Advanced Sound-Based Pest Detection in Agriculture Using Deep Learning and Adaptive Optimization

[1*]S. Jayakumar and [2]Dr. L. Lakshmanan
[1*]Research Scholar, School of Computing, Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119, Tamil Nadu, India.
[2]Professor and Head, School of Computing, Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119, Tamilnadu, India.
Corresponding author email: jaikumar.s.cse@gmail.com

Abstract
Large agricultural fields need to detect pests for crop health, but conventional techniques like chemical traps and manual inspections are ineffective and time-consuming. Sound-based analyses provide a possible respond by using pests' unique acoustic characteristics for constant, real-time monitoring. The efficacy of existing methods is limited by issues such unclear acoustic data, environmental noise interference, and the requirement for scalable, real-time systems, which make it difficult to accurately identify pest sounds. In order to address these issues, we propose a novel pest detection system that employs deep learning methods with precise sound analysis. Initially, the sound data of distinct pest species are collected from the dataset. The proposed method employs the audio preprocessing techniques in sound analysis includes, audio cleaning or denoising, audio segmentation, feature extraction, data augmentation, normalization and scaling, labeling, and data splitting. It incorporates methods like the MFCC algorithm, FFT, DFT, STFT, Hann window, hop window, and HPF. The suggested method then evaluated statistical data, such as the mean, median, variance, and standard deviation of each sound, and used feature extraction to create a feature matrix. Subsequently, the generated feature matrix is fed into the fine-tuned hybrid deeplearning model known as, Inception-ResNet model for sound analysis. The adaptive Pelican Optimisation (APO) Algorithm is used to enhance the Inception-ResNet model's hyperparameters. Accuracy, sensitivity, specificity, and F1-score are among the evaluation metrics used to evaluate the efficacy of the suggested strategy. According to the experimental findings, the suggested pest detection model outperformed state-of-the-art methods with an accuracy of 99.89%. The entire framework, model training, and evaluation, is implemented using Python.

Keywords: Pest detection, audio preprocessing, feature matrix, statistical measurements, Inception-ResNet, Adaptive Pelican Optimization
How to cite this article: Jayakumar S, Lakshmanan L. Advanced Sound-Based Pest Detection in Agriculture Using Deep Learning and Adaptive Optimization. Int J Drug Deliv Technol. 2026;16(51s): 1016-1028. DOI: 10.25258/ijddt.16.51s.84

1. Introduction

One of the most basic means by which humans on Earth receive enough food is through agriculture. It is impossible to exaggerate this economic impact [1]. It plays a crucial part in every country's financial structure. An agriculturalist's financial success is dependent on the quality of the agricultural goods they generate, which is decided by crop growth and production. Pests are a serious threat to agricultural growth [2]. Crop productivity is affected by the aforementioned issue, which lowers output. Given how important agriculture is to the country, protecting these crops is a concern. The detection of insects in crops serves a vital and constantly evolving purpose. In order to address the issues related to farming, including sustainability, effectiveness, and environmental impact, and nutrition security, intelligent agriculture methods must be put into place [3].It is possible to automate irrigation and pest control systems with research and ideas [4]. In order to increase crop productivity, farmers enhance fertilisation, irrigation, and other farming methods. Effective use of fertiliser and water saves both the environment and water. Crop losses and massive epidemics are avoided by early pest and disease detection [5]. Farmers have more autonomy due to remote agriculture. Utilising sustainable farming methods reduces waste and resource use.

In recent years, rice weevil and smaller grain beetle groups have been successfully detected by acoustic measures. The commercialising has occurred for the usage of sound detectors to find hidden insects in products that are stored. Additionally, similar devices are being developed, among them the portable Postharvest Insect Detection System (PDS) [6], which records signals using electret microphones and compares them to sound to estimate the extent of infestation. By evaluating the spectra acquired from sound capturing employing a free-field microphone [7], the bio-acoustic identification approach and artificial neural network (ANN) in forecasting fault density in preserved green gramme were capable of to accurately identify and categorise major storage bug pests [8]. However, the creation and uptake of low-cost, easily navigable acoustic instruments for the identification and tracking of economically significant insect pests in grains that have been stored have fallen behind. Insect research has heavily relied on breakthroughs in a range of domains, particularly image classification, voice recognition, and analysis of videos. Deep neural networks and convolutional neural networks (CNNs) are utilised to calculate the short-time Fourier transform (STFT) and mel-frequency cepstral coefficients (MFCCs) of sound data [9]. Sound data must be preprocessed before being applied to a deep learning algorithm. Because audio data is multidimensional via multiple frequencies and amplitudes which modify significantly over time, it is recommended to obtain a feature value which represents the signal properties via preprocessing [10]. Preprocessing transforms a sound the flow into spectral-temporal information that can be used to display the frequency and energy shift over time as an image.

These feature values are useful since CNNs designed for image classification can readily use them. In order to gather features and lessen dimensionality, these CNN models use a convolutional kernel, which considers neighbouring components of the feature values while calculating weight. Throughout this process, many sounds are identified, and patterns of energy modulation based on frequency and time are successfully learnt. A key component of smart precision agriculture is crop health monitoring, which particularly determines the infectious condition of insect pests on the farm [11]. Conventional methods and manual insect pest detection are inadequate, costly, and time-consuming. As a result, farmers place a high premium on early plant pest detection in order to apply the appropriate insecticides and prevent crop loss. In order to address these issues, we propose an entirely novel pest detection system that combines deep learning methods with advanced sound analytics. Our technology uses specialized microphones placed across the field to record acoustic signals. After that, pre-processing is done to remove extraneous background noise. We accurately identify insect activity by analysing the acoustic features using deep learning algorithms. With its scalable design and capacity to function in real-time across vast agricultural areas, the system seeks to give farmers timely, useful insights to maximise pest control and minimise crop damage.

2. Literatiure Review

We examined several studies on the implementation of pest audio analysis obtained from bugs in various environments for pest identification. Based on our review, we produced a survey of works from [12-19].

A Deep Learning-powered pest detection system using sound analytics has been offered by Md. Akkas Ali et al. [12] for the Internet of Agricultural Products (IoT). In order to identify economically viable pests in big agricultural fields, the author employed a variety of acoustic preparation techniques, including Chebyshev filters, FFT, DFT, STFT, and LPC algorithms, to analyse pest audio sounds. A Multilayer Perceptron (MLP) model was employed for training, validation, and evaluation after the system analysed 800 pest sounds from 16 different pests using audio and PID sensors. Through proactive insect identification, the proposed MLP model exceeds current methods like DenseNet, VGG-16, YOLOv5, and ResNet-50, achieving remarkable metrics like accuracy and F1 score, demonstrating its potential to increase agricultural output and financial gains. However, the method is limited in that it only considers 16 pest species.

Omar Reyad et al. [13] developed an intelligent IoT-aided early detection system for red palm weevils (RPWs) in date palm farms by fine-tuning transfer learning classifiers, specifically InceptionResNet-V2. The suggested architecture used TreeVibes sound sensors put on palm trunks to record brief vibration sounds, which were then routed to a cloud server for online analysis. Using the TreeVibes database, the classifier outperformed prior experiments, with a classification accuracy of 94.53% ± 1.69 after 10-fold cross-validation. This revolutionary method demonstrated great promise in assisting farmers with early RPW detection and enhancing agricultural practices. However, the approach relied on cloud computing, which might cause difficulties in resource-constrained environments.

In order to identify and categorise the primary insect in preserved rice grains, Carlito B. Balingbing et al. [14] created an ecologically friendly aural detection system using a multi-layer CNN and a MEMS microphone. Using spectrogram-based feature analysis, the system accurately described the sound profiles of Triboliumcastaneum and Sitophilus with an 84.51% classification accuracy. The inexpensive system offered a non-chemical and non-destructive insect monitoring option by combining a Raspberry Pi Zero computer with a MEMS sensor. The study emphasised the system's ability to guide farmers towards timely insect control without the use of dangerous chemicals. However, the system suffered data collecting issues, which may limit access and scalability for resource-constrained users.

José Neiva Mesquita-Neto et al. [15] explored the possibility of machine learning algorithms to automate the taxonomic detection of tomato-pollinating bees based on their buzzing noises, with the goal of assisting farmers in finding effective pollinators for increased crop yields. The study compared the performance of ML classifiers utilising MFCC to those based on fundamental frequency, and discovered that MFCC-powered classifiers, particularly SVM, outperformed others. The buzzing noises produced during sonication were discovered to be more relevant for taxonomic recognition than flying sounds, with a maximum accuracy of 73.39%. Despite its ability to lessen reliance on expert taxonomists, the approach encountered issues with domain dependency, with classifiers performing inconsistently across domains. However, the technique struggled to attain accurate classification due to a lack of data and domain-specific restrictions.

Tae-Young Heo et al. [16] presented the use of deep learning models for acoustic scene categorization of beehive sounds to detect irregularities that harm honeybee populations. The study compared machine learning algorithms and preprocessing techniques, such as Mel spectrograms and MFCCs, for identifying bee and non-bee sounds. The VGG-13 model that included MFCCs outperformed the other approaches in terms of accuracy among the models that were tested. The CNN model recognised sound sequences and highlighted important components for classification, as demonstrated by the study's use of Grad-CAM to illustrate the categorising procedure. This method revealed the feasibility of developing an automated monitoring system for early detection of aberrant conditions in beehives. However, the model had problems in managing overlapping noises, needing more work for multiclass classification settings.

Md. Akkas Ali et al. [17] introduced an IoT-based system involving advanced deep learning algorithms for pest diagnosis, prevention, and control in large agricultural areas. The authors used the MF-MDLNet model to extract unique features for training and validation by processing audio data from 9,600 pest noises using approaches such as MFCC, BFCC, and LFCC. The suggested system, consisting of an ultrasonic generator and a solar-powered network, exceeded traditional approaches such as VGG 16, YOLOv3, and DenseNet in terms of accuracy, precision, and recall. This novel approach not only lowered pesticide use, but also allowed for real-time insect monitoring and control, supporting environmentally friendly farming practices. However, the model's computing costs are extremely high.

Peter Knebel et al. [18] suggested an Artificial Intelligence of Things (AIoT)-based technique for detecting bark beetle infestations in trees. The study used affordable IoT devices to collect sound and pheromone data, which were then analysed using off-the-shelf algorithms, indicating the viability of detecting infestations early on. Field experiments in the Hunsrück-Hochwald National Park validated the system's potential, and plans to expand it with additional environmental sensors for drought stress assessment and improved anomaly detection models, such as kNN and SVM, were suggested. The system also looked into solutions such as LoRaWan for data transmission in areas with low network connectivity. However, the lack of clarity about the technical execution of information transfer remained a constraint.

Rodrigo Capobianco Guido et al. [19] created an auditory recognition method for Cicadidae pests in plantations of coffee that employs the Bark Scale (BS), Wavelet-Packet Transform (WPT), Paraconsistent Feature Engineering (PFE), and Support Vector Machines. The system was evaluated on a dataset of 1,366 recordings and obtained a 96.41% accuracy in differentiating cicada noises from background noise, which included a variety of environmental sounds like birds and industrial. The solution, which was designed for low-cost, real-time application, showed promise in assisting farmers by lowering manual work and improving detection precision. However, the strategy relied on a small dataset, which could restrict its applicability to more diverse environmental situations.

3. Proposed Methodology

Using a hybrid deep learning model, the proposed system aims to detect pest sounds accurately. The initial phase in the process is audio pre-processing, which involves a number of processes for input raw sound data, such as data augmentation, audio segmentation, normalisation and scaling, feature extraction, labelling, data splitting, and cleaning or denoising. To extract relevant features from the sound, a number of pre-processing techniques are employed, including Hann window, hop window, MFCC, DFT, FFT, and STFT. Following preprocessing, a feature matrix is generated. This matrix is then given to a fine tunedhybrid Inception-ResNet model for sound analysis, which classifies the input into two categories: detected pest classes and pests that were not detected. By fine-tuning the crucial hyperparameters, the APO Algorithm improves the performance of the classifier model thus making it more effective at correctly identifying the pest sounds. The structure of the proposed work is depicted in the figure 1.

Figure 1: Overall architecture of proposed framework

3.1. Data collection

The USDA-ARS collecting system has the dataset of pest sounds used in this study [20], and the hearing frequencies of these pests are available. The recordings of audio for the gathering were produced by recording the sound emission produced by many kinds of insects using a range of acoustic sensors. The potential to record and capture acoustic signals produced by pests is made possible by setting up an IoT network across large fields. The 96 pests produced a total of 9,600 sound signals that were evaluated as part of the research methodology employed in this study.

3.2. Audio preprocessing

Deep learning requires a number of successive steps to prepare audio files for insect identification. In order to guarantee that the data is properly prepared for later feature collection and training of models, several steps are required. During the capturing procedure, environmental circumstances change the pest's auditory noises, causing audio interruptions. In relation to these instances, the audible sound of the pest must be pre-processed. This entails denoising, spectral leakage reduction, converting overlapping frames to non-overlapping frames, cleaning, improving, and converting the raw audio signals into a format suitable for further modelling and assessment. The following describes the sound analysis methods and audio pre-processing stages that the recommended system is employed for pest acoustic sound data:

3.2.1. Audio denoising or cleaning

In audio files, denoising step eliminates background noise. This can be achieved using a variety of techniques, including spectrum subtraction, noise reduction, and filtering. Non-pest-related frequencies can be filtered out, and any unnecessary noises can be eliminated using noise reduction techniques. The proposed approach employs a high pass filter to clean and denoise pest noises.

High pass filter: The elements of this particular HPF include a passive filter and an operational amplifier. The op-amp's bandwidth and gain specifications thus define the cutoff frequency of this filter, which acts as a bandpass filter. The bandwidth limit of an amplifier is thus determined. With a rise in the input frequency, the amplifier gain gradually decreases until it reaches 0 dB. Combined with low tolerance components such as capacitors and resistors, this operational amplifier HPF offers remarkable performance and accuracy. The frequency of the input signal, denoted by f and expressed in Hertz, determines the strength of a filter when employing a functioning amplifier which does not change its signal. The symbol for the cut-off frequency is f_c.

$$z(k) = \frac{\mathrm{Af}\left(\frac{f}{f_c}\right)}{\sqrt{1+\left(\frac{f}{f_c}\right)^2}}$$

(1)

$$\mathrm{Af} = 1 + \frac{r_2}{r_1}$$

(2)

3.2.2. Audio segmentation

Recordings must be divided into smaller segments or time periods in order to isolate specific pest sounds or vocalizations. The use of windowing or framing techniques allows audio signals to be separated into frames that overlap or do not overlap at a predetermined period. The proposed method eliminated spectral leakage by using the Hann window. Consider of *W_h(k)* as the Hann window function and *Z(k)*as a spectral leakage signal from a single audio source. The suggested system assessed the convolution that resulted among *z(k)* and *W_h(k)*, as shown in Eqn. (3&4), and produced the final signal *Z_{Wh}(k)* while removing spectral leakage.

$$W_h(k) = \frac{(1-\cos((2\pi k)-(k-1)))}{2} \qquad (3)$$

$$Z_{W_h}(k) = Z(k) \cdot W_h(k)$$

(4)

The hop window method was used in the suggested method to convert overlapping frames into non-overlapping ones. The overlap factor is divided by the FFT size to determine Fere's hop size. This is the total quantity of samples between each subsequent FFT window. For example, 256 samples make up the hop size due to the 512 frame size and the 2 overlap. H stands for the hop, l for the window length, n for the current discrete time, and $\left(\frac{n-H}{l}\right)$for the percentage of overlap.

$$\sum_{m=-\infty}^{+\infty} w\left(\frac{n-\mathrm{mH}}{l}\right) = 1$$

(5)$X_m[n-\mathrm{mH}] = X[n]w\left(\frac{n-\mathrm{mH}}{l}\right)$

(6)$X_m[n] = X[n+\mathrm{mH}]w\left(\frac{n}{l}\right)$

(7)

$$X[n] = X[n] * 1$$

(8)

$$= X[n] * \sum_{m=-\infty}^{+\infty} w\left(\frac{n-\mathrm{mH}}{l}\right)$$

(9)

$$X[n] = \sum_{m=-\infty}^{+\infty} X_m[n-\mathrm{mH}]$$

(10)

Thus, $X_m[n]$ is an audio frame of l length and a finite length. It depicts the sound close to $X[n+\mathrm{mH}]$, or m hops from the start. Each input frame $X_m[n]$ is converted into an output frame $Y_m[n]$ after processing, and the output is reconstructed using the overlapping addition of the output frames in the same manner as the input.

$$Y[n] = \sum_{m=-\infty}^{+\infty} Y_m[n-\mathrm{mH}]$$

(11)

3.2.3. Spectral transformations using FFT, DFT, and STFT

FFT:In digital signal processing, FFT may extract frequency domain features from incoming data. There are numerous applications for these attributes. These functions can therefore be processed by the neural network. Multibranch residual synthesis is a technique used by deep neural networks to recover frequency domain information from input signals using the FFT. An IoT-based pest detection system with sound analysis may help farmers control insect infestations over large agricultural areas. This approach can lessen chemical damage while increasing crop productivity. Time-domain signals are transformed into frequency-domain representations using the FFT method. In this inquiry, additional signals were examined. Since frequency domain analysis separates frequencies, it can be applied to a variety of situations.

$$F(\alpha) = \int f(m) e^{-\iota\alpha m}\mathrm{dm} \quad (12)$$

$$f(m) = \frac{1}{2\pi}\int F(\alpha) e^{\iota\alpha m}\mathrm{dm} \quad (13)$$

DFT: To reduce noise in a signal, the DFT attenuates or selectively eliminates undesirable frequencies. Information is obtained from frequency domain signals using the DFT in DL and audio analysis. These characteristics can be used by a machine learning system to identify and categorise problems. To determine the average frequency of sound bands related to pest species, the spectral centroid is calculated.

$$P(s) = \sum \mathrm{Hw}(|U(l\cdot a)| - |U(l-1,a)|) \quad (14)$$

$$C = \sum f \frac{p[f]}{\sum p[f]} \quad (15)$$

$$\sum U^2[f] = 0.85 \sum U^2[f] \quad (16)$$

Where, $P(s)$ is the spectrum flux, while $\mathrm{Hw}(U)$ represents the half-wave rectifier function. f represents the frequency domains, C denotes the spectrum centroid, and $\mathrm{N}f$ is the maximum frequency. Fs stands for the Nyquist frequency, and Fsr for the roll-off point for spectral. The formula can be used to calculate the roll-off level spectrum, which is related to the fundamental chopping frequencies.

STFT: When audio frames crossover, they need to be separate from one another. Both the particular use case and the degree of pest noise determine the window's size and overlap. Apply each frame to the Fourier Time-transform. STFTs produce spectrograms that show how a signal's frequency content changes over time. Analysing pest sounds has many other benefits. For every time interval, the component with the highest amplitude corresponds to the dominant frequency.

$$u_l(p) = \sum u_l p\,[n - \mathrm{mH}] e^{-\iota\alpha k} \quad (17)$$

3.2.4. Feature extraction

A pest sound signal is converted into a compact representation of its spectral properties via the MFCC computation. The MFCC computation is a multi-step process for extracting important characteristics from auditory sources. The preprocessed audio signals from the pest are used as input for the feature extraction stage. MFCCs, PSD, and spectral contrast are among the spectral features commonly used in audio analysis for pest identification. Other characteristics, such as pitch, duration, or amplitude, may be crucial depending on the kind of pest being targeted. The proposed system uses the MFCC approach to determine the feature matrix for the pest sound.

It is notable that using the features extracted from the sound waveform as a basic model produces better results than using the raw data as feed. In audio signal processing, the MFCC technique is widely used for feature extraction. The MFCC method is implemented in the following way to create the feature matrices from the pest acoustic data:

1. It includes measuring a signal's Amp or PS.
2. Subsequent frequency carriers can overlap to produce triangle waves with the same bandwidth and Mel scale.
3. Compiling a summary of each category's data.
4. Each logarithmic total is estimated.
5. Performing the band's DCT computations.
6. Removing the cosine transformation's earlier components. The suggested system used the appropriate expression to calculate the Mel-scale.

$$M = 25951\log_{10}\left(\frac{f}{700}+1\right) = 11271\log_{e}\left(\frac{f}{700}+1\right) \quad (18)$$

In order to begin, the time-domain signal is transformed into the frequency domain using the Fourier Transform, as shown in equation (). DFT and STFT are used to process the audio signal in order to examine its frequency components and record changes in the spectral content over time. The coefficients, which capture the key spectrum features of the audio signal for additional study, are finally returned. The feature matrices from the list of recorded pest sounds were fed into the suggested method.

3.2.5. Data augmentation

The training data is made more diverse by generating more audio recordings with varying pitch, speed, or background noise levels. This is carried out to increase the range of training data and enhance the model's capacity to extrapolate its results to different contexts or situations.

3.2.6. Normalization and scaling

When normalised or scaled, the extracted audio features are guaranteed to be on the same scale, with a mean of zero and variance of one. In the process of training the model, this can assist keep some qualities from being prioritised over others.

3.2.7. Labeling and data splitting

In order to determine whether pests are present or not, the audio data has already undergone preprocessing by applying pertinent labels or annotations. These labelled data can be used as training data for supervised detecting methods that create pest detection models.

The proposed system computed the MFCC feature matrix, mean (μ), median (β), variance (σ^2), and standard deviation (σ) from each pest sound in order to train pest identification models using deep learning techniques. The previously processed audio data should be separated into three sets for testing, validation, and training in order to build and analyse models. After being trained on the training set, the hyperparameters of the model are modified, it is chosen on the validation set, and its performance is assessed on the testing set. Within this proposed system, training accounted for 70% of the data, testing for 20%, and validation for 10%.

3.3. Feature Matrix Generation

The suggested method obtained the feature matrix as an example from the feature matrix production stage by using our feature extraction structure. After that, it determined each sound's mean, median, variance, and standard deviation, among other statistical measurements, and saved them in the database for later examination. Algorithm 1 mentions the pest sounds analytics algorithm.

$$\mu = \frac{\sum x_i}{N} \tag{19}$$

$$\beta = \begin{cases} \text{size of } \left\{\left(\frac{n}{2}\right)\text{th} + \left(\frac{n}{2}+1\right)\right\}\text{th observation; if } n \text{ is even} \\ \text{Size of } \left(\frac{n}{2}+1\right)\text{th observation; if } n \text{ is odd} \end{cases} \tag{20}$$

$$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}} \tag{21}$$

$$\sigma^2 = \frac{\sum(x_i-\mu)^2}{N} \tag{22}$$

Algorithm 1: Analysis of pest sound

Input: Pest sound

Output: Mean(μ), Median (β), Standard deviation (σ)Variance (σ^2), Feature matrix $M = []$

```
//Step 1: Calculate the feature matrix
for i=0 to z, do
    for i=0 to z, do
    Calculate $z(k)$ and $W_h(k)$ utilizing eqn. 1, and 3correspondingly
    $Z_{W_h}(k) = z(k) * W_h(k)$
        for m=0 to N, do
            $Y[n] = \sum_{m=-\infty}^{+\infty} Y_m[n - \mathrm{mH}]$
end
        Calculate: $f(m) = \frac{1}{2\pi}\int F(\alpha)e^{i\alpha m}\mathrm{dm}$
        for $m = \frac{\mathrm{Nf}}{2}$ to $\frac{\mathrm{Nf}}{2} - 1$ do
            $P(s) = \sum \mathrm{Hw}(|U(l \cdot a)| - |U(l-1,a)|)$
        end
        for f=1 to Nf, do
            $C = \sum f \frac{p[f]}{\sum p[f]}$
        end
        Compute: $M[] = 11271\log_e\left(\frac{f}{700}+1\right)$
//Step 2: Compute statistical measures
for i=1to N, do
        $\mu = \frac{\sum x_i}{N}$
    If mod n=0
        $\beta = \left\{\left(\frac{n}{2}\right)\mathrm{th} + \left(\frac{n}{2}+1\right)\right\}$
    else
        $\beta = \left(\frac{n+1}{2}\right)$
    else if
        $\sigma^2 = \frac{\sum(x_i-\mu)^2}{N}$
        $\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}}$
    return $\mu, \beta, \sigma^2, \sigma$
end
    return $M = []$
end
```

3.4. Sound based detection using fine-tuned Inception-ResNet model

Following feature matrix production, the proposed framework uses the fine-tuned Inception-ResNet-V2 hybrid deep learning model for sound analysis. With the help of the multi-scale feature evaluation capabilities of Inception modules and the residual connection (skip connections) concept from ResNet, Inception-ResNet-V2 is a strong and advanced convolutional neural network architecture that enables gradient flow during training.

After this fusion, a deep neural network with many residual blocks is produced. These blocks are made up of complex parallel convolutional pathways that capture patterns at different scales. Using global average pooling and stacking these blocks, Inception-ResNet-V2 is excellent at identifying complex audio patterns. The structure of a residual block in Inception-ResNet-V2 usually consists of input, a convolutional layer, and Inception Modules, Convolution, and Merge elements. Multiple residual blocks are stacked by Inception-ResNetV2, and the number of blocks and Inception modules within each block can be changed to alter the network's depth. These residual blocks are stacked to create the entire network, which is then followed by global average pooling and a fully connected layer to generate the final classification output. Figure 2 shows the architectural layout of this Inception-ResNetV2 model and the figure 3 depicts the stem block of the model. In the task of pest sound identification and classification, the

Inception-ResNetV2 model's unique design and efficient filtering have produced outstanding results.

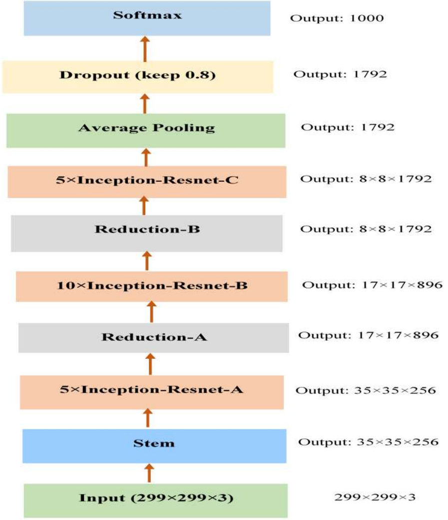


Figure 2: Architecture diagram of Inception-ResNetV2 model

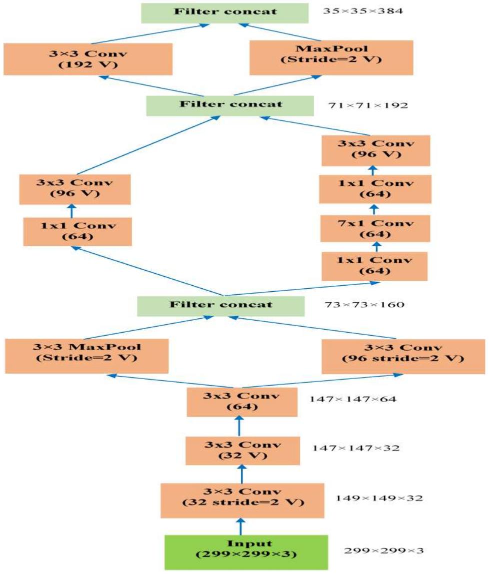


Figure 3: Structure of Stem block utilised in Inception-ResNet-V2 architecture

3.4.1. Adaptive Pelican Optimization Algorithm for hyperparameters optimization of Inception-ResNet-V2

In this study, the Inception-ResNet model's hyperparameters are optimised using the APO algorithm. It has the benefits of easy computation and quick convergence. The Pelican Optimization algorithm is a natural optimization approach that is motivated by the fascinating hunting behaviours of pelicans (Trojovsky and Dehghani, 2022) [21]. With their long beaks and huge throat pouches, these birds are able to grasp their prey and drain the water. They dwell in groups and hunt both alone and with others. Pelican birds consume a variety of animals, including fish, turtles, reptiles, and crabs. If they were hungry, they would also eat marine stuff. The pelican swiftly plunges into the water to get its prey after locating it. Pelicans hunt fish in shallow water during low tide, making it simple to catch them by extending their wings above the water's surface. When a pelican catches its prey, its mouth fills with too much water. To get rid of extra water, the pelican tilts its head forward before eating. With a mathematical framework and APOA pseudo-code offered for simple explanation, the APOA seeks to mimic pelican hunting behaviour.

Step 1: Initialization: We modify the main hyperparameters of the Inception-ResNet model, including the Learning Rate (L_r), Batch Size (B_s),andweight decay (W_d).Depending on its location within the search space, each pelican in the population of the APO algorithm contributes values to the problem variables. As a result, each pelican represents a solution to the problem, which has been quantitatively represented using its application of an equation and its execution.

$$P = \{p_1, p_2, p_3, \dots \dots, p_m\}$$
(23)

Where, $P$is the population and p_m represents the m^{th} solution.

$$p_1 = [L_{r1}, B_{s1}, W_d]$$
(24)

Step 2: Fitness estimation:-This algorithm evaluates fitness in terms of detection accuracy. The fitness calculation is described as follows:

$$\text{fit}_N = \text{Max (Accuracy)} \qquad (25)$$

Step 3: Update the solutions: The proposed APOA updates potential solutions by simulating the strategies and behaviour of pelicans during prey seeking and attack. There are two phases to simulating this hunting strategy:

(i) Searching for the prey (phase of exploration).

(ii) Winging on the water's surface (phase of exploitation).

Exploration Phase: During the exploration stage, the population's Pelicans find the prey and move into that part of the search arena. Using the search dimension to generate the prey's position arbitrarily is one of this APOA's primary characteristics. The APOA's exploratory potential is enhanced when optimisation issues are resolved. Here is a numerical representation of the exploration process:

$$Z_{i,j}^{m_1} = \{z_{i,j} + \text{rand} \cdot (m_j - R \cdot z_{i,j}), F_m F_r; |$$
(26)

According to the 1st phase, $Z_{i,j}^{m_1}$represents the new position produced for the i$^{\text{th}}$ pelican within the j$^{\text{th}}$ dimension, $F_m$represents its value of objective function i.e. Objective function values are the best measure of the quality of candidate solutions, randrepresents random numbers, A number that can be arbitrarily equal to either 1 or 2 is the parameter$R$.

This parameter is chosen at random for every member and every iteration. By increasing their movement, a member can access newer areas of the search space when this parameter's value is equivalent to two. The POA exploring capability to accurately search the search space is therefore affected by parameter$R$.

The proposed APOA accepts a new location for a pelican if it improves the objective function. Effective updating prevents the algorithm from moving into suboptimal locations. Equation 27 is used to model this process.

$$Z_i = \left\{Z_i^{m_1}, F_i^{m_{1_1}} F_i, \right| \tag{27}$$

Where$Z_i^{m_1}$represents the newly produced position of pelican for the ithstepand $F_i^{m_1}$represents its value according to the first stage's objective function.

The escaping and fighting phase (Exploitation): In order to lift fish higher and collect them in their throat pouches, pelicans spread their wings during the second phase. By using this technique, pelicans capture more fish in the area they assault. The suggested APOA provides the best hunting spots based on a model of pelican behaviour. This process enhances APOA's local search and exploitation capabilities. The algorithm should take into account points close to the pelican's location in order to produce a better result. The hunting behaviour of pelicans is modelled by equation (28).

$$Z_{i,j}^{m_2} = Z_{i,j} + V \cdot \left(1 - \frac{t}{T}\right) \cdot (2 \cdot \text{rand} - 1) \cdot Z_{i,j} \tag{28}$$

Where, $Z_{i,j}^{m_2}$ represents the new position produced for the ith pelican in the jth dimension, $V$is the constant, that is equivalent to 0.2, $V \cdot (1 - tT)$represents the neighbourhood radius of $Z_{i,j}$where t is the counter of iteration and T is the maximum value of iterations. Then the "$V \cdot (1 - tT)$" the coefficient depicts the radius of the population members' neighbourhood, allowing them to hunt for the ideal solution locally. This coefficient improves APOA exploitation power, leading to a more optimal global solution. During first iterations, a high coefficient value leads to a bigger area around each member being considered. As the process replicates, the "$V \cdot (1 - tT)$" coefficient declines, leading to lower radii of each member's neighbourhoods. The POA can arrive at solutions that are closer to the global (or even exact global) ideal based on consumption by conducting fewer and much more precise phases of scanning the region surrounding all members of the general population.

Equation (29) models the process of accepting or rejecting a new pelican position through effective updating.

$$Z_i = \{Z_i^{m_3}, F_i^{m_{1_2}} F_i; | \tag{29}$$

Based on the 2nd phase, $Z_i^{m_3}$represents the new position that has generated for the ithpelican, $F_i^{m_{1_2}}$represents its objective function.

Levy flight function

Levy motion, or Levy flight, is a non-Gaussian stochastic process that employs random walks generated via Levy stabilisation. Pelicans can catch more fish in their hunting area thanks to Levy flight-based jumps, which strike a balance between exploration and exploitation. The power law formula governs the distribution:$L(s) \sim |s|^{-l-\beta}$, where $0 < \beta < 2$ indicates an index and s_L signifies the length of the step.The step length can be determined as follows:

$$s_L = \frac{\mu}{|v|^{1\beta}} \tag{30}$$

Where u and v are derived from normal distributions, which is:

$$\text{uN}(0, \sigma_u^2), \text{vN}(0, \sigma_v^2) \tag{31}$$

Where,

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta) x \sin(\pi\, x\, \beta/2}{\Gamma\left(((1+\beta/2))\, x\, \beta\, x 2^{(\beta-1)/2}\right)}\right] \text{and} \sigma_v = 1 \tag{32}$$

The parameters u and v, which vary from 0 to 1, add variability and randomness to the model.Here, Γ and β is the standard gamma function.

The Levy function is included during winging in the APO algorithm. The following is an evaluation of the new pelican position:

$$Z_i = \left\{Z_i^{m_3} + \alpha \oplus \text{Levy}, F_i^{m_{1_2}} F_i; \right| \tag{33}$$

Where,

$$\alpha = 0.01 \times s_L \times (Z_i^{m_2} - Z_{\text{best}}) \tag{34}$$

Step 4: Termination: Using equations (26) through (29), the procedure is continually iterated. The APO algorithm returns the best solution found during the procedure once it has finished processing.

Algorithm 1: Pseudocode of APO

Input: Learning Rate (L_r), Batch Size (B_s), and weight decay (W_d)

Output: Optimized Learning Rate (L_r), Batch Size (B_s), and weight decay (W_d)

```
Begin APOA algorithm
Provide all the optimization problem information
Initialize the pelican population
Initialize the position of pelican and estimate the objective function using equation (25).
For t=1:T
Produce the location of the prey at random.
For I = 1:N
Phase 1: Searching for the prey (phase of exploration)
For j = 1:m
Compute new status of the jth dimension by Equation (26).
End.
```

```
Apply Equation (27), ith population member
Phase 2: Surface water winging (extraction phase)
For j = 1, m
Calculate the jth dimension's new status usi
Equation (28)
End
Utilising Equation (29), update the ith population member
End
According to the levy flight operator illustrated in Equation 33, the solutions revise their positions.
End
Revise the best possible solution
End.
Provide the best candidate solution discovered through APOA.
End APOA.
```

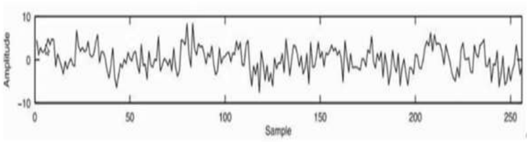
The proposed approach achieves successful pest sound analysis by optimising the Inception-ResNet hyperparameters using the APO algorithm. This optimisation procedure increases the overall performance and quality of sound detection systems, making them both reliable and efficient.

4. Results and Discussions

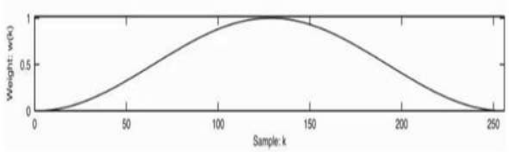
In this study, we investigated an automatic pest detection system that uses pest sounds. We used a novel neural network architecture called the "fine-tuned Inception-ResNet-V2" to complete this classification task. We evaluated the suggested model's performance using standard classification measures. The F1-score, recall, accuracy, specificity, and precision are the main evaluation metrics. The dataset of pest sounds used in this study is from the USDA-ARS collection system. The proposed methodology is implemented using Python. To execute the proposed methods detailed in this study, use a high-performance processor, such as an Intel Core i9 or AMD Ryzen 9, and additionally make sure the system has 32GB or more of RAM.

4.1. Experimental results

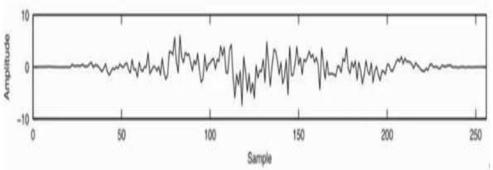
This section describes the results of our experiments, which were indicated to compute the performance of the proposed approach for autonomous pest detection from insect sounds using the finely adjusted Inception-ResNet-V2.



(a)



(b)



(c)

Figure 4: (a) Signal including spectral leakage, (b) Hann window, (c) Signal excluding spectral leakage

Figure 4 shows three graphs exhibiting signal processing operations. The graphic (a) depicts a noisy signal with fluctuating amplitudes throughout 250 samples. The plot (b) depicts a weight function that progressively increases to a peak before symmetrically declining, mimicking a window function applied to the signal. The plot (c) shows the resultant signal after applying the weight function, with the amplitude decreasing towards the edges but maintaining the middle region's structure, suggesting the smoothing or filtering effect of the applied window function.

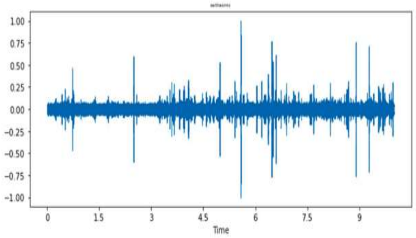


Figure 5: Signal for sample sound

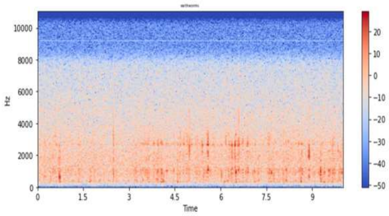
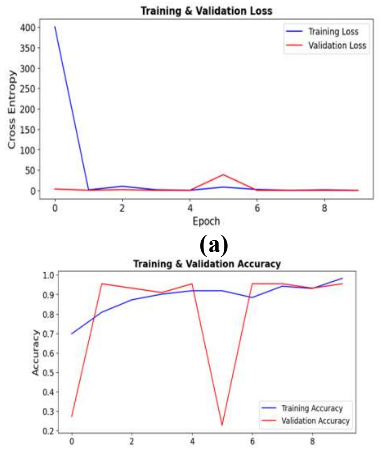


Figure 6: Spectrogram of sample sound

The graphic in figure 5 displays a time-domain signal whose amplitude varies dramatically across the specified time range. The signal has periodic spikes of greater intensity mixed with areas of lower amplitude oscillations, reflecting variable energy levels. The spectrogram shows in figure 6 the frequency content of the signal across time, with frequency (Hz) on the vertical and time on the horizontal axes. The frequency components' intensities are indicated by a colour scale ranging from red (high intensity) to blue (low intensity). Lower frequencies, below 2000 Hz, exhibit considerable energy in red and orange hues, indicating high activity in this region. Higher frequencies above 6000 Hz are predominantly blue, indicating low energy.



(a)

(b)

Figure 7: (a) Loss vs. epoch (b) Accuracy vs. Epoch

The training and validation loss graphs in figure 7 (a) show that the training loss begins at a very high value and rapidly falls throughout the first few epochs before stabilising at a low level. In contrast, the validation loss is constantly low throughout the training phase, with only modest changes. The accuracy graph in figure 7 (b) indicates a consistent improvement in training accuracy, whereas validation accuracy varies dramatically, particularly around the middle epochs, when it decreases sharply but then returns to closely match training accuracy. These trends reveal that, while the model performs well, the sharp validation accuracy decreases could reflect sensitivity to specific batches or noise in the validation data.

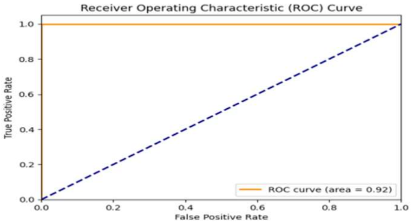


Figure 8: ROC curve

Figure 8 depicts the classification model's performance using the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) versus the False Positive Rate. The curve rises steeply towards the top-left corner, indicating a high degree of sensitivity and specificity. The model's good discriminatory power to distinguish between classes is indicated by the area under the curve (AUC), which is 0.92. The diagonal line represents random guessing, and the ROC curve's substantial divergence from it demonstrates the model's effectiveness in making accurate predictions.

4.2. Ablation study

This research's ablation study is essential because it thoroughly examines the Fine-tuned Inception-ResNet V2 model to determine the role and importance of each component, guaranteeing the model's robustness and effectiveness in identifying pest sounds. An ablation study essentially acts as a diagnostic tool to pinpoint and confirm the key elements that accelerate the model's success, making sure that each one is required and enhances performance as a whole. The study offers insights into how each component impacts the model's performance by carefully analysing variations in the batch size and learning rates. In addition to confirming the architectural choices made, this thorough research finally leads the evolution of a very accurate and trustworthy model for the early identification of pest sounds.

(i) Ablation study: changing Batch Size

Using different batch sizes can result in variations in the classification's performance. As a result, we used a variety of batch sizes in our experiments, including 64, 32, 16 and 8. As Table 1 illustrates, evaluation accuracy increases as batch size rises from 8 to 64. The greatest accuracy of 98.72% is achieved with a batch size of 64. Batch size 64 improves accuracy and is used in future assessments.

Table 1. Performance Measures for the Proposed ModelWithChangingBatch Size

Proposed Fine-tuned Inception-ResNet V2				
Metrics	**Batch size=8**	**Batch size=16**	**Batch size=32**	**Batch size=64**
Accuracy	97.18	97.74	98.13	98.72
Precision	97.17	97.78	98.22	98.79
Recall	97.02	97.60	98.02	98.69
F1 score	97.25	97.78	98.14	98.82
Specificity	97.30	97.68	98.75	98.78

(ii) Ablation study: changing learning rates

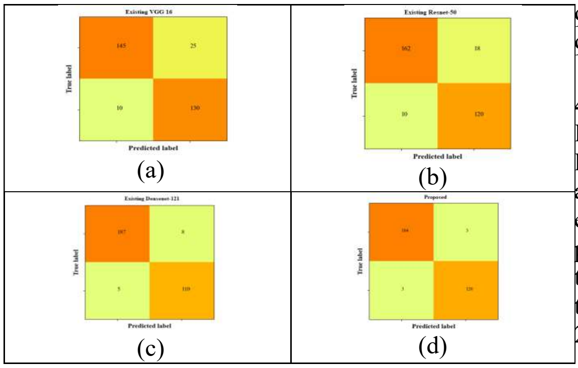
The most effective learning rate is determined in the experimental study through evaluation of the various learning rates (0.1, 0.05, 0.01, and 0.001) shown in Table 2. The learning rate of 0.1 achieved the maximum accuracy of 98.78%. Even though other learning rates reached close to this accuracy, the 0.1 learning rate has selected for further investigation because of its superior performance.

Table 2. Performance Measures for the Proposed Model with ChangingLearning Rate

Proposed Fine-tuned Inception-ResNet V2				
Metrics	**Learning Rate = 0.001**	**Learning Rate = 0.01**	**Learning Rate = 0.05**	**Learning Rate = 0.1**
Accuracy	95.68	96.44	97.35	98.78
Precision	95.04	96.75	97.75	98.87
Recall	95.46	96.38	97.72	98.69
F1 score	95.35	96.22	97.49	98.86
Specificity	95.58	96.49	96.44	98.68

4.3. Comparative analysis

The performance of the suggested system is compared to that of other models, including VGG 16, Resnet-50, and Densenet-121, in this section. In pest audio analysis, the main emphasis is on important metrics such as accuracy, specificity, precision, recall, and the F1-score. The research examines classification accuracy by comparing true positive rates to false positive rates using confusion matrices.



(a) (b)

(c) (d)

Figure 9:Confusion matrix (a) Existing VGG 16 (b) Existing Resnet-50 (c) Existing Densenet-121 (d) Proposed

Figure 9 shows confusion matrices comparing the performance of VGG-16, ResNet-50, DenseNet-121, and the proposed model classification approach. In Figure (a), VGG-16 accurately identified 145 and 130 examples in each class, with 25 and 10 misclassifications, respectively. Figure (b) shows ResNet-50, which outperformed VGG-16 with 162 and 120 correct classifications, respectively, while reducing misclassifications to 18 and 10. Figure (c) shows that DenseNet-121 performed much better, with 187 and 110 correct predictions and low misclassifications of 8 and 5. Finally, the proposed method in Figure (d) produced the greatest results, with 184 and 120 correct classifications and only three misclassifications in each class, demonstrating greater accuracy and reliability.

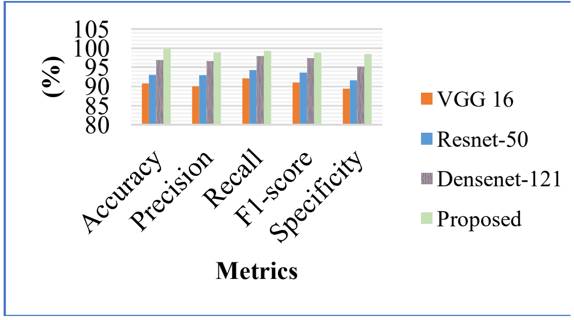


Figure 10: Performance analysis of various techniques in terms of accuracy, specificity, precision, recall, and the F1-score

Figure 10 compares the efficacy of VGG-16, ResNet-50, DenseNet-121, and the proposed technique using several evaluation metrics. VGG-16 had an accuracy of 90.8%, a precision of 90%, a recall of 92.13%, an F1-score of 91.05%, and a specificity of 89.43. ResNet-50 performed better with an accuracy of 93.08%, precision of 92.96%, recall of 94.29%, F1-score of 93.62%, and specificity of 91.67%. DenseNet-121 outperformed both, with 96.8% accuracy, 96.62% precision, 97.95% recall, 97.28% F1-score, and 95.19% specificity. The proposed strategy outperformed all existing models, with the highest accuracy of 99.89%, precision of 98.85%, recall of 99.36%, F1-score of 98.76%, and specificity of 98.45%, exhibiting better effectiveness.

4.4. Comparison of published works

In this comparison, the proposed fine-tuned Inception-ResNet-V2 model has the best accuracy, specificity, and F-score, suggesting higher overall performance. It excels at both accurate classification and reducing false positives. Certainly, we may make a comparison using the references [13], [14], [16], [17], and [19], as well as the proposed method. They were discussed in Section 2.

Table 3. Published work Comparison

Ref	Technique	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	Specificity (%)
[13]	InceptionResNet-V2	97.18	97	99	98	-
[14]	multi-layer convolutional neural network	84.51	-	-	-	-
[16]	VGG-13	91.93	-	94	96	-
[17]	MF-MDLNet	99.82	97.86	99.14	-	-
[19]	SVM	96.41	-	-	-	-
Proposed Fine-tuned Inception-ResNet-V2		99.89	98.85	99.36	98.76	98.45

The performance of several published methods and the proposed improved Inception-ResNet-V2 model is contrasted in Table 3 using a variety of evaluation metrics. The accuracy, precision, recall, and F-score of InceptionResNet-V2 were 97.18%, 97%, and 99%, respectively. The multi-layer convolutional neural network had an accuracy of 84.51%, while the VGG-13's F-score, accuracy, and recall were 96%, 91.93%, and 94%, respectively. The performance of MF-MDLNet was superior to the others, with an accuracy of 99.82%, precision of 97.86%, and recall of 99.14%. In the meantime, 96.41% accuracy had been achieved by the SVM approach. By attaining the maximum accuracy of 99.89%, precision of 98.85%, recall of 99.36%, F-score of 98.76%, and specificity of 98.45%, the suggested fine-tuned Inception-ResNet-V2 performed better than any other comparable method. This proves the models efficacy when compared to existing procedures.

5. Conclusion

Since insect attacks damage a significant amount of the yield and lower its quality, farmers find it challenging to identify pests in the fields where their crops are cultivated. The drawback of traditional pest detection techniques is that they depend on the knowledge of experts who have received substantial training in the applicable field in order to accurately identify pests based on their outward characteristics. The unintended consequence of employing pesticides to get rid of unwanted pests is that they can harm agriculture and nutrients from the environment. Finally, the proposed pest detection system shows great promise in tackling the issues of pest detection in large agricultural fields. Using deep learning algorithms and precise sound analysis, the system effectively overcomes obstacles such as unclear acoustic data, environmental noise interference, and the necessity for scalable, real-time detection. The use of audio preprocessing techniques such as denoising, segmentation, and feature extraction ensures high-quality input data, and then the hybrid Inception-ResNet model fine-tuned with the APO Algorithm detected the pest sound. This robust and efficient system, developed in Python, provides a scalable and dependable solution for accurate pest detection, helping to improve crop health and agricultural productivity. In the future, we plan to use cloud computing to detect pests early on a mobile app for specialists and farmers, reducing processing power requirements. Future research will focus on improving results by assigning several labels to sections in situations where various noises are heard at the same time, creating a multiclass classification problem.

Declarations

Funding

The authors declare that they have competing interests and funding

Conflict of Interests

On behalf of all authors, the corresponding author states that there is no conflict ofinterest.

Data Availability

Data sharing is not applicable to this article because of proprietary nature.

Code Availability

Code sharing is not applicable to this article because of proprietary nature.

Authors' contributions

All authors read and approved the final manuscript.

Clinical Trial Number

Not Applicable

Ethics, Consent to Participate, and Consent to Publish declarations

Not applicable

References

1. Weis, A.J., 2007. *The global food economy: The battle for the future of farming*. Zed Books.
2. Skendžić, S., Zovko, M., Živković, I.P., Lešić, V. and Lemić, D., 2021. The impact of climate change on agricultural insect pests. *Insects*, *12*(5), p.440.
3. Musa, S.F.P.D. and Basir, K.H., 2021. Smart farming: towards a sustainable agri-food system. *British Food Journal*, *123*(9), pp.3085-3099.
4. Talaviya, T., Shah, D., Patel, N., Yagnik, H. and Shah, M., 2020. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture*, *4*, pp.58-73.
5. Savary, S., Ficke, A., Aubertot, J.N. and Hollier, C., 2012. Crop losses due to diseases and their implications for global food production losses and food security. *Food security*, *4*(4), pp.519-537.
6. Johnson, G.I. and Hofman, P.J., 2009. Postharvest technology and quarantine treatments. *The mango: botany, production and uses*, pp.529-605.
7. Vorländer, M. and Bietz, H., 1994. Novel broad-band reciprocity technique for simultaneous free-field and diffuse-field microphone calibration. *ActaAcustica united with Acustica*, *80*(4), pp.365-377.
8. Singh, K.U., Kumar, A., Raja, L., Kumar, V., Singh kushwaha, A.K., Vashney, N. and Chhetri, M., 2022. An Artificial Neural Network-Based Pest Identification and Control in Smart Agriculture Using Wireless Sensor Networks. *Journal of Food Quality*, *2022*(1), p.5801206.
9. Presannakumar, K. and Mohamed, A., 2023. Deep learning based source identification of environmental audio signals using optimized convolutional neural networks. *Applied Soft Computing*, *143*, p.110423.
10. Rezaul, K.M., Jewel, M., Islam, M.S., Siddiquee, K.N.E.A., Barua, N., Rahman, M.A., Sulaiman, R.B., Shaikh, M.S.I., Hamim, M.A., Tanmoy, F.M. and Haque, A.U., 2024. Enhancing Audio Classification Through MFCC Feature Extraction and Data Augmentation with CNN and RNN Models. *International Journal of Advanced Computer Science and Applications*, *15*(7), pp.37-53.
11. Lucas, J.A., 2011. Advances in plant disease and pest management. *The Journal of Agricultural Science*, *149*(S1), pp.91-114.
12. Dhanaraj, R.K. and Ali, M.A., 2023. Deep Learning-Enabled Pest Detection System Using Sound Analytics in the Internet of Agricultural Things. *Engineering Proceedings*, *58*(1), p.123.
13. Karar, M.E., Reyad, O., Abdel-Aty, A.H., Owyed, S. and Hassan, M.F., 2021. Intelligent IoT-aided early sound detection of red palm weevils. *Cmc-Comput. Mater. Contin*, *69*, pp.4095-4111.
14. Balingbing, C.B., Kirchner, S., Siebald, H., Kaufmann, H.H., Gummert, M., Van Hung, N. and Hensel, O., 2024. Application of a multi-layer convolutional neural network model to classify major insect pests in stored rice detected by an acoustic device. *Computers and Electronics in Agriculture*, *225*, p.109297.
15. Ribeiro, A.P., da Silva, N.F.F., Mesquita, F.N., Araújo, P.D.C.S., Rosa, T.C. and Mesquita-Neto, J.N., 2021. Machine learning approach for automatic recognition of tomato-pollinating bees based on their buzzing-sounds. *PLoS computational biology*, *17*(9), p.e1009426.
16. Kim, J., Oh, J. and Heo, T.Y., 2021. Acoustic Scene Classification and Visualization of Beehive Sounds Using Machine Learning Algorithms and Grad-CAM. *Mathematical Problems in Engineering*, *2021*(1), p.5594498.
17. Ali, M.A., Sharma, A.K. and Dhanaraj, R.K., 2024. Multi-Features and Multi-Deep Learning Networks to identify, prevent and control pests in tremendous farm fields combining IoT and pests sound analysis.
18. Knebel, P., Appold, C., Guldner, A., Horbach, M., Juncker, Y., Müller, S. and Matheis, A., 2022. An artificial intelligence of things based method for early detection of bark beetle infested trees.
19. Escola, J.P.L., Guido, R.C., da Silva, I.N., Cardoso, A.M., Maccagnan, D.H.B. and Dezotti, A.K., 2020. Automated acoustic detection of a cicadid pest in coffee plantations. *Computers and Electronics in Agriculture*, *169*, p.105215.
20. USDA national agricultural library. U.S. department of agriculture (2023). https://data.nal.usda.gov/dataset/bug-bytes-sound-library-stored-product-insect-pest-sounds.
21. Trojovský, P. and Dehghani, M., 2022. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, *22*(3), p.855.