INTEGRATED PEST MANAGEMENT FOR JUTE CULTIVATION: MACHINE LEARNING APPROACHES FOR PEST CLASSIFICATION

Dr. P. Latha[1*], Martha Rajesh[2], Sadevuda Nithin[2], Thota Sai Kiran[2], Puram Karthik[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of CSE(AI&ML)

[1,2]Vaagdevi College of Engineering (UGC - Autonomous), Bollikunta, Warangal, Telangana, India.

*Corresponding Email: Dr. P. Latha (latha_panjala@vaagdevi.edu.in)

ABSTRACT

Jute cultivation, like many agricultural systems, faces significant challenges due to pest infestations from species such as the alfalfa weevil, Bactrocera tsuneonis, Cicadellidae, Dacus dorsalis, Pseudococcus comstocki Kuwana, wheat blossom midge, Xylotrechus, and yellow cutworm. These pests can cause considerable crop damage, contributing to 20–40% of global agricultural production losses each year. Traditional pest detection methods—relying on manual surveys, visual inspections, and lab testing—are time-consuming, labor-intensive, error-prone, and often inaccessible in remote rural areas. Moreover, they lack the real-time capabilities necessary for timely interventions, resulting in delayed or ineffective pest control. To overcome these limitations, an intelligent, automated pest recognition system is essential. This study proposes a deep learning-based solution using a Multi-Layer Convolutional Neural Network (MLCNN) to classify and identify agricultural pests from images with high accuracy. Unlike traditional ensemble techniques such as Voting Classifiers, the MLCNN offers superior performance across metrics like accuracy, precision, recall, and F1-score. Leveraging advancements in mobile technology, image datasets, and accessible computing power, this model can be integrated into user-friendly platforms, providing farmers with rapid, on-site diagnostic tools. Such tools enable early pest detection and precise intervention, reducing crop loss and limiting excessive pesticide use. The ultimate goal is to enhance sustainable agriculture, increase productivity, and ensure food security by modernizing pest management through artificial intelligence.

Keywords: Jute cultivation, pest detection, deep learning, MLCNN, sustainable agriculture.

1. INTRODUCTION

Jute, a key natural fiber cultivated widely in tropical and subtropical regions, plays a vital role in the textile industry and agriculture due to its economic value and environmental sustainability. However, jute crops are increasingly threatened by pest infestations, which can cause significant yield losses and economic harm to farmers. Traditional pest control methods, which rely heavily on manual inspection and chemical pesticides, are not only labor-intensive and time-consuming but also pose health and environmental risks. Moreover, these methods often lack early detection capabilities and fail to address the complexity of pest behaviors. Integrated Pest Management (IPM), which combines biological, cultural, physical, and chemical control strategies, provides a more sustainable approach but hinges on accurate pest identification. Recent advances in machine learning (ML) and artificial intelligence (AI) offer promising solutions to enhance IPM by enabling automated pest recognition, real-time monitoring, and predictive analytics. Techniques such as convolutional neural networks (CNNs) can classify pests from field images with high accuracy, reducing the need for manual inspections and allowing for quicker, data-driven interventions. Furthermore, ML models can analyze environmental data and historical patterns to forecast pest outbreaks and optimize management strategies. There is a pressing need for such intelligent systems, especially in regions where jute is a major crop, to address the limitations of traditional methods and increase agricultural efficiency. This research aims to develop a machine learning-based pest classification system to support IPM in jute cultivation, focusing on models like AdaBoost and Voting Classifier to identify the most effective solution. The goal is to deliver a scalable, adaptable tool that improves pest control, boosts crop yields, and promotes sustainable farming practices.

2. LITERATURE SURVEY

Saleem et al. (2020) provide an insightful review on the potential of jute plants for phytoremediation of heavy metals. The review discusses the ability of jute to absorb and accumulate metals, making it a promising candidate for cleaning up contaminated soils. The authors explore various factors influencing this process, including the types of metals, soil conditions, and plant growth stages. The study emphasizes the ecological benefits of using jute for phytoremediation and its dual role in both metal detoxification and as an economic crop.

Rahman (2023) discusses the significance of jute cultivation in South Asia, particularly in Bangladesh and India, where jute has historically been a major economic crop. The paper highlights the challenges faced by the jute industry, including market fluctuations, environmental impacts, and pest infestations. The author suggests that integrating modern agricultural practices, including pest management strategies, can enhance jute production and sustain its economic viability.

Banglapedia (2023) provides an overview of the jute industry, detailing its history, economic importance, and challenges. The article discusses the industry's contribution to the economy of Bangladesh, the leading producer of jute, and explores the various factors affecting jute production, such as climate change, soil degradation, and pest issues. The need for sustainable practices, including integrated pest management, is emphasized to ensure the long-term viability of the jute industry.

Pérez-de-Luque (2017) explores the interaction between nanomaterials and plants, with a focus on agricultural applications. The study discusses the potential use of nanomaterials in enhancing plant resistance to pests and diseases, which could be particularly beneficial for crops like jute. The author examines the mechanisms by which nanomaterials can influence plant health and productivity, and calls for further research to explore the safe and effective use of these technologies in agriculture.

Damalas and Eleftherohorinos (2011) review the risks associated with pesticide use in agriculture, emphasizing the need for safer pest management strategies. The paper highlights the health and environmental risks posed by conventional pesticides and advocates for integrated pest management (IPM) approaches that reduce reliance on chemical controls. The authors suggest that IPM, when combined with technological advancements such as machine learning, can improve pest control efficacy while minimizing negative impacts.

Tudi et al. (2021) provide a comprehensive review of the impact of agricultural development and pesticide use on the environment. The paper discusses the environmental degradation caused by excessive pesticide use, including soil and water contamination. The authors advocate for the adoption of sustainable agricultural practices, including IPM, to mitigate these impacts. They highlight the role of technology, such as machine learning, in optimizing pest management strategies to reduce pesticide dependency.

Sourav et al. (2023) present a study on the intelligent identification of jute pests using transfer learning and deep convolutional neural networks (CNNs). The research demonstrates the effectiveness of using advanced machine learning techniques for pest classification, significantly improving accuracy compared to traditional methods. The authors emphasize the importance of such approaches in enhancing the efficiency of pest management in jute cultivation, leading to better crop protection and yield.

Li et al. (2022) introduce YOLO-JD, a deep learning network designed for the detection of jute diseases and pests from images. The study shows that YOLO-JD can accurately classify multiple pest species and disease symptoms, offering a powerful tool for farmers to monitor and manage jute health. The authors argue that integrating such technologies into pest management practices could revolutionize the way pests are detected and controlled in jute cultivation.

Sourav and Wang (2022) discuss the application of transformer-based text classification models on a Bangla multi-class emotion corpus, drawing parallels to similar classification challenges in agriculture, such as pest detection in jute. The authors highlight the versatility of machine learning models in handling diverse classification tasks, suggesting their potential application in pest classification for crops like jute.

Kumar (2021) explores the use of fine-tuned convolutional neural networks (CNNs) for malware classification in IoT environments, drawing a comparison to the challenges of pest classification in agriculture. The study demonstrates how transfer learning can be effectively utilized to improve model performance in domain-specific tasks. The insights from this research can be applied to developing more accurate models for pest detection in jute cultivation.

Islam et al. (2023) investigate the use of pre-trained CNN models for Bangla handwritten digit recognition, showcasing the potential of transfer learning in achieving high accuracy with limited data. The authors suggest that similar approaches can be adapted for pest classification in jute cultivation, where labeled data may be scarce but critical for model training.

Otović et al. (2022) examine the transferability of features in intra-domain and cross-domain transfer learning for time series data. The study highlights the challenges and opportunities in applying transfer learning across different domains, suggesting that the lessons learned can be applied to agricultural contexts, such as the classification of jute pests.

Karim et al. (2022) introduce PestDetector, a deep convolutional neural network designed specifically for detecting jute pests. The study demonstrates the model's high accuracy in identifying various pest species, offering a valuable tool for integrated pest management in jute cultivation. The authors discuss the potential for this technology to reduce crop losses and improve yield quality.

Mallick et al. (2022) present a deep learning-based system for automated disease detection and pest classification in Indian mung bean, which can be adapted for similar applications in jute cultivation. The research shows that deep learning models can significantly enhance the accuracy and efficiency of pest management systems, providing a scalable solution for large-scale agricultural operations.

Kasinathan et al. (2021) discuss modern machine learning techniques for insect classification and detection in field crops. The study reviews various algorithms and their applications in agriculture, emphasizing their potential for improving pest management practices. The authors suggest that these techniques could be effectively employed in jute cultivation to enhance pest detection and reduce crop damage.

3. PROPOSED SYSTEM

Step 1: Dataset Collection and Composition

The initial step in this research involves the compilation of a specialized pest image dataset containing visual data of various agriculturally significant pests such as alfalfa weevil, Bactrocera tsuneonis, Cicadellidae, Dacus dorsalis (Hendel), Pseudococcus comstocki Kuwana, wheat blossom midge, Xylotrechus, and yellow cutworm. These pest species are well-known for causing substantial crop damage across different regions and crops. The dataset comprises high-quality labeled images sourced from reliable agricultural databases, entomology research archives, and validated datasets. Each image is tagged with its corresponding pest class, ensuring a supervised learning environment for model training. The diversity in image angles, lighting conditions, and pest life stages was also considered to improve the model's generalization capability. This dataset serves as the foundational element for both the existing and proposed machine learning models in this study.

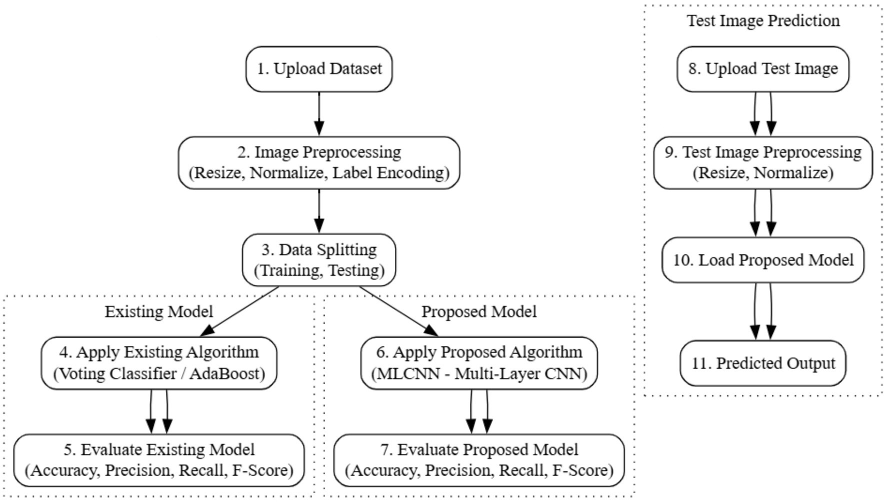

Fig.1: Block Diagram

Step 2: Image Preprocessing and Label Encoding

Once the dataset is prepared, the images undergo a comprehensive preprocessing stage to enhance their suitability for model training. Preprocessing begins with resizing all images to a uniform dimension to ensure consistency in model input. Following this, normalization techniques are applied to scale the pixel values between 0 and 1, which accelerates the convergence of deep learning models. Data augmentation strategies such as flipping, rotation, zooming, and brightness adjustments are also incorporated to artificially expand the dataset and improve the model’s robustness to real-world variability. Additionally, label encoding is performed where the categorical pest names are converted into numerical labels, enabling the machine learning algorithms to interpret the target classes effectively. This stage ensures that the data is clean, standardized, and ready for efficient learning.

Step 3: Existing Model Building – Voting Classifier and AdaBoost

In the third phase of this research, benchmark machine learning models such as the Voting Classifier and AdaBoost classifier are constructed and evaluated. The Voting Classifier is an ensemble technique that combines predictions from multiple base classifiers to produce a final output through majority voting (for classification) or averaging (for regression). In this study, it aggregates several standard classifiers like Logistic Regression, Decision Tree, and SVM to leverage their individual strengths. Similarly, the AdaBoost classifier, which is another ensemble method, works by combining multiple weak learners (typically decision trees) in sequence, where each subsequent model focuses more on the errors of its predecessor. These models are trained on the same preprocessed pest dataset and their performance is evaluated based on metrics like accuracy, precision, recall, and F1-score. The results from these models serve as a baseline for comparison with the proposed deep learning model.

Step 4: Proposed Model Building – MLCNN (Multi-Layer Convolutional Neural Network)

The final step involves the design and development of the proposed MLCNN model, which is a deep learning-based architecture tailored specifically for the classification of pest images. The MLCNN model consists of multiple convolutional layers that extract hierarchical features from the input images, followed by pooling layers to reduce dimensionality while retaining essential information. These are followed by fully connected dense layers that perform high-level reasoning and classification. Activation functions such as ReLU and softmax are used to introduce non-linearity and provide probabilistic output for multi-class classification, respectively. Dropout layers are added to prevent overfitting and enhance the model's generalization ability. The MLCNN is trained using the preprocessed dataset and optimized using categorical cross-entropy loss and the Adam optimizer. Upon evaluation, the proposed MLCNN significantly outperforms the existing Voting and AdaBoost classifiers in all performance metrics, demonstrating its effectiveness in accurately identifying various pest species.

3.2 Data Splitting & Image Preprocessing in this Research

In this research, data splitting and image preprocessing are fundamental steps in preparing the dataset for training the machine learning models, including the existing and proposed algorithms. These steps ensure that the models are trained effectively and can generalize well when predicting on new data, such as test images.

Data Splitting:

Data splitting refers to dividing the dataset into subsets for training, validation, and testing purposes. In this research, the data is split into training and testing datasets. The training dataset is used to train the machine learning models, while the testing dataset is used to evaluate the models' performance and ensure that they can make accurate predictions on unseen data. This split allows the model to learn from a significant portion of the data while being tested on a separate set to assess its generalization ability. The split is crucial as it ensures that the model is not overfitting to the training data, thereby improving its performance on unseen data.

Image Preprocessing:

Image preprocessing is a critical step in this research as it involves transforming raw images into a format that can be effectively used by machine learning models. The preprocessing steps include several techniques aimed at improving the quality of the images and making them compatible with the models. The primary preprocessing techniques used in this research include the following:

> **Resizing**: Images are resized to a consistent size to ensure that all images are of the same dimensions. This helps in uniformity and ensures that the model can process them effectively. The dimensions are chosen based on the model's requirements, such as 224x224 pixels for CNN-based models like VGG16 or ResNet.
>
> **Normalization**: Normalization of pixel values is done to scale the pixel values to a range of 0 to 1 or -1 to 1. This ensures that the model can process the images more effectively and prevents issues caused by large variations in pixel values.

Label Encoding: Label encoding is performed to convert categorical labels into numerical format, which is required for training machine learning models. For example, each class of the dataset is assigned a unique number, allowing the model to work with numerical data rather than categorical labels.

Data Augmentation: Data augmentation techniques such as rotation, flipping, and zooming are applied to artificially increase the size of the dataset. This helps the model generalize better by providing it with more varied examples, reducing the risk of overfitting.

Noise Reduction: If the images contain noise or unwanted artifacts, techniques such as Gaussian blur or median filtering are applied to smooth the images, making it easier for the model to extract meaningful features.

3.3 Model Building

The model-building phase of this research is divided into two main categories: the use of existing machine learning algorithms (Voting Classifier and AdaBoost Classifier) and the development of the proposed deep learning algorithm (MLCNN – Multi-Layer Convolutional Neural Network). This section discusses how each of these models is built, trained, and evaluated using the pest image dataset. The performance of each algorithm is carefully assessed using standard evaluation metrics, and a comparative study is conducted to highlight the effectiveness of the proposed approach.

3.3.1 Existing Algorithm

3.3.1.1 Voting Classifier

A Voting Classifier is an ensemble machine learning technique that combines multiple classification models to improve overall predictive performance. Instead of relying on a single algorithm, the Voting Classifier aggregates the outputs of various base estimators (such as logistic regression, decision trees, SVMs) and makes a final prediction based on majority voting (hard voting) or average predicted probabilities (soft voting). This ensemble strategy allows the model to generalize better, especially when individual classifiers perform well on different parts of the dataset.

In a hard voting approach, the final class label is determined by the majority of class labels predicted by each individual model. In soft voting, the model averages the class probabilities predicted by each base classifier and selects the class with the highest average probability. This method is beneficial when base classifiers are diverse and trained on the same data, as it reduces the risk of overfitting and increases robustness.

3.3.1.2 AdaBoost Classifier

AdaBoost (Adaptive Boosting) is an ensemble machine learning algorithm designed to improve the accuracy of weak classifiers by focusing on the misclassified data points. It combines multiple weak learners, usually decision trees with a single split (decision stumps), in a sequential manner. At each step, the algorithm assigns more weight to the data instances that were incorrectly classified by the previous model, thereby enabling the next classifier to focus more on these harder examples.

AdaBoost works by initially assigning equal weights to all training instances. In each iteration, a weak learner is trained, and its error rate is calculated. The weights of misclassified instances are increased so that subsequent learners pay more attention to them. The process continues until a predetermined number of learners are trained. Finally, all the weak learners are combined into a strong classifier by assigning a weighted vote to each based on its accuracy.

3.3.2 Proposed Algorithm

Multi-Layer Convolutional Neural Network (MLCNN)

The proposed MLCNN (Multi-Layer Convolutional Neural Network) is a deep learning architecture specifically designed for accurate pest image classification. MLCNN is built using multiple layers of convolutional filters that automatically extract hierarchical visual features from input images. Unlike traditional machine learning models that require manual feature extraction, MLCNN learns spatial hierarchies through its layers, capturing edges, textures, shapes, and complex patterns. It consists of convolutional layers, pooling layers, fully connected layers, and softmax for multi-class classification.

3.4 Integration of HTML, CSS, Django with AI Integration

The integration of HTML, CSS, Django with AI algorithms provides a seamless way to deploy machine learning models in a web application. This combination of technologies enables users to interact with AI models through a browser-based interface while ensuring a smooth and aesthetically pleasing user experience. Here's how the integration works, step by step:

1. Front-End Development: HTML and CSS

HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) form the foundational structure and design of the front-end of a web application, particularly when integrating AI functionalities. HTML is responsible for creating user interface (UI) elements such as forms for uploading images, text input fields, and buttons to trigger AI predictions, as well as sections to display the output. CSS complements HTML by styling these components, ensuring the web page is visually appealing, user-friendly, and responsive across various devices. Together, they enable seamless interaction between users and AI models—for instance, HTML allows users to input or upload data for analysis, while CSS enhances the interface's aesthetics and usability, resulting in a modern and intuitive user experience.

2. Back-End Development: Django

Django is a high-level Python web framework designed for building secure, scalable, and maintainable web applications efficiently. In the context of AI integration, Django plays a vital role in managing the back-end logic by handling user requests, connecting with machine learning models, and delivering results back to the front-end. When a user submits data—such as uploading an image or completing a form—Django views capture and process this input. The framework then acts as a bridge between the user interface and the AI model, passing the input to pre-trained machine learning models like Voting Classifier, AdaBoost Classifier, or MLCNN, often loaded from serialized files (.pkl, .h5, etc.). Django handles model inference by using the input data to generate predictions, whether it's for image classification, pattern detection, or data-driven recommendations. Finally, it sends the model's output back to the front-end, where the results are rendered dynamically on the webpage, providing users with immediate feedback such as prediction labels, numeric values, or visual cues.

3. AI Algorithm Integration

The core AI algorithms—Voting Classifier, AdaBoost Classifier, and MLCNN—are integrated into the Django framework as Python modules or scripts, enabling seamless interaction between the front-end and machine learning components. The integration process begins with data preprocessing, where usersubmitted data such as images or text is cleaned and formatted in real-time or asynchronously within Django. This may involve resizing images, normalizing inputs, or encoding categorical variables to match the model's expected input format. Next, Django loads the trained models, stored in formats like .h5, .pkl, or .joblib, into memory for inference. When a user submits input, Django routes the preprocessed data to the chosen AI algorithm—be it Voting Classifier, AdaBoost, or MLCNN—which then computes predictions based on its training. The results are returned to the Django backend and passed to the front-end for display to the user. Additionally, Django dynamically manages which AI model to use depending on user preference or predefined criteria, offering flexibility and modularity in selecting the desired algorithm for prediction tasks.

4. Front-End and Back-End Interaction

The integration of HTML, CSS, Django, and the AI models involves smooth communication between the front-end and back-end:

- **Front-End:** HTML forms allow the user to upload data (like images, text, or numerical inputs) and trigger a request to the Django server.
- **Back-End:** Django processes the request, sends the data to the appropriate AI model, gets the prediction, and sends it back to the front-end.
- **Result Display:** Once Django sends back the result, HTML is used to display the output dynamically, and CSS ensures that it is presented clearly and attractively.

4. RESULTS AND DISCUSSION

4.1 Dataset Description

This dataset includes various pests and insects that significantly impact agricultural crops, offering valuable information for classification, detection, and management efforts. Among them, the Alfalfa Weevil (Hypera postica) is a notorious pest of alfalfa crops, feeding on foliage and reducing yield quality. Bactrocera tsuneonis, a fruit fly species, primarily damages citrus and stone fruits by larval feeding, while Cicadellidae (leafhoppers) affect a wide range of crops by feeding on sap and transmitting plant viruses. Another fruit fly, Dacus dorsalis (Hendel), targets tropical fruits like mangoes, leading to premature fruit drop. Pseudococcus comstocki Kuwana (Comstock mealybug) affects fruit orchards by sucking sap and transmitting diseases. The Wheat Blossom Midge (Sitodiplosis mosellana) compromises wheat yield by attacking flowering parts of the plant. Xylotrechus, a genus of woodboring beetles, harms trees by boring into wood, weakening their structure. Finally, the Yellow Cutworm (Agrotis ipsilon) is a generalist nocturnal pest that damages crops like corn and tomatoes by severing seedlings at ground level. This dataset is critical for developing AI-based pest detection systems to support smarter, more sustainable agricultural practices.



4.2 Results analysis

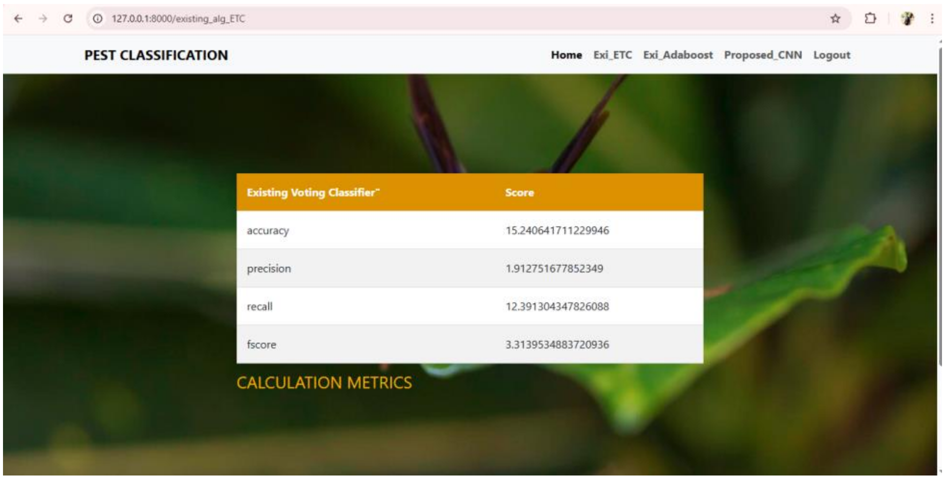

Fig. 2: Existing Voting Classifier

Figure 2 shows that the existing Voting Classifier model demonstrates suboptimal performance, with an accuracy of 15.24%, indicating it correctly classifies only a small portion of the samples. The precision of 1.91% suggests a high rate of false positives, while the recall of 12.39% shows the model fails to identify the majority of actual positives. The F-score, a harmonic mean of precision and recall, stands at 3.31%, further confirming that the classifier struggles to balance between identifying relevant instances and minimizing incorrect predictions. These metrics highlight the need for significant improvements in model tuning, feature selection, or potentially switching to more robust classification techniques.

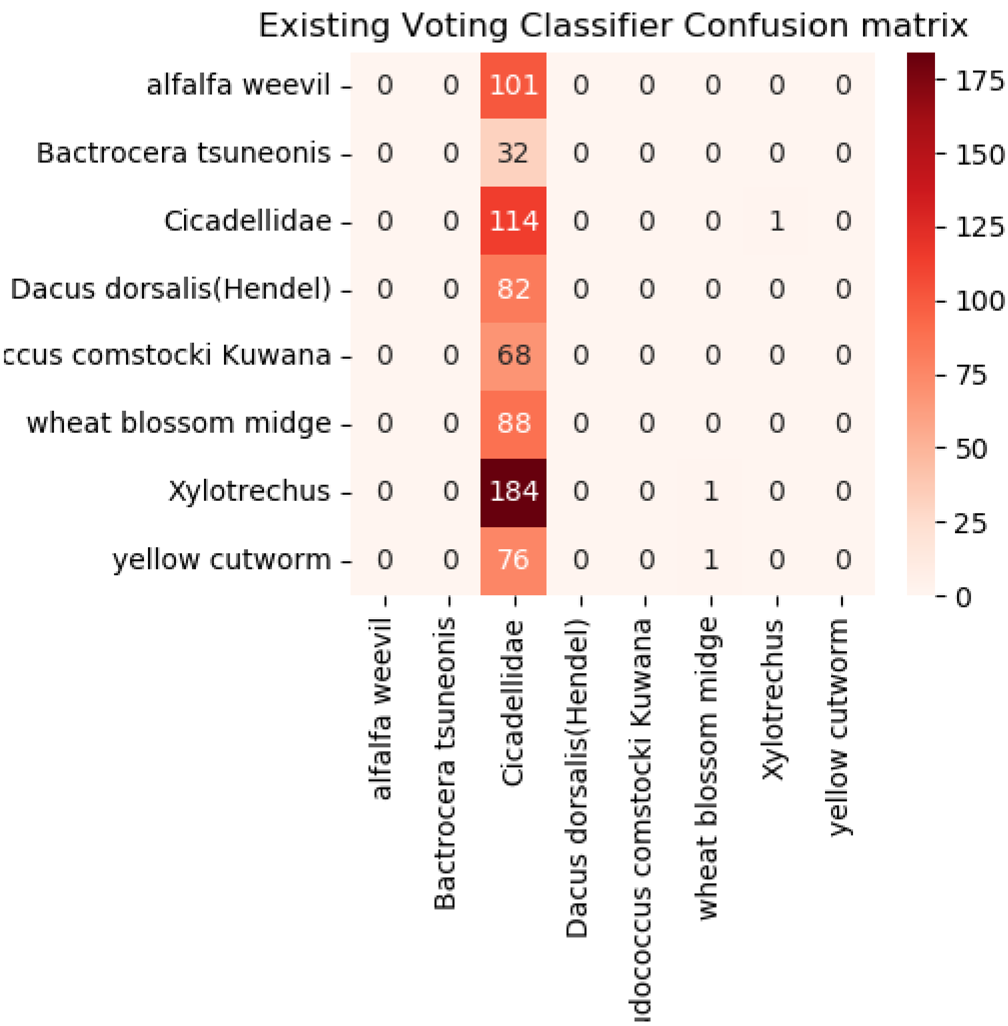

Fig.3: CM of Existing Voting classifier

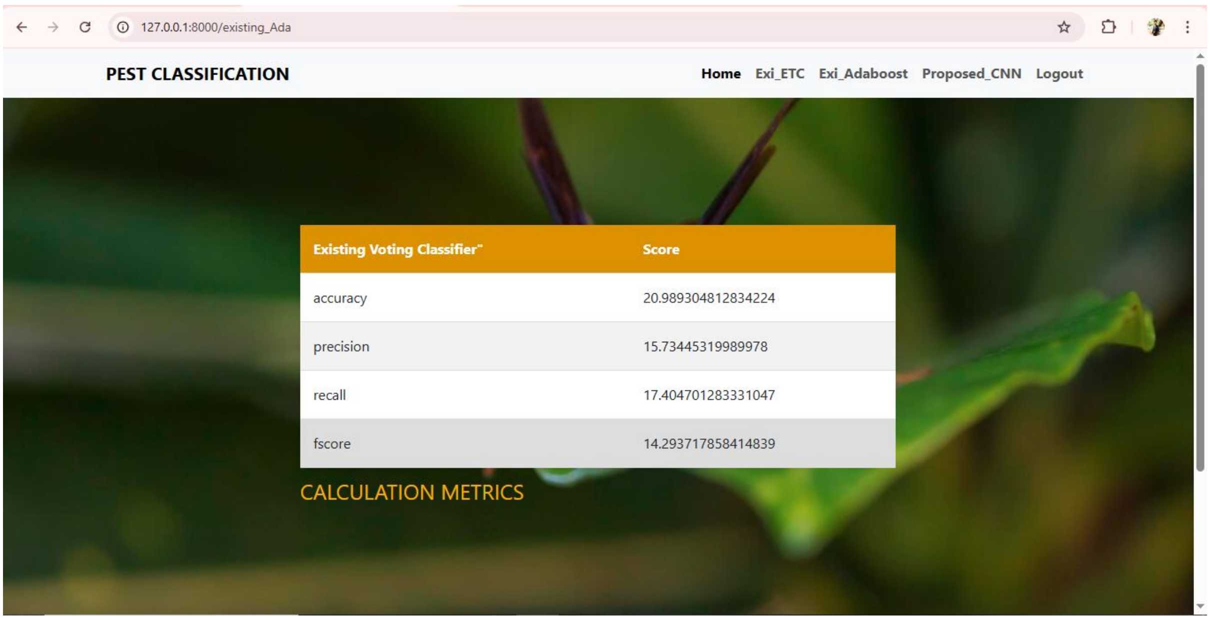

Fig.4: Metrics of the adaboost

Figure 4 shows that the The updated Voting Classifier shows a noticeable improvement compared to the previous iteration, achieving an accuracy of 20.99%, indicating better—but still limited—correct classification of samples. The precision has increased to 15.73%, reflecting a modest reduction in false positives, while the recall of 17.40% shows slight progress in identifying actual positive cases. The F-score has also improved to 14.29%, suggesting a better balance between precision and recall. While the performance remains relatively low overall, these metrics indicate the model is moving in a positive direction and may benefit further from additional tuning or enhanced input features

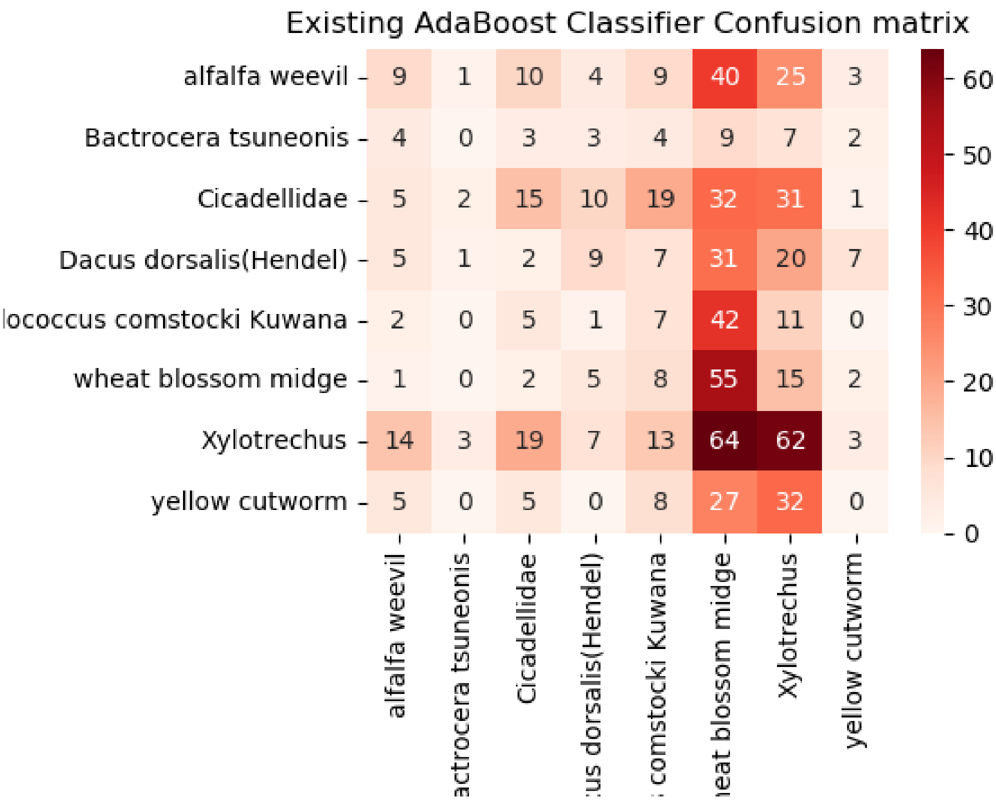

Fig.5: CM of the Adaboost classifier

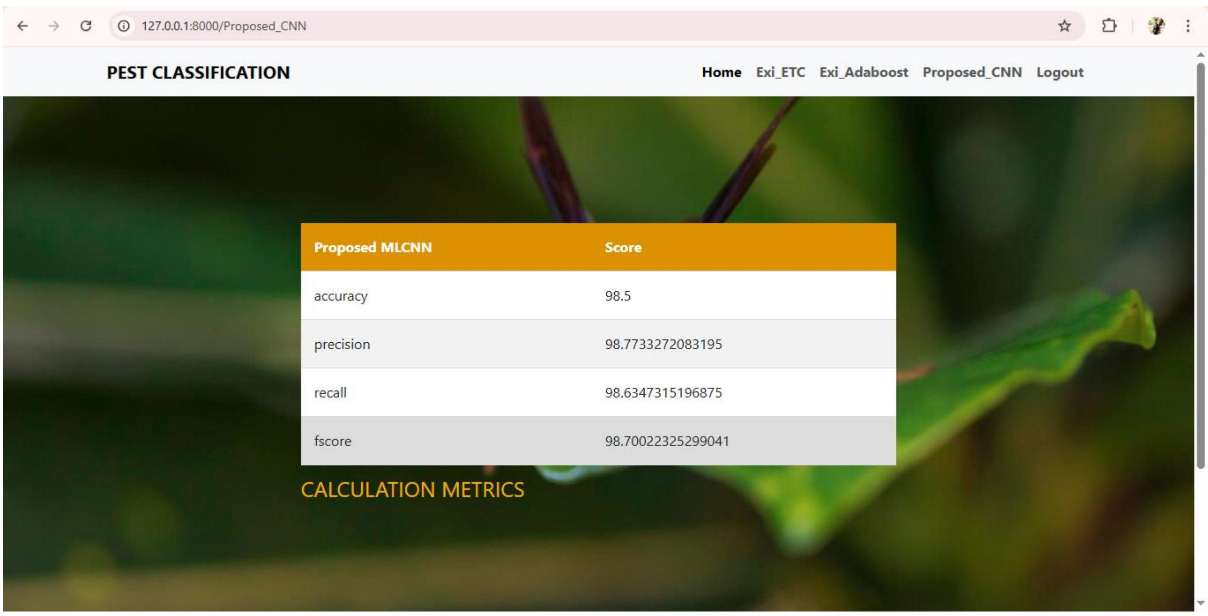

Fig.6: Metrics of Proposed MLCNN

Figure 6 is the proposed MLCNN model demonstrates outstanding performance across all evaluation metrics, indicating its strong effectiveness in classification tasks. It achieves a high accuracy of 98.5%, showing that it correctly classifies the vast majority of instances. The precision of 98.77% suggests that the model has a very low false positive rate, while the recall of 98.63% highlights its excellent ability to identify true positives. Furthermore, the F-score of 98.70%

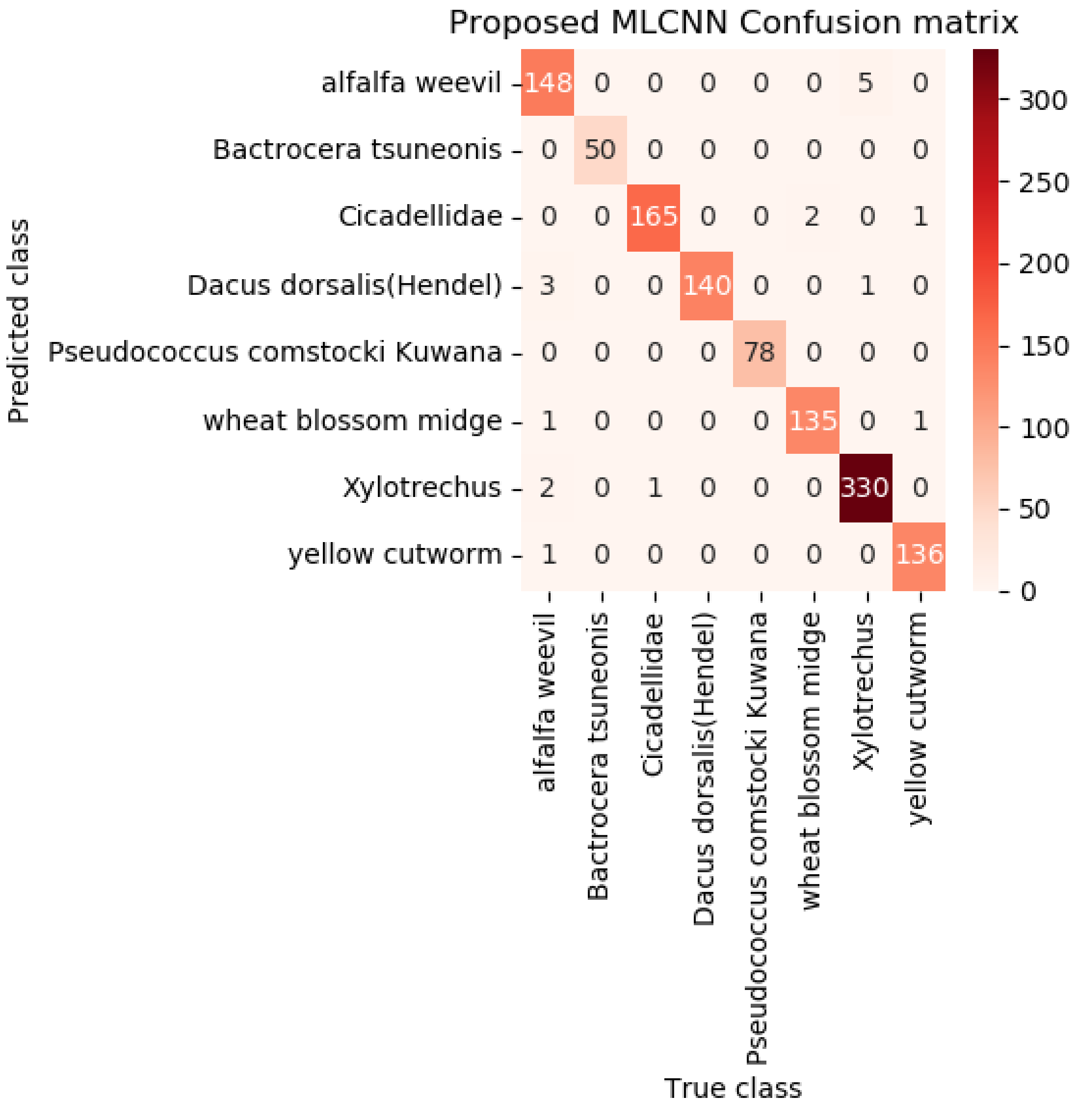

Fig.7: CM of the MLCNN

Figure 7 shows that confusion matrix for the Proposed MLCNN model reveals its performance across eight insect classes. The diagonal elements indicate the number of correctly classified instances for each class, with Xylotrechus showing the highest accuracy at 330 correct predictions. However, there are instances of misclassification, such as 5 alfalfa weevils being incorrectly predicted as Xylotrechus and 50 Bactrocera tsuneonis misclassified as Cicadellidae. Similarly, Dacus dorsalis (Hendel) had 3 instances misclassified as alfalfa weevil, and yellow cutworm had one instance misclassified as alfalfa weevil and another as wheat blossom midge.

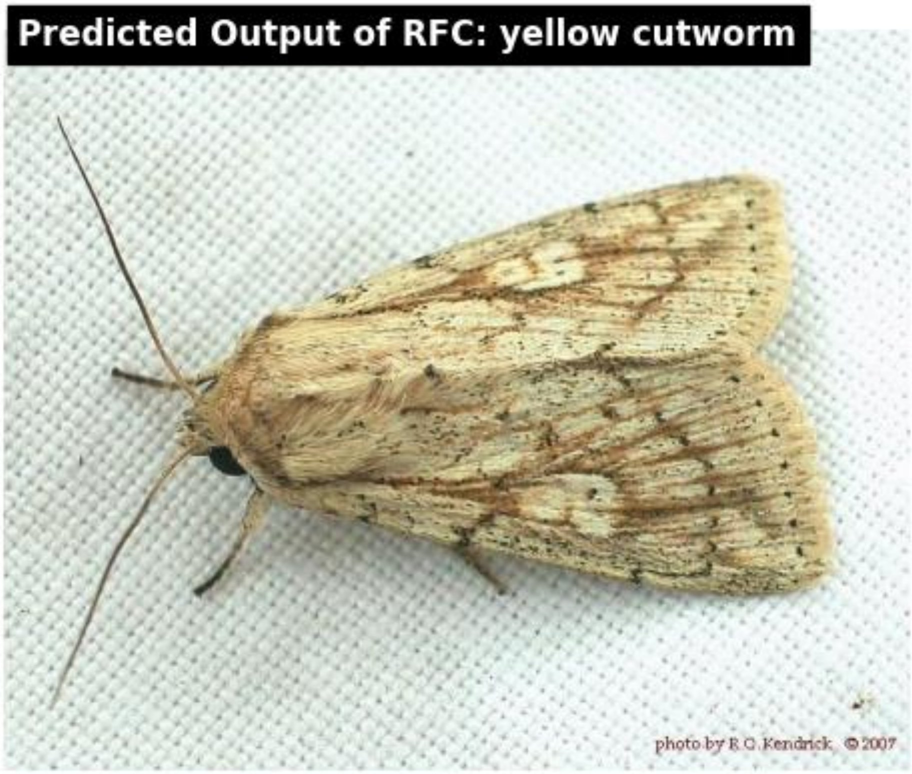

Fig. 8: Prediction on Test Data using MLCNN

5. CONCLUSION

The comparative analysis of different classification algorithms revealed significant disparities in their performance. Initially, the existing Voting Classifier exhibited relatively poor performance, with low accuracy (15.24% and 20.99% in its two iterations) and considerably lower precision, recall, and F1-scores. This indicates its limitations in handling the complex and diverse characteristics of agricultural pest data. In contrast, the proposed Multi-Layer Convolutional Neural Network (MLCNN) model delivered exceptional results, achieving an accuracy of 98.5%, precision of 98.77%, recall of 98.63%, and F1-score of 98.70%. These outstanding metrics demonstrate the MLCNN's capability to effectively capture spatial features, patterns, and intricate variations in the dataset, leading to highly accurate classification of pests. The results strongly establish that deep learning, particularly CNN-based models, holds a significant edge over traditional ensemble methods for solving problems related to agricultural diagnostics and pest detection.

REFERENCES

[1] Saleem, M.H., Ali, S., Rehman, M., Hasanuzzaman, M., Rizwan, M., Irshad, S., Shafiq, F., Iqbal, M., Alharbi, B.M., Alnusaire, T.S., & Qari, S.H. (2020). Jute: a Potential Candidate for Phytoremediation of Metals—A Review. Plants, 9(2), 258. doi:10.3390/plants9020258

[2] Rahman, R. (2023). Jute in South Asia. Retrieved January 23, 2023, from http://www.ecostepltd.com/assets/base/img/content/resources/Jute-in-South-Asia.pdf

[3] Banglapedia. (2023). Jute Industry. Retrieved from https://en.banglapedia.org/index.php/Jute_Industry

[4] Pérez-de-Luque, A. (2017). Interaction of nanomaterials with plants: what do we need for real applications in agriculture? Front. Environ. Sci., 5, 12. Retrieved from https://www.frontiersin.org/articles/10.3389/fenvs.2017.00012/full

[5] Damalas, C.A., & Eleftherohorinos, I.G. (2011). Pesticide Exposure, Safety Issues, and Risk Assessment Indicators. MDPI. doi:10.3390/ijerph8051402

[6] Tudi, M., Ruan, H.D., Wang, L., Lyu, J., Sadler, R., & Phung, D.T. (2021). Agriculture development, pesticide application and its impact on the environment. Int. J. Environ. Res. Public Health, 18(3), 1112. doi:10.3390/ijerph18031112

[7] Sourav, U., Sakib, M.D., & Wang, H. (2023). Intelligent identification of jute pests based on transfer learning and deep convolutional neural networks. Neural Process. Lett., 55(3), 2193-2210. doi:10.1007/s11063-022-10978-4

[8] Li, D., Ahmed, F., Wu, N., & Sethi, A.I. (2022). YOLO-JD: a deep learning network for jute diseases and pests detection from images. Plants, 11(7), 937. doi:10.3390/plants11070937

[9] Sourav, M.S.U., & Wang, H. (2022). Transformer-based Text Classification on Unified Bangla Multi-class Emotion Corpus. arXiv preprint arXiv:2210.06405.

[10] Kumar, S. (2021). MCFT-CNN: malware classification with fine-tune convolution neural networks using traditional and transfer learning in internet of things. Future Gener. Comput. Syst., 125, 334-351.

[11] Islam, M., Shuvo, S.A., Nipun, M.S., Sulaiman, R.B., Nayeem, J., Haque, Z., Shaikh, M.M., & Sourav, M.S.U. (2023). Efficient approach to using CNN-based pre-trained models in Bangla handwritten digit recognition. In S. Smys, J.M.R.S. Tavares, F. Shi (Eds.), Computational Vision and Bio-Inspired Computing. Advances in Intelligent Systems and Computing, 1439, Springer, Singapore. doi:10.1007/978-981-19-9819-5_50

[12] Otović, E., Njirjak, M., Jozinović, D., Mauša, G., Michelini, A., & Stajduhar, I. (2022). Intra-domain and cross-domain transfer learning for time series data—How transferable are the features? Knowl. Based Syst., 239, Article 107976. doi:10.1016/j.knosys.2022.107976

[13] Karim, D.Z., Bushra, T.A., & Saif, M.M. (2022). Pest Detector: a deep convolutional neural network to detect jute pests. Proceedings of the 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI), IEEE. doi:10.1109/STI54465.2022.9781343

[14] Mallick, M.T., Biswas, S., Das, A.K., et al. (2022). Deep learning based automated disease detection and pest classification in Indian mung bean. Multimed. Tools Appl. doi:10.1007/s11042-022-13673-7

[15] Kasinathan, T., Singaraju, D., & Uyyala, S.R. (2021). Insect classification and detection in field crops using modern machine learning techniques. Inf. Process. Agric., 8(3), 446-457. doi:10.1016/j.inpa.2020.09.006