The agricultural industry has the potential to meet the increasing food production requirements and supply nutritious and healthy food products. Since the Internet of Things (IoT) phenomenon has achieved considerable growth in recent years, IoT-based systems have been established for pest detection so as to mitigate the loss of crops and reduce serious damage by employing pesticides. In the event of pest attack, the detection of crop insects is a tedious process for farmers since a considerable proportion of crop yield is affected and the quality of pest detection is diminished. Based on morphological features, conventional insect detection is an option, although the process has a disadvantage, i.e., it necessitates highly trained taxonomists to accurately recognize the insects. In recent times, automated detection of insect categories has become a complex problem and has gained considerable interest since it is mainly carried out by agriculture specialists. Advanced technologies in deep learning (DL) and machine learning (ML) domains have effectively reached optimum performance in terms of pest detection and classification. Therefore, the current research article focuses on the design of the improved artificial-ecosystem-based optimizer with deep-learning-based insect detection and classification (IAEODL-IDC) technique in IoT-based agricultural sector. The purpose of the proposed IAEODL-IDC technique lies in the effectual identification and classification of different types of insects. In order to accomplish this objective, IoT-based sensors are used to capture the images from the agricultural environment. In addition to this, the proposed IAEODL-IDC method applies the median filtering (MF)-based noise removal process. The IAEODL-IDC technique uses the MobileNetv2 approach as well as for feature vector generation. The IAEODL-IDC system is utilized for optimal hyperparameter tuning of the MobileNetv2 approach. Furthermore, the gated recurrent unit (GRU) methodology is exploited for effective recognition and classification of insects. An extensive range of simulations were conducted to exhibit the improved performance of the proposed IAEODL-IDC methodology. The simulation results validated the remarkable results of the IAEODL-IDC algorithm with recent systems.

Keywords: Internet of Things; agriculture; insect recognition; crop productivity; computer vision; deep learning; parameter tuning



Citation: Aljebreen, M.; Mengash, H.A.; Kouki, F.; Motwakel, A. Improved Artificial Ecosystem Optimizer with Deep-Learning-Based Insect Detection and Classification for Agricultural Sector. *Sustainability* **2023**, *15*, 14770. <https://doi.org/10.3390/su152014770>

Academic Editors: Bhaskar B. Gardas, Nima Jafari Navimipour, Vaibhav S. Narwane and Nilesh P. Ghongade

Received: 26 July 2023

Revised: 20 August 2023

Accepted: 20 September 2023

Published: 11 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the application of Internet of Things (IoT) technology in agricultural practices has proved to have great potential in terms of developing dynamic pest management practices. One of the major effective applications is the utilization of IoT devices to classify and inspect pests in agricultural settings. With the help of a network of sensors that capture environmental images and data, farmers gain clearer insights into the existence of pests

and their activities [1]. Such sensors gather vital data, namely soil moisture, temperature, humidity, and the images of crops along with the surrounding vegetation. With real-time data analytics and processing techniques, the system rapidly identifies the pest outbreaks and immediately alerts the agricultural experts or farmers, thus allowing them to proceed with timely and targeted pest management measures [2]. This combination of the IoT and pest classification techniques has the potential to revolutionize pest control in agriculture, causing more sustained practices, decreased pesticide use, and, ultimately, maximum crop yields.

Insect detection in IoT-based agricultural practice is vital to advancing sustainability by enabling accurate and targeted pest management practices. By leveraging the IoT technologies, farmers can observe the insect populations on a real-time basis, which in turn enables them to reduce the indiscriminate use of pesticides, optimize resources, and mitigate environmental impact. This type of data-driven approach not only preserves essential resources but also promotes various other features such as biodiversity preservation, crop health, and resilience in the face of climate change, aligning agricultural practices with the overarching goal of sustainable and environmentally responsible food production. Early identification of pest infestations enables timely intervention and also prevents massive crop damage. Further, this preemptive process helps in maintaining the crop yield and also minimizing the risks of crop failure, thus contributing to food security and sustainable agricultural production. Sustainable agriculture often involves conservation and improvement of biodiversity. IoT-based insect detection techniques help farmers to differentiate the harmful pests from beneficial insects, such as pollinators and natural predators. It is vital to protect these beneficial species in order to maintain a balanced ecosystem within agricultural environments, which in turn supports long-term sustainability.

Pest attack is one of the major problems encountered by the agricultural sector, and it degrades the quality of crops and accordingly the yield [3]. Weeds, pests, and germs cause heavy losses to the crops and negatively impact the market for final goods. So, it is of immense importance to identify innovative ways to obtain even small growths in efficacy [4]. Extra caution should be taken regarding pest attacks on crops that greatly affect the quality and yield of the crop. The high demand for cash crops mainly contributes to the large scale of production. The key reason behind the degradation of crop quality is insects, which also diminish the crop yield [5]. On the other hand, it is essential to evaluate and monitor the losses incurred due to insects so as to overcome the safety and crop quality issues in agriculture. Manual identification and classification of insects necessitate professional knowledge of the field and are also time-consuming processes [6]. Conventionally, expert entomologists classify insects through manual procedures. However, it is challenging for non-professionals to categorize insects without entomological knowledge. Thus, the study of automatic detection of insect types using image processing technology is extremely useful [7].

Computer vision techniques have been successfully adopted for monitoring soil and crop environments, insect pest recognition, plant disease detection, and fruit grading. Various developments have been achieved recently in agriculture with the involvement of machine learning (ML) techniques to classify and detect insects in stored grain settings [8]. Yet, the prevailing ML methods have some limitations if applied in the image detection process due to the compromised performance of the hand-engineered features that affect the whole results. Further, learning such features is a complicated and time-consuming task considering that the system has to be changed when the dataset changes or new problems arise. Hence, these features demand expensive efforts that hinge on professionals and are ineffective [9]. Many researchers have established that deep learning (DL) techniques possess substantial benefits in terms of feature extraction from the images [10].

In this context, the current research article concentrates on the design of an improved artificial-ecosystem-based optimizer with deep-learning-based insect detection and classification (IAEODL-IDC) approach in the IoT-based agricultural sector. The IoT sensors are primarily used for data collection purposes. The presented IAEODL-IDC technique uses the

median filtering (MF)-based noise removal process. In addition, the IAEODL-IDC method makes use of the MobileNetv2 system for feature vector generation. Meanwhile, the IAEODL methodology is also exploited for optimal hyperparameter tuning of the MobileNetv2 system. Lastly, the gated recurrent unit (GRU) methodology is exploited for an effective recognition and classification of insects. An extensive range of simulations were conducted to exhibit the superior performance of the IAEODL-IDC algorithm. The key contributions of the current study are summarized as follows:

- Development of the automated IAEODL-IDC technique comprising MF-based preprocessing, a MobileNetv2 feature extractor, IAEODL-based parameter tuning, and GRU-based classification for insect detection in the IoT-based agricultural sector. To the best of the authors' knowledge, there are no existing reports on the IAEODL-IDC technique in the literature.
- Design of a new IAEODL technique by integrating the lens imaging dynamic learning approach and the AEO algorithm to overcome the local optima problem.
- Hyperparameter optimization of the GRU model using the IAEODL algorithm with cross-validation, which helps in boosting the predictive outcome of the IAEODL-IDC model for unseen data.

2. Related Works

Kundu et al. [11] developed a process named the 'Automated and Intelligent Data Collector and Classifier' structure by incorporating the IoT and DL techniques. This 'Custom-Net' technique was utilized on the cloud server. The Grad-CAM was also applied for visualizing the attributes derived by 'Custom-Net'. Additionally, the effect of the TL on Inception ResNetV2, 'Custom-Net', InceptionV3, VGG19, VGG16, and ResNet50 is shown in this study. Kusrini et al. [12] presented an advanced ML approach to detect the onset of biological threats with the help of CV and DL technologies and examine large-scale mango fields. The ML method extended the pre-trained VGG16 DL approach in order to supplement the final layer with FC network training. In Reference [13], a pest detection technique was modeled using the improved YOLOv5-based method with high precision. Firstly, the Transformer (C3TR) and SWin Transformer (SWinTR) systems were incorporated into the YOLOv5 network in such a way that a larger number of global features are captured, which can increase the receptive sector.

In a previous study [14], an insect pest detection technique was presented with contour detection and foreground extraction in order to identify insects. To enhance the outcome of the classification method, a 9-fold cross-validation method was implemented. The highest classifier rates of 90% and 91.5% were attained for 9 and 24 class insects through the convolutional neural network (CNN) method. Li and Chao [15] devised an artificial neural network (ANN)-based continual classification technique through retrieval and memory storage with two advantages, namely high flexibility and few data. The presented ANN-based method integrated both a CNN and a generative adversarial network (GAN). Islam et al. [16] examined the performances of several ML approaches such as the random forest (RF), K-nearest neighbor (KNN), and a support vector machine (SVM) to detect weeds through UAV images captured in chili crop fields. Rong et al. [17] presented an object detection technique based on the enhanced Mask RSCNN method that is targeted at enhancing the efficiency and accuracy in the pest detection process. In Reference [18], the authors presented non-destructive approaches to find the syndromes that affect tomato crops such as the target spot (TS), bacterial spot (BS) and tomato yellow leaf curl (TYLC) for two tomato types utilizing hyperspectral sensing under two conditions: (a) in the field using a drone-based application and (b) in the lab (benchtop scanning).

3. The Proposed Model

In the current study, an automated insect recognition technique, i.e., the IAEODL-IDC technique, has been developed to properly classify various types of insects. The presented IAEODL-IDC technique comprises MF-based noise removal, MobileNetv2 feature extrac-

tion, IAEO-based hyperparameter optimization, and GRU-based classification. Figure 1 exhibits the workflow of the IAEODL-IDC approach.

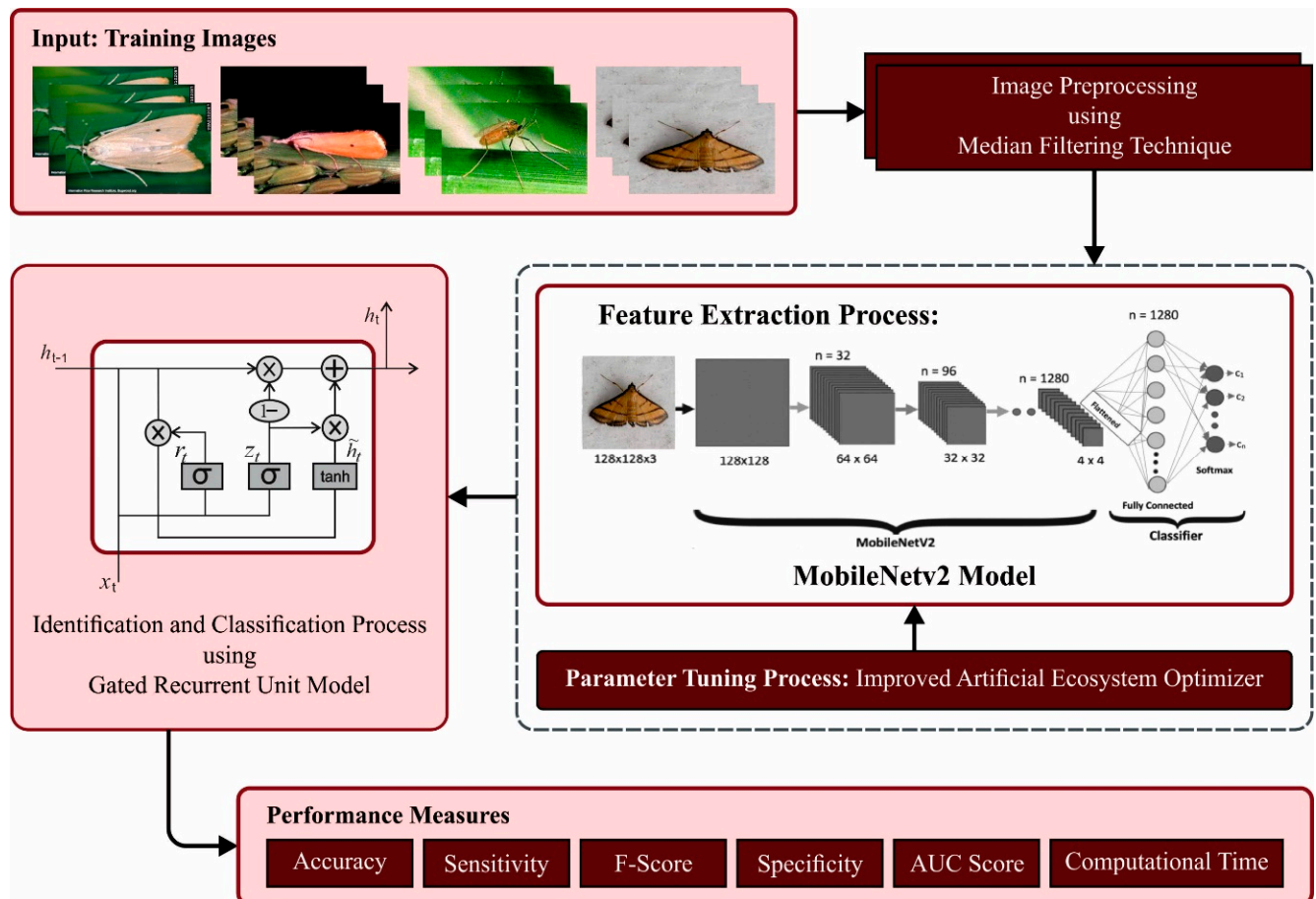


Figure 1. Workflow of the IAEODL-IDC approach.

3.1. Image Pre-processing

Initially, the MF system is employed to eradicate the noise present in the images. The MF system is a digital signal processing method that is used for the removal of noise from signals or images [19]. This system works by replacing all the pixels in an image with the median values of the nearby pixel in a predetermined neighborhood. MF is a non-linear filter, i.e., the output of the filter relies on the order of input.

3.2. Feature Extraction

In order to develop the feature vectors, the MobileNetV2 model is utilized in this study. A CNN is an NN with more than one layer, and it involves multiple convolutional pooling layer pairs along with full-connection output layers [20]. The aim of a typical CNN is to recognize that an image's shape remains partially invariant to the position of the shape. In the convolution layer, the input image is convolved with the help of the 2D filter. For instance, when a 2D image I is fed as an input and a 2D convolutional kernel K is applied, then

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n K(i - m, j - n) \quad (1)$$

Next, the feature map following the convolution filter is downsized to small instances from the pooling layer. Both the network filters (kernel) and the weights in the convolution layer are learnt via the backpropagation model in order to mitigate the classification error. The network is exploited to train $N_F = 30$ filters in total, sized at $N_h \times 3$. The input image

is convolved by the kernel that is to be trained, and later it is passed over the f output function to generate the output map in the convolutional layer. The map of the k -th feature is attained using the following equation:

$$h_{ij}^k = f(a) = f((W^k * x)_{ij} + b_k) \quad (2)$$

In Equation (2), W^k signifies the weight matrix for the k th filter, and b_k characterizes the bias values; for $k = 1, 2, \dots, N_F$ and x signify the input image. In this work, the 1D filter has $N_h \times 3$ dimensions, $j = 1$ and $= 1, 2, \dots, N_t - 2$. The f output function is designated as ReLU.

$$f(a) = \text{ReLU}(a) = \ln(1 + e^a) \quad (3)$$

The result of the convolution layer is an N_F vector sized at $(N_t - 2) \times 1$. The max-pooling layer is a full-connection layer with two outcomes that characterize the MI of L.H.S and R.H.S [21]. Here, the BP model is utilized for learning the CNN parameters. Using the presented method, the network is provided with the labeled trained set, and the error E is calculated by assuming that the selected output is dissimilar to the actual output. Then, the gradient descent model is exploited to minimize the error E that takes place with changes in the mode parameter and is represented as follows.

$$W^k = W^k - \eta \frac{\partial E}{\partial W^k} \quad (4)$$

$$b_k = b_k - \eta \frac{\partial E}{\partial b_k} \quad (5)$$

where W^k signifies the weight matrix for kernel k , η refers to the learning rate, and b_k characterizes the bias value. Eventually, the trained network is used to categorize the new sample in the testing set.

3.3. Hyperparameter Tuning

The IAEO algorithm is exploited for the optimal hyperparameter tuning process. AEO is a metaheuristic optimization technique that is modeled on the flow of energy in the Earth's ecosystem [22], such as production, consumption, and the decomposition processes. In such a system, food offers energy together with sunlight, CO_2 , nutrients, and water generated by decomposers. The generation operator supports the AEO in arbitrarily replacing the preceding one (x_n) with a novel individual (x_{rand}). The production behavior is determined as given below.

$$\chi_1(t+1) = (1-a)x_{NP}(t) + ax_{rand}(t) \quad (6)$$

$$a = \left(1 - \frac{t}{T}\right)r_1 \quad (7)$$

$$x_{rand} = r(U - L) + L \quad (8)$$

Here, NP denotes the size of the population, T implies the maximal iteration count of this technique, and L and U denote the upper and lower boundaries of the searching space, respectively.

During the AEO, random walking, at a point dependent upon Levy flight (LF), is termed as the consumption feature and is calculated as follows.

$$C = \frac{1}{2} \frac{v_1}{|v_2|} \quad (9)$$

$$v_1 \sim N(0,1), \quad v_2 \sim N(0,1) \quad (10)$$

Herbivore: The consumption design of the herbivore is as follows

$$x_i(t+1) = x_i(t) + C.(x_i(t) - x_1(t)), i \in [2, \dots, n] \quad (11)$$

Carnivore: The consumption design of the carnivore is expressed using Equation (12):

$$\begin{cases} x_i(t+1) = x_i(t) + C.(x_i(t) - x_1(t)), & i \in [2, \dots, NP] \\ j = randi([2 \ i - 1]) \end{cases} \quad (12)$$

Omnivore: When a consumer assumes another consumer, it could either eat the consumer, which has high energy, or the producer. This performance is determined as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + C.(x_i(t) - \chi_{1(t)}) + (1 - r_2)(x_i(t) - x_j(t)), & i = 3, \dots, NP \\ j = randi([2 \ i - 1]) \end{cases} \quad (13)$$

r_2 implies the random number from the interval. During the decomposition procedure, the decomposer decomposes the situation, when the individual dies. The decomposition procedures are determined as follows.

$$\chi_i(t+1) = x_n(t) + D.(e.x_n(t) - h.x_i(t)), i = 1, \dots, NP \quad (14)$$

$$D = 3u, \quad u \sim N(0,1) \quad (15)$$

$$e = r_3.randi([1 \ 2]) - 1 \quad (16)$$

$$h = 2.r_3 - 1 \quad (17)$$

The lens imaging dynamic learning system can be assumed to prevent the AEO technique from becoming trapped in the local optima. Hence, on the L.H.S of the y -axis, the person is noticeable with F , whereas its estimate on the x -axis is exposed with X , and distance in the x -axis is noticeable with ζ . If accepted by a convex lens, the F proceeds with an inverse F' where the image on the x -axis is referred to as X' , and its distance in the x -axis is determined as ζ' . The person X and the opposite person X' are chosen.

X' is attained as follows.

$$X' = \frac{\psi_u + \psi_l}{2} + \frac{\psi_u + \psi_l}{2 \times \mu} - \frac{X}{\mu} \quad (18)$$

The scaling, dependent upon non-linear dynamic reduction μ , is determined as follows.

$$\mu = \lambda^{\min} - (\lambda^{\max} - \lambda^{\min}) \times \left(\frac{t}{T}\right)^2 \quad (19)$$

where $\lambda^{\max}$ and $\lambda^{\min}$ signify the upper and lower scaling features (100 and 10), respectively. Equation (18) is determined for n -dimensional space as given below.

$$X'_j = \frac{\psi_{uj} + \psi_{lj}}{2} + \frac{\psi_{uj} + \psi_{lj}}{2 \times \alpha} - \frac{X_j}{\alpha} \quad (20)$$

Here, X_j and X'_j denote the X_0 and X mechanisms in the dimension j , and ψ_{lj} and ψ_{uj} signify the upper and lower limits of the dimension j , respectively.

The IAEO technique not only develops a fitness function (FF) from attaining a supreme classifier solution, but it also expresses a positive integer to exemplify the high efficacy of candidate performances. A reduced classifier rate of errors is regarded as an FF and is expressed as follows.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{no. \text{ of misclassified instances}}{Total \text{ no. of instances}} * 100 \quad (21)$$

3.4. Image Classification

Lastly, the GRU technique subjects the image to class labeling. The GRU is composed of several GRU cells, and the number of hidden states is fixed at 2 [23]. Here, the update gate is used for controlling the extent to which the data of the preceding moment are taken into the existing state. The reset gate is the same gate as the forget gate in LSTM, which controls the number of data that are neglected from the preceding moment.

In each GRU cell, the forward propagation formula is given as follows:

$$\begin{aligned} z_t &= \sigma(a_t U^z + h_{t-1} W^z + b^z) \\ r_t &= \sigma(a_t U^r + h_{t-1} W^r + b^r) \\ h_t &= \tanh(a_t U^h + (h_{t-1} \circ r_t) W^h + b^h) \\ h_t &= (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \end{aligned} \quad (22)$$

In Equation (22), h_t , z_t , r_t , and $\tilde{h}_t$ refer to the active layers of the present hidden node outcome, the update gate, reset gate, and candidate layer of the existing hidden node at time t , respectively; $a_t \in R^d$ denotes the input vector to all the GRU cells; $\circ$ denotes the component-wise multiplication; U and W imply the weight matrices learned during model training; σ denotes the sigmoid activation function; b shows the bias vector; and $\tanh$ indicates the hyperbolic tangent function.

4. Results and Discussion

The proposed model was simulated using Python 3.6.5 on a PC configured with an i5-8600k CPU, a GeForce 1050Ti 4 GB GPU, 16 GB RAM, a 250 GB SSD, and a 1 TB HDD. The insect classification performance of the IAEO-IDL algorithm was validated using the IP102 database [24], containing 300 instances for six classes, as represented in Table 1. The IP102 database has a hierarchical taxonomy, and the insect pests that mainly affect one specific agricultural product are grouped in the same upper-level category. Figure 2 shows some of the sample images.

Table 1. Detailed database.

Insect Class	Label	Number of Insects
Rice leaf roller	Insect-1	50
Asiatic rice borer	Insect-2	50
Yellow rice borer	Insect-3	50
Rice gall midge	Insect-4	50
Rice Stemfly	Insect-5	50
Rice leafhopper	Insect-6	50
Total Number of Insects		300

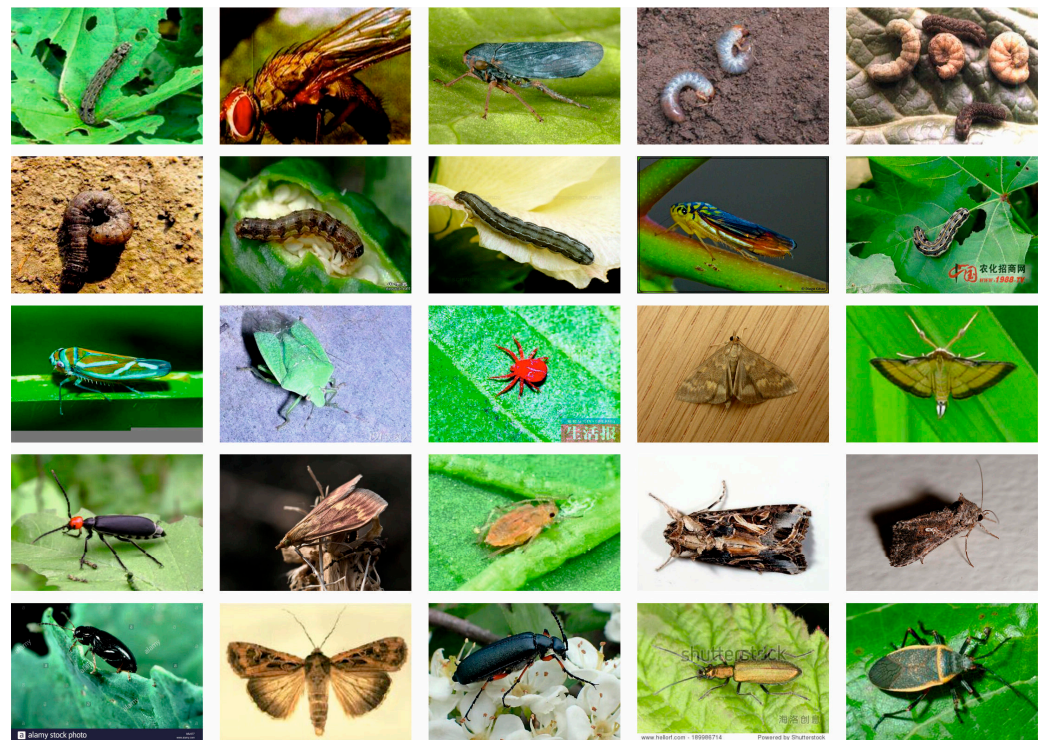


Figure 2. Sample images.

Figure 3 exhibits the classification results achieved by the IAEODL-IDC approach with the 70:30 TR/TS sets. Figure 3a,b show the confusion matrices generated by the ACC-CBOEFF algorithm with the 70:30 TR/TS sets. The outcomes denote that the ACC-CBOEFF method identified and classified all six class labels accurately. Also, Figure 3c demonstrates the PR investigation outcomes of the IAEODL-IDC algorithm with the 70:30 TR/TS sets. The outcome signifies that the IAEODL-IDC algorithm attained enhanced PR solutions in all the classes. On the other hand, Figure 3d exhibits the ROC investigation results achieved by the IAEODL-IDC model with the 70:30 TR/TS sets. The outcomes demonstrate that the IAEODL-IDC system achieved excellent results with maximal ROC values for all six class labels.

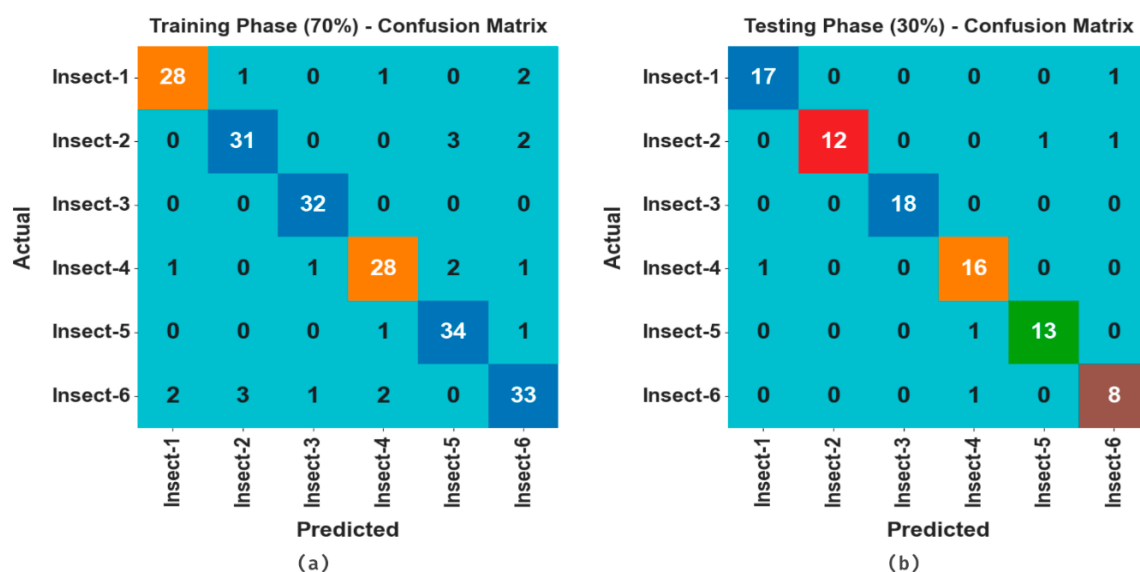


Figure 3. Cont.

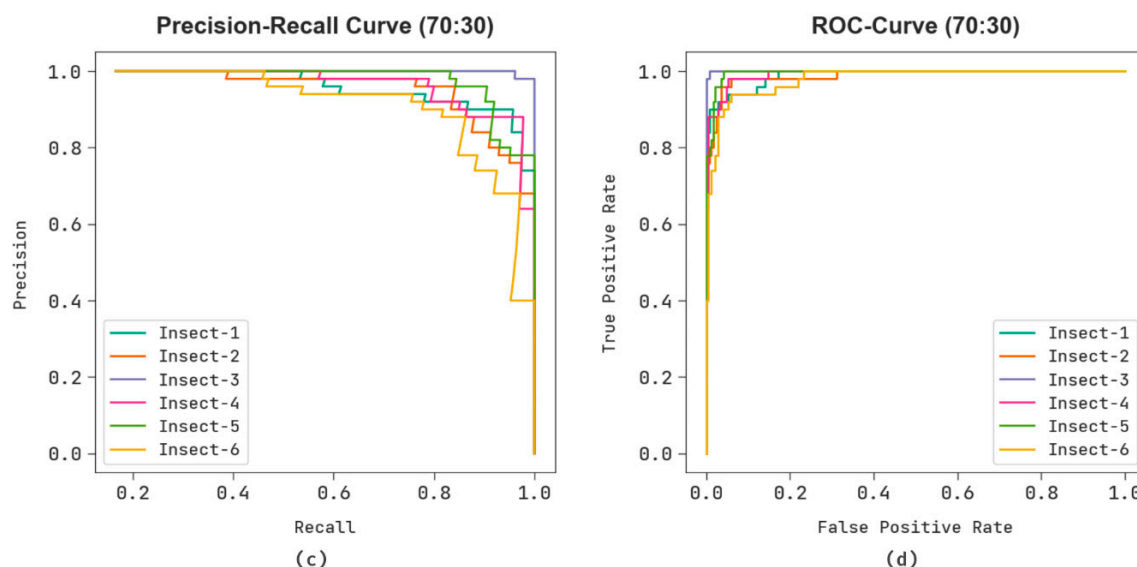


Figure 3. Classifier results for the 70:30 TR/TS sets: (a,b) confusion matrices, (c) PR curve, and (d) ROC curve.

Figure 4 exhibits the classification performance of the IAEODL-IDC system for the 70% TR set. The results show that the IAEODL-IDC technique determined all six types of insects. For the insect-1 class, the IAEODL-IDC technique obtained an $accu_y$ of 96.67%, $sens_y$ of 87.50%, $spec_y$ of 98.31%, F_{score} of 88.89%, and an AUC_{score} of 92.91%. Meanwhile, for the insect-3 class, the IAEODL-IDC system achieved an $accu_y$ of 99.05%, $sens_y$ of 100%, $spec_y$ of 98.88%, F_{score} of 96.97%, and an AUC_{score} of 99.44%. Furthermore, for the insect-6 class, the IAEODL-IDC technique reached an $accu_y$ of 93.33%, $sens_y$ of 80.49%, $spec_y$ of 96.45%, F_{score} of 82.50%, and an AUC_{score} of 88.47%.

Figure 5 exhibits the classification outcomes of the IAEODL-IDC approach for the 30% TS set. The outcome depicts that the IAEODL-IDC system determined all six types of insects. For the insect-1 class, the IAEODL-IDC technique obtained an $accu_y$ of 97.78%, $sens_y$ of 94.44%, $spec_y$ of 98.61%, F_{score} of 94.44%, and an AUC_{score} of 96.53%. Meanwhile, for the insect-4 class, the IAEODL-IDC system achieved an $accu_y$ of 96.67%, $sens_y$ of 94.12%, $spec_y$ of 97.26%, F_{score} of 91.43%, and an AUC_{score} of 95.69%. Additionally, in the insect-6 class, the IAEODL-IDC algorithm achieved an $accu_y$ of 96.67%, $sens_y$ of 88.89%, $spec_y$ of 97.53%, F_{score} of 84.21%, and an AUC_{score} of 93.21%.

Figure 6 demonstrates the classification outcomes of the IAEODL-IDC methodology for the 60:40 TR/TS sets. Figure 6a,b depict the confusion matrices generated by the ACC-CBOEFF algorithm for the 60:40 TR/TS sets. The outcomes show that the ACC-CBOEFF method recognized and classified all six class labels accurately. Next, Figure 6c displays the PR curve of the IAEODL-IDC method for the 60:40 TR/TS sets. The results state that the IAEODL-IDC method obtained the highest PR curve in all six classes. Lastly, Figure 6d shows the ROC investigation outcomes of the IAEODL-IDC system for the 60:40 TR/TS sets. The figure exhibits that the IAEODL-IDC method achieved excellent outcomes with the highest ROC values for dissimilar class labels.

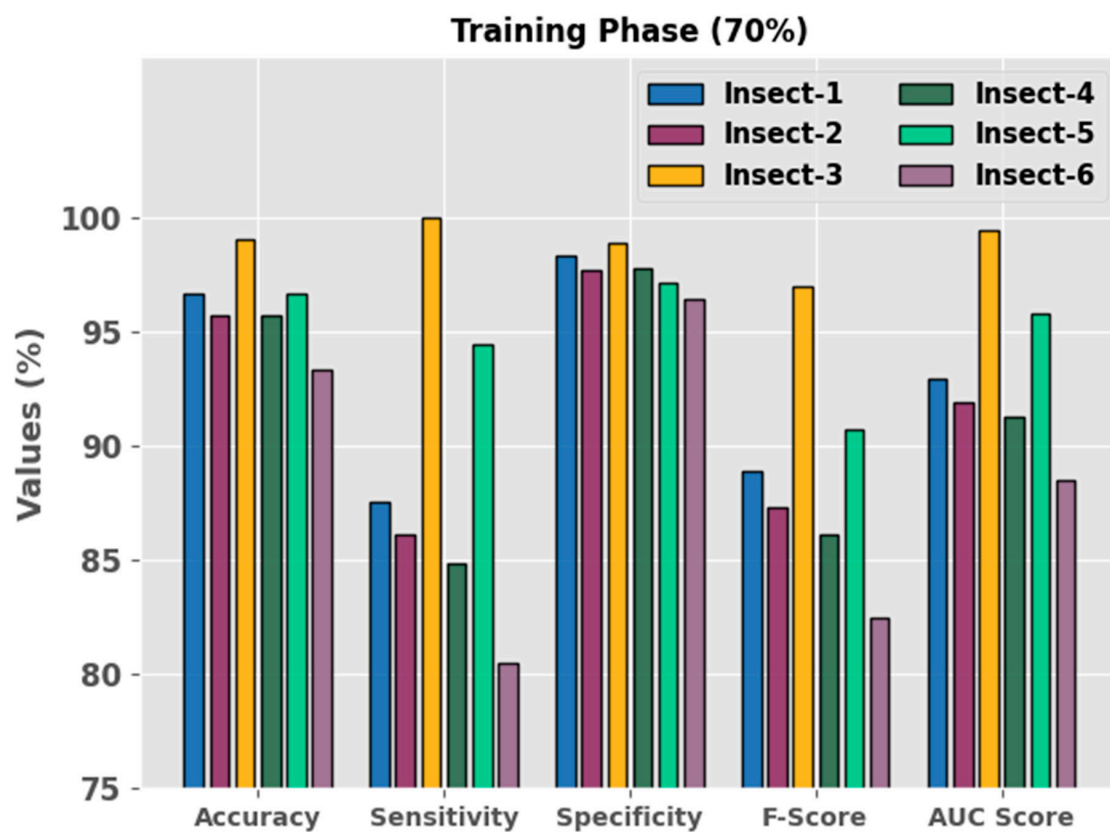


Figure 4. Insect classifier outcome of IAEODL-IDC algorithm on 70% of TR set.

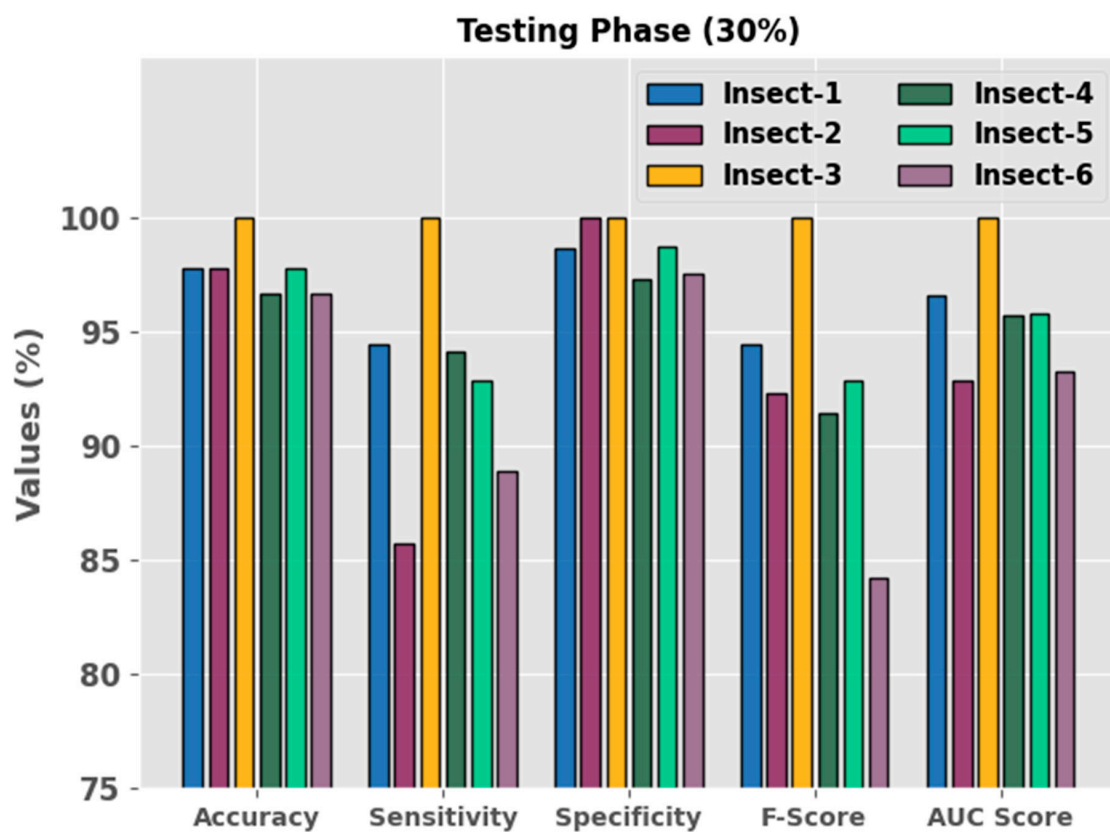


Figure 5. Insect classification outcomes of the IAEODL-IDC system for the 30% TS set.

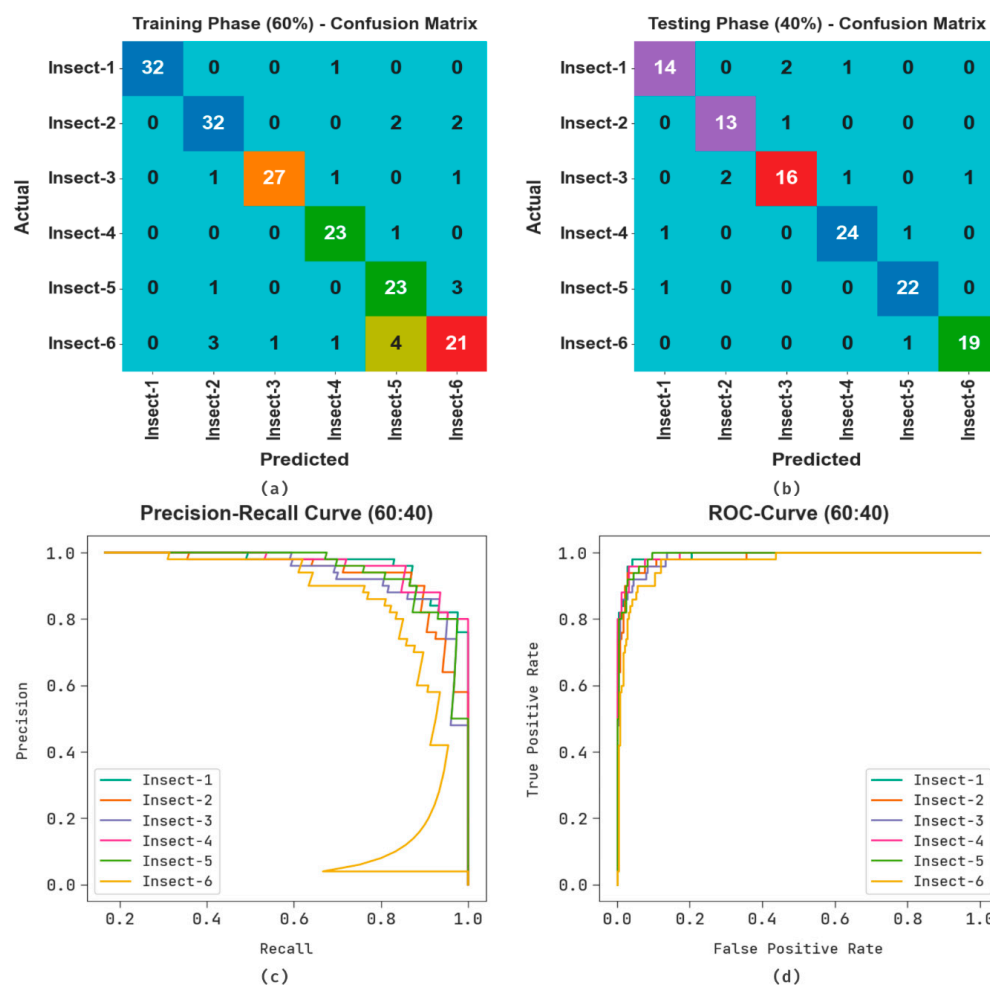


Figure 6. Classification results with the 70:30 TR/TS sets: (a,b) confusion matrices, (c) PR curve, and (d) ROC curve.

Figure 7 exhibits the classification outcomes of the IAEODL-IDC method for the 60% TR set. The results show that the IAEODL-IDC method determined all six types of insects. In the insect-1 class, the IAEODL-IDC method attained an $accu_y$ of 99.44%, $sens_y$ of 96.97%, $spec_y$ of 100%, F_{score} of 98.46%, and an AUC_{score} of 98.48%. Meanwhile, for the insect-3 class, the IAEODL-IDC technique achieved an $accu_y$ of 97.78%, $sens_y$ of 90%, $spec_y$ of 99.33%, F_{score} of 93.10%, and an AUC_{score} of 94.67%. Moreover, for the insect-6 class, the IAEODL-IDC method yielded an $accu_y$ of 91.67%, $sens_y$ of 70%, $spec_y$ of 96%, F_{score} of 73.68%, and an AUC_{score} of 83%.

Figure 8 displays the overall classification outcomes of the IAEODL-IDC system for the 40% TS set. The results demonstrate that the IAEODL-IDC system determined all six types of insects. For the insect-1 class, the IAEODL-IDC technique obtained an $accu_y$ of 95.83%, $sens_y$ of 82.35%, $spec_y$ of 98.06%, F_{score} of 84.85%, and an AUC_{score} of 90.21%. Meanwhile, for the insect-4 class, the IAEODL-IDC method attained an $accu_y$ of 96.67%, $sens_y$ of 92.31%, $spec_y$ of 97.87%, F_{score} of 92.31%, and an AUC_{score} of 95.09%. Moreover, for the insect-6 class, the IAEODL-IDC method reached an $accu_y$ of 98.33%, $sens_y$ of 95%, $spec_y$ of 99%, F_{score} of 95%, and an AUC_{score} of 97%.

Figure 9 exhibits the classifier outcomes of the IAEODL-IDC method with the 70:30 and 60:40 datasets. Figure 9a,c show the $accu_y$ analysis outcomes achieved by the IAEODL-IDC methodology with the 70:30 and 60:40 datasets. The outcome demonstrates that the IAEODL-IDC system obtained the maximum $accu_y$ values with an increase in the number of epochs. In addition, the maximum $valid_n$ over $train_g$ $accu_y$ exhibits that the IAEODL-IDC methodology learns efficiently. Lastly, Figure 9b,d exhibit the loss examination out-

comes of the IAEODL-IDC algorithm with the 70:30 and 60:40 datasets. The outcomes illustrate that the IAEODL-IDC methodology obtained nearby *train_g* and *valid_n* loss values. The IAEODL-IDC system obtained effective outcomes with the test database.

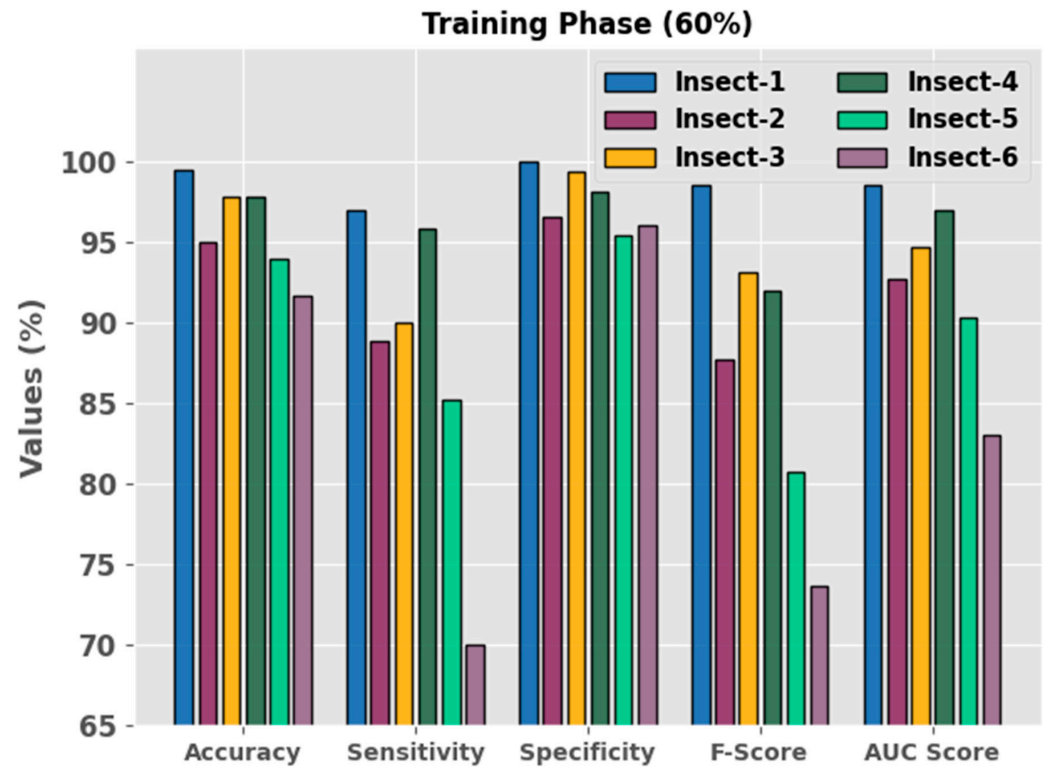


Figure 7. Insect classification outcomes of the IAEODL-IDC system for the 60% TR set.

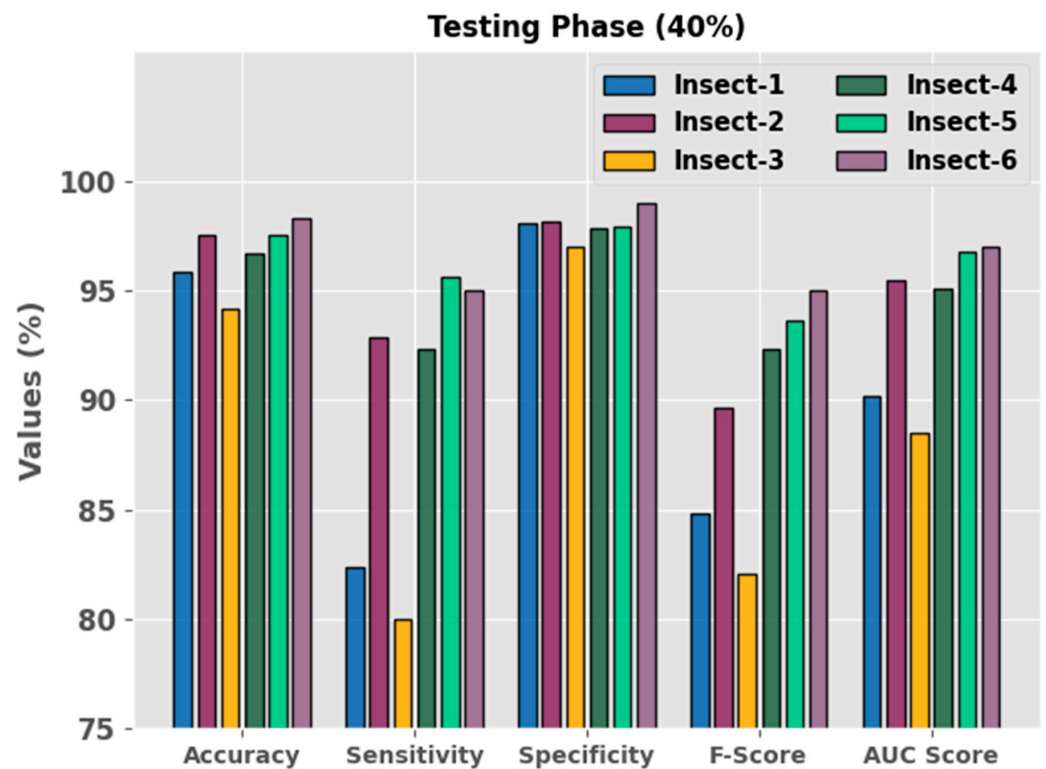


Figure 8. Insect classification outcomes of the IAEODL-IDC methodology for the 40% TS set.

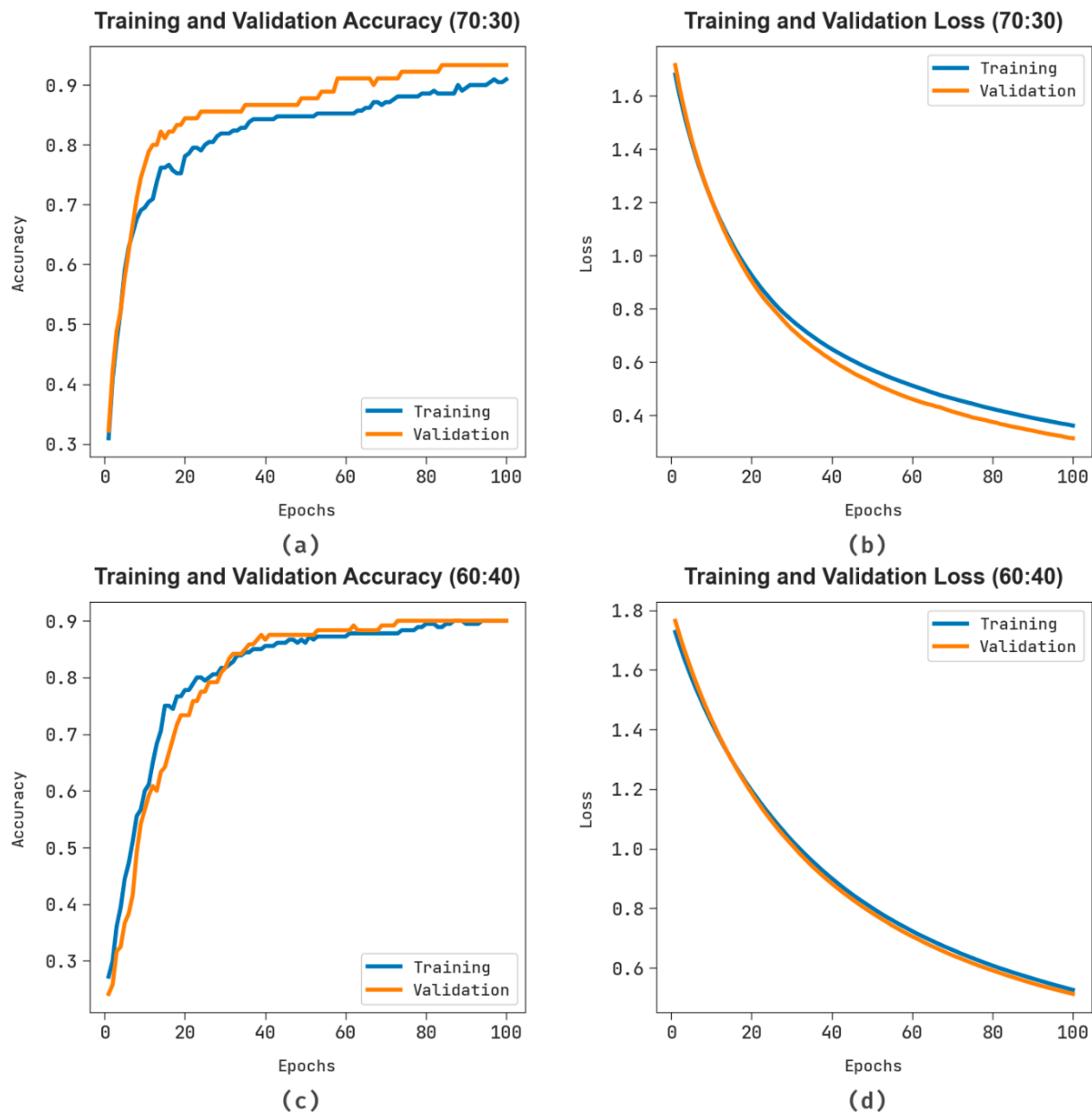


Figure 9. (a,c) $Accu_y$ curve for 70:30 and 60:40 and (b,d) loss curve for 70:30 and 60:40 datasets.

Figure 10 shows the $accu_y$ examination outcomes of the IAEODL-IDC system and other existing methods [14,25,26] for the TR set. The results indicate that the SqueezeNet, LR, CNN, ANN, NB, AlexNet, and ShuffleNet models reached low $accu_y$ values of 94.38%, 94.86%, 94.80%, 94.68%, 94.67%, 93.87%, and 91.99%, respectively. Though the FFGWO-CNN model attained a reasonable $accu_y$ of 95.32%, the IAEODL-IDC technique outperformed all other models and achieved a maximum $accu_y$ of 96.19%.

Figure 11 depicts the $accu_y$ examination results achieved by the proposed IAEODL-IDC technique and other methods for the TS set. The outcome demonstrates that the SqueezeNet, FFGWO-CNN, CNN, ANN, NB, AlexNet, and ShuffleNet methods attained the lowest $accu_y$ values of 96.55%, 94.58%, 92.56%, 92.52%, 92.16%, 96.06%, and 96.20%, respectively. Though the LR method obtained a reasonable $accu_y$ of 96.98%, the IAEODL-IDC technique outperformed the rest of the methods and achieved the highest $accu_y$ of 97.78%.

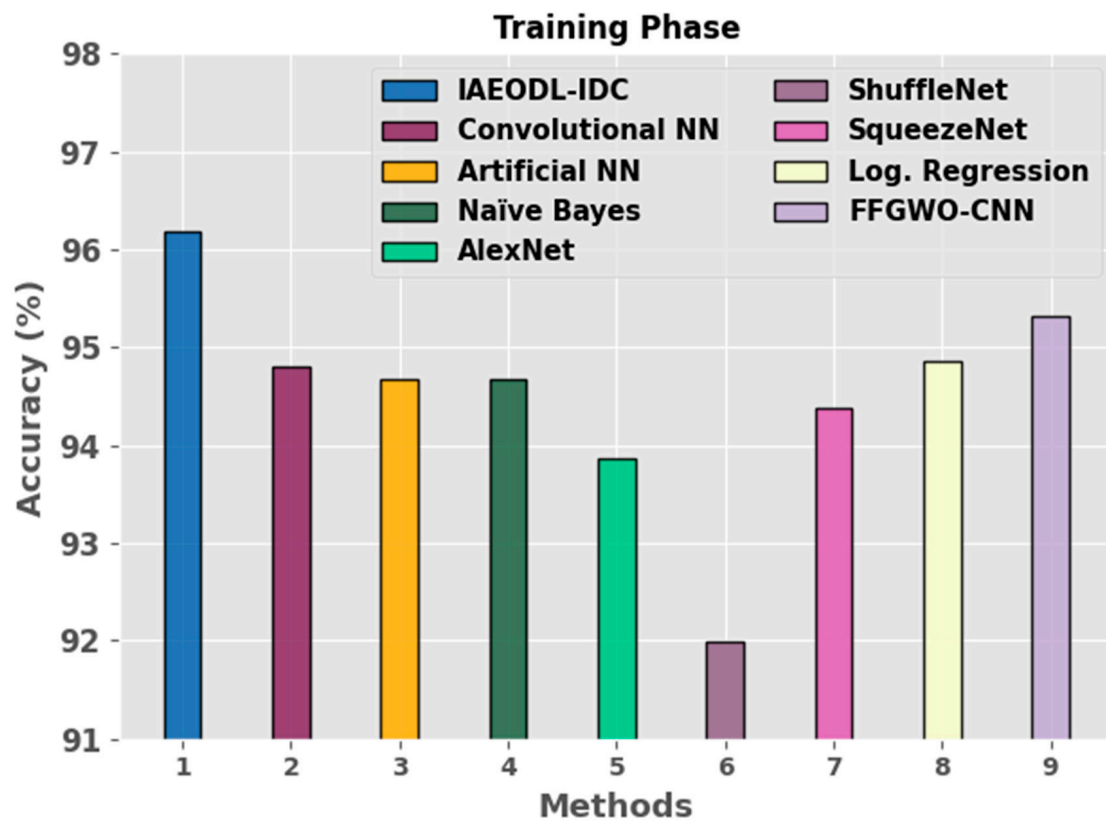


Figure 10. $Accu_y$ analysis outcomes of the IEODL-IDC approach and other systems for TR set.

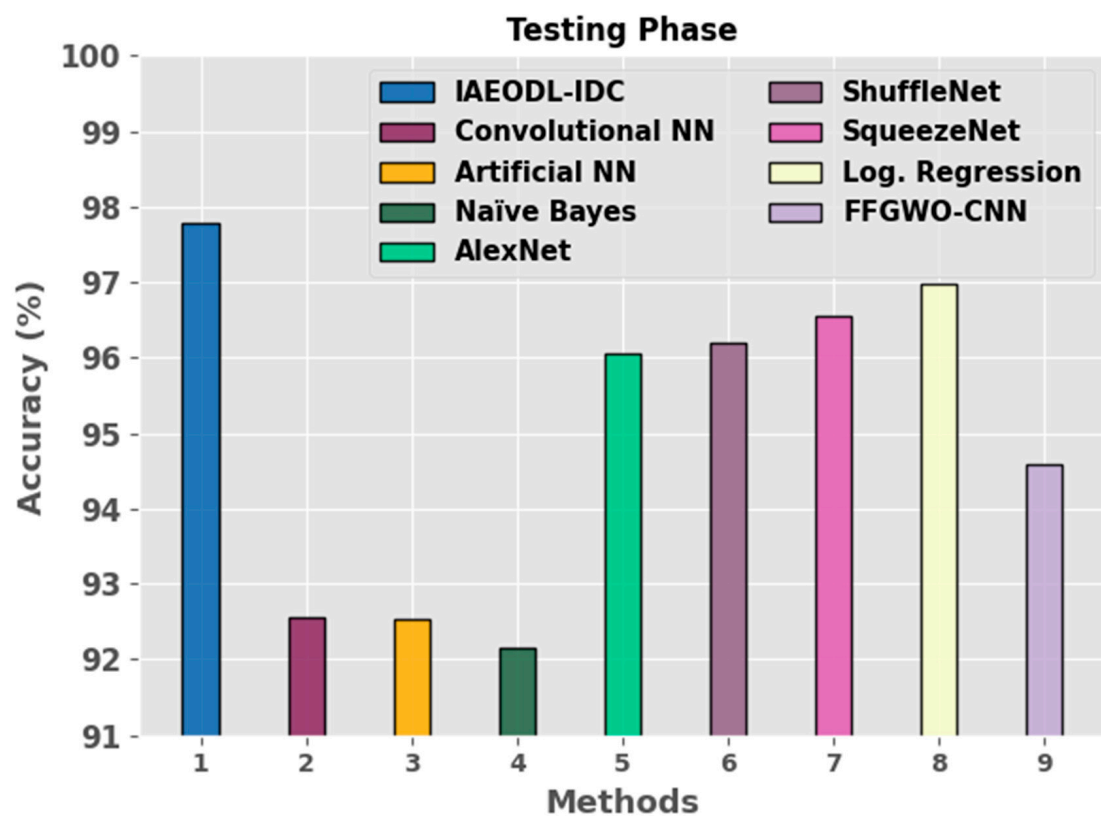


Figure 11. $Accu_y$ analysis outcomes of the IEODL-IDC approach and other systems for TS set.

Lastly, a computation time (CT) analysis was conducted for the IAEODL-IDC methodology and other existing approaches, and the results are shown in Figure 12.

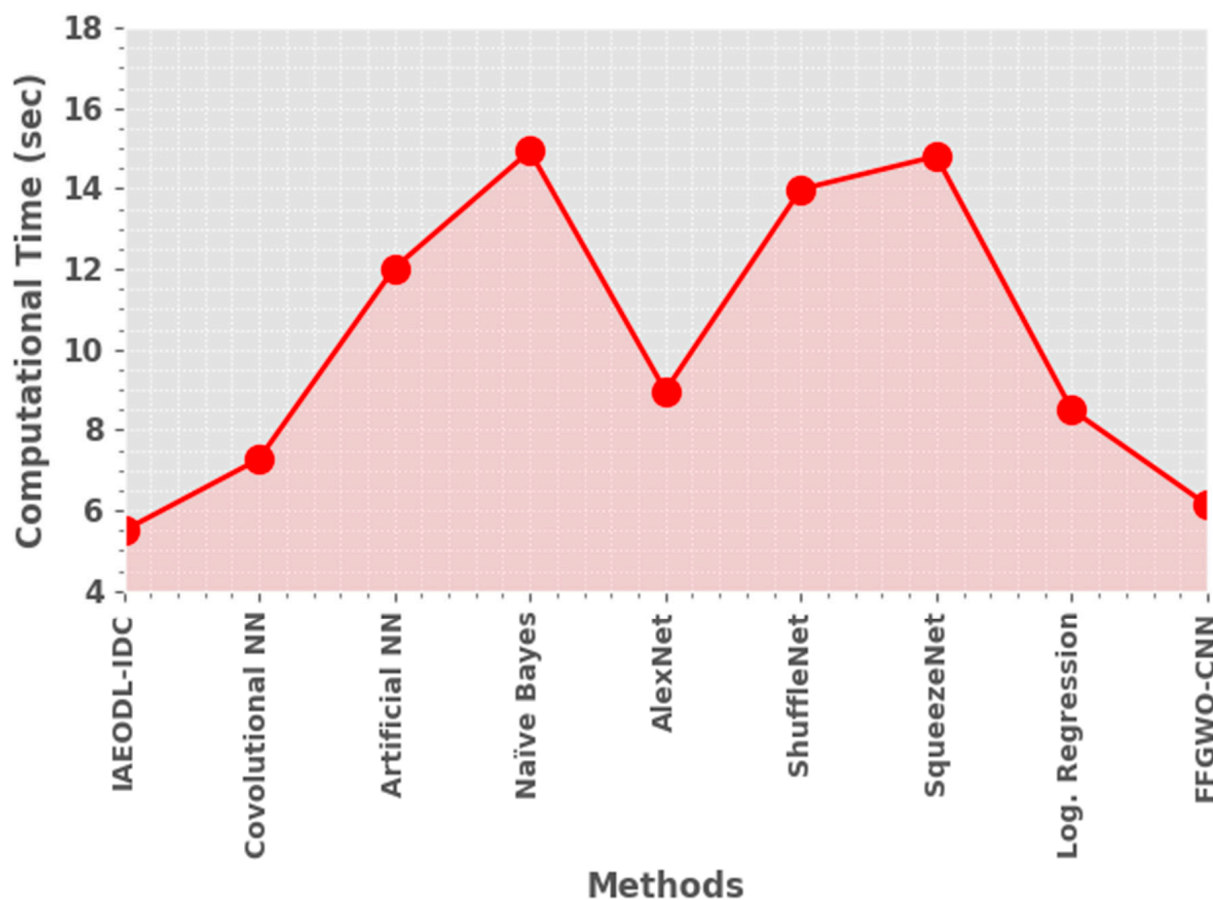


Figure 12. CT outcomes of the IAEODL-IDC algorithm with recent systems.

The outcomes demonstrate that the IAEODL-IDC technique achieved better performance over existing models with a minimal CT of 5.49 s. These results establish the superior performance of the IAEODL-IDC system as an insect classification model.

5. Conclusions

In the current study, the authors present an automatic insect detection technique named the IAEODL-IDC technique, which properly classifies various types of insects in an IoT-assisted environment. The presented IAEODL-IDC technique comprises MF-based noise removal, MobileNetv2 feature extraction, an IAEO-based hyperparameter optimizer, and GRU-based classification processes. Additionally, the IAEO method is utilized for optimal hyperparameter tuning of the MobileNetv2 system. Moreover, the GRU methodology is exploited for effectual recognition and classification of insects. An extensive range of simulations were conducted to demonstrate the superior performance of the IAEODL-IDC algorithm; the simulation results highlight the remarkable outcomes of the IAEODL-IDC methodology in comparison with the existing systems. In the future, a multimodal feature fusion procedure can be designed to improve the outcomes of the IAEODL-IDC technique.

Author Contributions: Conceptualization, M.A.; Methodology, M.A., H.A.M. and A.M.; Software, A.M.; Validation, F.K.; Investigation, H.A.M.; Resources, F.K.; Data curation, F.K.; Writing—original draft, M.A., H.A.M. and A.M.; Writing—review & editing, H.A.M. and A.M.; Visualization, F.K.; Supervision, F.K.; Project administration, A.M.; Funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/48/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R114), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2023R459), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University, project number (PSAU/2023/R/1444).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

1. Saranya, T.; Deisy, C.; Sridevi, S.; Anbananthan, K.S.M. A comparative study of deep learning and Internet of Things for precision agriculture. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106034. [\[CrossRef\]](#)
2. Kiobia, D.O.; Mwitta, C.J.; Fue, K.G.; Schmidt, J.M.; Riley, D.G.; Rains, G.C. A Review of Successes and Impeding Challenges of IoT-Based Insect Pest Detection Systems for Estimating Agroecosystem Health and Productivity of Cotton. *Sensors* **2023**, *23*, 4127. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Hadipour-Rokni, R.; Asli-Ardeh, E.A.; Jahanbakhshi, A.; Sabzi, S. Intelligent detection of citrus fruit pests using machine vision system and convolutional neural network through transfer learning technique. *Comput. Biol. Med.* **2023**, *155*, 106611. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Tiwari, V.; Joshi, R.C.; Dutta, M.K. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecol. Inform.* **2021**, *63*, 101289. [\[CrossRef\]](#)
5. Shin, J.; Mahmud, M.; Rehman, T.U.; Ravichandran, P.; Heung, B.; Chang, Y.K. Trends and Prospect of Machine Vision Technology for Stresses and Diseases Detection in Precision Agriculture. *AgriEngineering* **2023**, *5*, 20–39. [\[CrossRef\]](#)
6. Gomes, J.C.; Borges, D.L. Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification. *Agronomy* **2022**, *12*, 1733. [\[CrossRef\]](#)
7. Sambasivam, G.; Opiyo, G.D. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egypt. Inform. J.* **2021**, *22*, 27–34. [\[CrossRef\]](#)
8. Sullca, C.; Molina, C.; Rodríguez, C.; Fernández, T. Diseases detection in blueberry leaves using computer vision and machine learning techniques. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 656–661. [\[CrossRef\]](#)
9. Chen, J.W.; Lin, W.J.; Cheng, H.J.; Hung, C.L.; Lin, C.Y.; Chen, S.P. A smartphone-based application for scale pest detection using multiple-object detection methods. *Electronics* **2021**, *10*, 372. [\[CrossRef\]](#)
10. Jayswal, H.S.; Chaudhari, J.P. Plant leaf disease detection and classification using conventional machine learning and deep learning. *Int. J. Emerg. Technol.* **2020**, *11*, 1094–1102.
11. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woźniak, M. IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* **2021**, *21*, 5386. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Kusriani, K.; Suputa, S.; Setyanto, A.; Agastya, I.M.A.; Priantoro, H.; Chandramouli, K.; Izquierdo, E. Data augmentation for automated pest classification in Mango farms. *Comput. Electron. Agric.* **2020**, *179*, 105842. [\[CrossRef\]](#)
13. Dai, M.; Dorjoy, M.M.H.; Miao, H.; Zhang, S. A New Pest Detection Method Based on Improved YOLOv5m. *Insects* **2023**, *14*, 54. [\[CrossRef\]](#)
14. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques. *Inf. Process. Agric.* **2021**, *8*, 446–457. [\[CrossRef\]](#)
15. Li, Y.; Chao, X. ANN-based continual classification in agriculture. *Agriculture* **2020**, *10*, 178. [\[CrossRef\]](#)
16. Islam, N.; Rashid, M.M.; Wibowo, S.; Xu, C.Y.; Morshed, A.; Wasimi, S.A.; Moore, S.; Rahman, S.M. Early weed detection using image processing and machine learning techniques in an Australian chilli farm. *Agriculture* **2021**, *11*, 387. [\[CrossRef\]](#)
17. Rong, M.; Wang, Z.; Ban, B.; Guo, X. Pest identification and counting of yellow plate in field based on improved Mask R-CNN. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 1913577. [\[CrossRef\]](#)
18. Abdulridha, J.; Ampatzidis, Y.; Qureshi, J.; Roberts, P. Laboratory and UAV-based identification and classification of tomato yellow leaf curl, bacterial spot, and target spot diseases in tomato utilizing hyperspectral imaging and machine learning. *Remote Sens.* **2020**, *12*, 2732. [\[CrossRef\]](#)
19. Shah, A.; Bangash, J.I.; Khan, A.W.; Ahmed, I.; Khan, A.; Khan, A.; Khan, A. Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 505–519. [\[CrossRef\]](#)
20. Dai, M.; Zheng, D.; Na, R.; Wang, S.; Zhang, S. EEG classification of motor imagery using a novel deep learning framework. *Sensors* **2019**, *19*, 551. [\[CrossRef\]](#)

21. Srinivasu, P.N.; SivaSai, J.G.; Ijaz, M.F.; Bhoi, A.K.; Kim, W.; Kang, J.J. Classification of skin disease using deep learning neural networks with MobileNetv2 and LSTM. *Sensors* **2021**, *21*, 2852. [CrossRef]
22. Yang, J.; Chen, Y.L.; Yee, P.L.; Ku, C.S.; Babanezhad, M. An Improved Artificial Ecosystem-Based Optimization Algorithm for Optimal Design of a Hybrid Photovoltaic/Fuel Cell Energy System to Supply A Residential Complex Demand: A Case Study for Kuala Lumpur. *Energies* **2023**, *16*, 2867. [CrossRef]
23. Jin, X.B.; Yang, N.X.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Kong, J.L. Deep hybrid model based on EMD with classification by frequency characteristics for long-term air quality prediction. *Mathematics* **2020**, *8*, 214. [CrossRef]
24. Xiaoping, W. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. Available online: <https://github.com/xpwu95/IP102> (accessed on 12 July 2023).
25. Divya, B.; Santhi, M. Automatic Detection and Classification of Insects Using Hybrid FF-GWO-CNN Algorithm. *Intell. Autom. Soft Comput.* **2023**, *36*, 1881–1898. [CrossRef]
26. Yang, Z.; Yang, X.; Li, M.; Li, W. Automated garden-insect recognition using improved lightweight convolution network. *Inf. Process. Agric.* **2021**, *10*, 256–266. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.