Intelligent Pest Identification for Precision Agriculture using Deep Learning

Anika Bhat
Moreau Catholic High School
Hayward, CA, USA
email: anika.bhat1022@gmail.com

Atul Dubey
AIClub Research Institute
Mountain View, CA, USA
email: atul.dubey@aiclub.world

***Abstract*—Global wheat production suffers annual losses of ~157 million metric tons due to pests, causing food insecurity and economic damages exceeding $70 billion. Traditional detection methods, such as manual inspections, are slow, labor-intensive, and often fail to identify early infestations, forcing farmers to use excessive pesticides. To address this, an image analysis system driven by Artificial Intelligence (AI) was developed, trained on pest imagery, and deployed via an accessible web application, enabling early detection to prevent crop losses. The IP102 dataset with wheat pest categories only was used to train the Machine Learning (ML) models. Two approaches were used to build an ML model that can detect wheat pests. The first employed transfer learning on MobileNetV2, and it gave the best validation accuracy of 55.32%. The second used ConvNeXtLarge to extract robust image features of 9 categories of wheat pests. Four ML algorithms, K-Nearest Neighbors (KNN), Random Forest, Multi-Layer Perceptron (MLP), and XGBoost, were trained and evaluated. The MLP model, optimized with 30 epochs and a learning rate of 0.001, achieved the highest validation accuracy of ~79% and test accuracy of ~75%. The system was integrated into a user-friendly web application, paired with a low-cost, WiFi-enabled camera device for field image capture. This system facilitated early-stage pest detection, enabling farmers to remotely monitor and take preventative measures promptly. This AI-driven model can contribute to efficient, sustainable, and precise agricultural practices and bolster global food security.**

Keywords-Wheat pest identification; Deep Learning; Machine Learning; MobileNetV2; ConvNeXtLarge; Internet of Things.

I. Introduction

Wheat is the second most produced grain in the world, 785 tons in 2023-24 [1], and the US is the 4th largest producer. Wheat provides 21% of the global food requirement, but pests destroy ~157 million tons of grain/yr, causing food insecurity [1][2]. 20–40% of global crop loss/yr due to pests: $70 billion loss/yr [3].

The most common wheat pests are Aphids, Green bugs, Ceredonta Denticonis, Spider mites, Penthaleus Major, Wheat Blossom midges, and Wheat sawflies. In current pest monitoring methods, farmers rely on reactive and delayed pest detection, leading to irreversible crop damage and overuse of pesticides. They rely on visual monitoring, which is time-consuming and labor-intensive. Some of them use satellite/drone imagery, but it provides low-resolution images and delayed information, which can lead to missed detections and hinder early intervention efforts.

Current pest monitoring challenges include variable life cycle, nighttime activity, being hidden within plant canopies, and the need for precise timing to catch peak activity. The overuse of broad-spectrum pesticides increases pest resistance.

AI-powered agricultural image analysis is crucial in modern agriculture [4]. The AI-driven image analysis in wheat pest control can enable continuous monitoring and early detection of pests to prevent yield loss. The Deep Learning (DL) based classification and detection techniques could be very effective in identifying the type of pest from the images. Specifically, different types of Convolutional Neural Network (CNN) architectures can identify image features easily and can be effective in pest classification. This can enable rapid, scalable, and precise pest identification. Figuring out a way to put a pest detection model in a device and be able to deploy it in the field will increase the effectiveness of early detection.

In this study, we used 2 different approaches to build classifiers to identify wheat pests. In the first approach, we performed training on transfer learning on MobileNetV2 [5] by removing the top layer and adding some custom neural network layers. In the second approach, we used ConvNeXtLarge [6] model to extract features from the images and used the features and dataset to build various classifiers using techniques like KNN, Random Forest, XGBoost, and MLP. The best-performing model was used with a web app to classify different types of pests. The web app gets input from a device that captures images of wheat pests in the field. The detection information is displayed in a web application with suggestions for remedy.

The rest of the paper is laid out as follows: Section II discusses the existing research done in the pest identification and control domain. Section III elaborates on the steps that were followed to perform the study and build the required ML model and device with the web application. Section IV contains the results from various experiments performed. In Section V, we present the behavior of the models and the results obtained from different experiments. Finally, Section VI concludes the research and mentions future work.

II. Related Work

Multiple studies have been done to identify wheat crop diseases. The study by Mundada and Gohokar [7] focused on the detection and classification of pests in greenhouses using traditional image processing techniques. Images of infected leaves were captured, and properties like entropy, mean, standard deviation, contrast, energy, correlation, and eccentricity were extracted. These features were then used to train a Support Vector Machine (SVM) for classification. They developed a software prototype system for early pest detection in greenhouses, achieving a training accuracy of 100% with the

SVM classifier. By utilizing featurization techniques, images from the existing dataset were filtered to form a new dataset, enhancing the ML model's effectiveness.

Shi et al. [8] conducted research on pest and disease detection in winter wheat using spectral indices and kernel discriminant analysis. Hyperspectral reflectance datasets at both leaf and canopy levels were utilized, involving fourteen Spectral Vegetation Indices (SVIs) as input. The approach showed better performance over conventional linear methods, achieving classification accuracies between 76% and 95% for various infestations.

Haider et al. [9] explored a generic approach to wheat disease classification, incorporating field surveys, expert opinion, and crowd-sourced data. Using symptoms as predictor variables, a decision tree model was developed for disease classification, utilizing a reduced error pruning algorithm (RepTree). This approach was supplemented by expert opinions to verify data, achieving a classification accuracy of 97.2% with a CNN model. Their methodology demonstrates a blend of traditional approaches with modern data collection methods.

The research by Kasinathan et al. [10] delved into insect classification and detection using modern ML techniques. They utilized datasets, such as the Wang dataset and the Xie dataset, extracting nine insect shape features after preprocessing images to grayscale. These shape features were classified using various algorithms, like Artificial Neural Network (ANN), SVM, KNN, and Naive Bayes (NB). The study demonstrated that CNNs provided the highest classification accuracy of 91% on certain insect datasets, showcasing a gradual shift towards more complex ML methods.

Abbas et al. [11] employed fuzzy logic-based histogram equalization to enhance image quality for better disease recognition in wheat leaves. This approach leverages fuzzy logic to improve image contrast, leading to more accurate disease recognition. The application of this technique represents an advancement beyond basic image processing, enhancing the quality and interpretability of images for subsequent analysis.

In their work, Kang et al. [12] proposed a DL model for pest detection, introducing an attention mechanism-enhanced single-stage object detection framework with multiscale feature fusion. This model focused particularly on identifying small-scale pests in complex backgrounds. It outperformed models, such as You Only Look Once (YOLO), EfficientDet, RetinaDet, and MobileNet, achieving the highest mean Average Precision (mAP) value of 0.91.

Xu et al.'s research [13] on wheat leaf disease identification leveraged an integrated DL framework called Recursive Feature Elimination-Convolutional Neural Network (RFE-CNN), which incorporates Residual Channel Attention Blocks (RCAB), Feedback Blocks (FB), and Elliptic Metric Learning (EML). It begins with using parallel CNNs to extract features from healthy and diseased leaves, optimizing them with RCABs, training them with FBs, and concluding with a CNN for classification. This approach resulted in an overall classification accuracy of 98.83% and a maximum testing accuracy of 99.95%.

Liu et al. [14] introduced PestNet, a DL model for multi-class pest detection and classification. PestNet blends a Channel-Spatial Attention (CSA) mechanism with a CNN backbone, utilizing a Region Proposal Network (RPN) and a Position-Sensitive Score Map (PSSM). This integration of attention mechanisms and advanced network architectures produced an mAP of 75.46%, outperforming other methods, indicating complexity in both design and application.

Chamara et al.'s [15] project is an effort toward real-time crop monitoring utilizing edge devices. They deployed a stack of Deep Convolutional Neural Networks (DCNN) models: CropClassiNet for crop type classification, CanopySegNet for canopy cover quantification, PlantCountNet for plant and weed counting, and InsectNet for insect identification. With CropClassiNet achieving 94.5% accuracy and CanopySegNet 92.83% accuracy, the project illustrated an implementation of DCNNs for integrated crop management solutions.

There are multiple dimensions explored in the earlier research; however, no ready-to-use practical solution is currently available to be used in the field. Some of them built edge devices, but they are not power-efficient and can not run for long in the field. The solutions also lack real-time monitoring capabilities. In our study, we used the IP102 dataset [16] with DL to create a real-time system for identifying pests in wheat crops. Our method improves upon past approaches by combining ConvNeXtLarge for feature extraction and MLP for classification, making it both accurate and efficient. Unlike earlier studies that relied on manual inspections or limited models, our solution includes a solar-powered device and a web app for easy, continuous monitoring. This makes pest detection faster, cheaper, and more practical for farmers, helping reduce crop losses early on. Table I compares the results and techniques of existing literature with this research.

III. Materials and Methods

A. Dataset

A database of images of pests was downloaded [16], and the images of wheat crop pests were extracted (9 categories). The distribution of images in each class is presented in Figure 1. The wheat crop pests database was split into training (2048), validation (340), and testing (1030) sets.

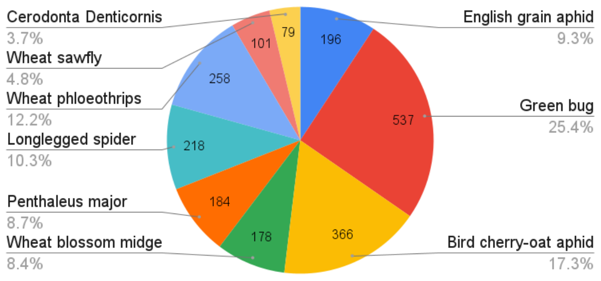


Figure 1. Distribution of images of wheat crop pests in the dataset.

B. Deep Learning models

1) MobileNetV2: A lightweight CNN model with 3.5M parameters and the ability to run on resource-constrained mobile

TABLE I. SUMMARY OF DATASETS, TECHNIQUES, AND RESULTS FROM DIFFERENT STUDIES.

Ref	Dataset Type	ML/DL Technique	Results
[7]	Camera captured whiteflies & aphids images	SVM	Training to the SVM is done with 100% accuracy
[8]	Leaf (314) and canopy (187) level hyperspectral reflectance datasets	Spectral Vegetation Indices-based Kernel Discriminant Approach (SVIKDA)	At leaf level: 89.2% At canopy level: 87%
[9]	2324 symptoms samples from our symptom-based text dataset	Decision Tree, Error Pruning Tree (RepTree), CNN	97.20%
[10]	Wang dataset (225 images) with nine insect classes and Xie dataset with 24 classes	ANN, SVM, KNN, and NB classifier	91%
[12]	Rice pest images with complex backgrounds and small-sized pests	Attention Mechanism, Multi-Scale Feature Fusion, Single-Stage Object Detection Model, YOLO, EfficientDet, RetinaDet, and MobileNet	91% mAP
[13]	CGIAR, Plant Diseases, LWDCD 2020	RCAB, FB, EML, and CNN	98.83%
[14]	Multi-class Pest Dataset 2018 (MPD2018) with over 80,000 images and 580,000 labeled pests across 16 classes	CNN with CSA, RPN, and PSSM	75.46% mAP
[15]	43,000 field crop images collected offline	Stack of four DCNN models: CropClassiNet, CanopySegNet, PlantCountNet, and InsectNet	94.5% accuracy
This research	IP102 dataset with wheat pest categories	MobileNetV2 (transfer learning), ConvNeXtLarge (featurization), KNN, Random Forest, MLP, and XGBoost	Validation accuracy: 78.72% Test accuracy: 74.51%

devices. This model introduces the concept of inverted residuals with linear bottlenecks. This approach preserves the input and output dimensions while performing the intermediate layers in a lower-dimensional space, reducing the computational cost.

2) ConvNeXtLarge: A CNN model with 197.7M parameters, with weights trained on the ImageNet dataset. ConvNeXt is a type of neural network that is built based on another design called Vision Transformers (ViTs). It uses a technique called depth-wise convolution. It is a special way of processing images where the network looks at different parts of the image separately. This technique helps to reduce the amount of calculations needed while still maintaining accuracy. We have used this model to featurize the images.

C. Machine Learning

The development of the ML model follows two main approaches. The first approach uses transfer learning on MobileNetV2, and the second approach uses feature extraction using ConvNeXtLarge and the application of classical ML techniques on the output features. Transfer learning with MobileNetV2 is similar to featurizing using MobileNetV2 and building a classifier using MLP. The dataset was featurized using ConvNeXtLarge and those features were used to create different ML models. The performance of these models was compared with the performance of the models created using MobileNetV2.

1) Transfer Learning on MobileNetV2: Considering the size of our dataset, it will not be an appropriate approach to train an architecture from scratch. This is where transfer learning works better, where we have the possibility of building high-performing models even with smaller datasets. This is the reason we decided to take this approach.

As mentioned in the left part of the flowchart in Figure 2, the final layer of a pre-trained MobileNetV2 model is removed to perform transfer learning. GlobalAveragePooling2D was applied to the output from the second-to-last layer to reduce the spatial dimensions of the input tensors. Next, a neural network layer with 100 neurons and a Rectified Linear Unit (ReLU) activation function was added. Finally, another neural network layer with 9 neurons and a softmax activation function was added to give the classification probabilities.

The model is then trained using the training dataset and then validated on the validation dataset. While training, the hyperparameters, such as learning rate (between 0.0001 and 0.01) and the number of epochs (between 10 and 50), are tuned to find the best-performing model. The ranges of learning rates and epochs are arbitrarily selected, and the plan was to extend them if the model converges in the right direction. The best model is then tested on the test dataset, and the performance is recorded in a Google sheet.

2) ConvNeXtLarge and Classical ML: As shown in the right part of the flowchart in Figure 2, ConvNeXtLarge without the last layer was used as a feature extractor to convert the images into a set of features, and the extracted features are saved into a CSV file. Generally, a CNN architecture is designed to extract various features, such as edges, shapes, etc., and the extracted features are nothing but a set of numbers. Those numbers stored in tabular data can be used with classical ML techniques to build various classifiers. Using ConvNeXtLarge, we have converted each image into a vector of size 2048. We have used various classical ML techniques, including KNN, Random Forest, XGBoost, and MLP, to train a model. The KNN technique was selected due to its simplicity and ability to work with fewer resources. During training, the K-values were varied from 2 to 14, and performance was recorded for the validation dataset. Similarly, the Random Forest technique is an ensemble of various decision trees, which are again faster to run and consume less computational resources. During training, the number of trees was varied from 10 to 100, and the depth from 1 to 7, to find the best-performing model on the validation dataset. The same settings were used for the XGBoost (which is an improvement on the Random Forest model by adding gradient boosting to it), and the performance on the validation dataset was recorded. The MLP technique is one of the simplest DL techniques and works well with datasets with fewer features. While training, the number of epochs varied from 10 to 100, and the learning rate varied from 0.00001 to 0.05; the validation results were recorded in a Google sheet. The best-performing model is then tested on the test dataset.

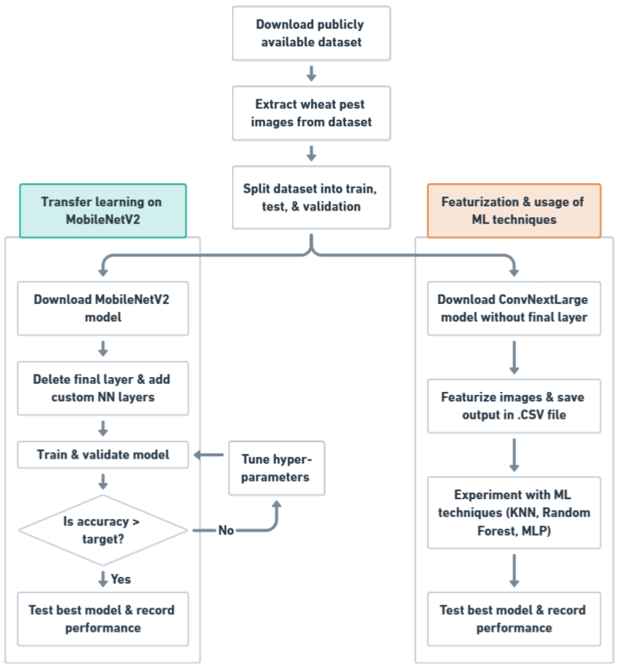

Download publicly available dataset
Extract wheat pest images from dataset
Split dataset into train, test, & validation
Transfer learning on MobileNetV2
Featurization & usage of ML techniques
Download MobileNetV2 model
Delete final layer & add custom NN layers
Train & validate model
Tune hyper-parameters
Is accuracy > target?
No
Yes
Test best model & record performance
Download ConvNextLarge model without final layer
Featurize images & save output in .CSV file
Experiment with ML techniques (KNN, Random Forest, MLP)
Test best model & record performance

Figure 2. Machine Learning pipeline: steps followed for building ML models.

D. Device Development and Model Deployment

As depicted in the hardware architecture diagram in Figure 3, for the device, the controller is a Seeed Studio XIAO ESP32S3 Sense, optimized for ML and suitable for real-time image recognition tasks. The controller has a dual-core 32-bit processor with a 240 MHz Frequency and 512 KB of SRAM. The module also has 8 MB of PSRAM that allows it to process images faster and run them through the neural network model. The controller is WiFi-enabled, which ensures it can be made part of the internet, and the captured data can be sent to an app via the cloud. It includes a built-in camera sensor with a maximum resolution of 1600 x 1200 pixels, with a Camera Serial Interface (CSI) connecting the camera to the controller. The system uses a solar-powered power bank for energy supply to ensure nonstop working even in a remote setup. As shown in the firmware flowchart in Figure 4, once the controller and camera are initialized, an image is captured and saved. The image is then serialized and sent to the web app for pest detection, where the type of pest is identified. After each image capture, the device waits five minutes before capturing the next image to ensure continuous monitoring. For demonstration purposes, pest predictions in the app are user-initiated.

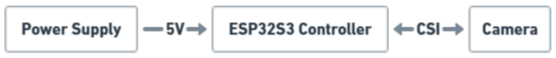


Figure 3. Hardware architecture diagram.

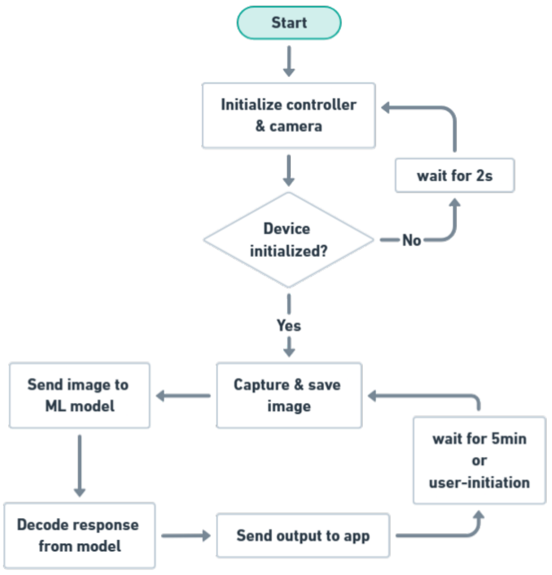


Figure 4. Firmware flowchart: logic implemented in the device.

E. Web App

The web app is developed using the Streamlit Python library and then deployed in the Streamlit cloud. The required featurizer and model are deployed as part of the app itself.

As shown in the application flowchart presented in Figure 5, when an image is received from the device in the web app, it is resized to 224 x 224 pixels and then featurized using ConvNeXtLarge. The image is processed, and the features are passed through the MLP classifier. The image and prediction of the model are then displayed to the user. Users can also see the time of detection and targeted pest removal suggestions. Each pest prediction is saved in the app for future reference.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

The code for the web application and notebooks used for featurization and training the models can be found in [17].

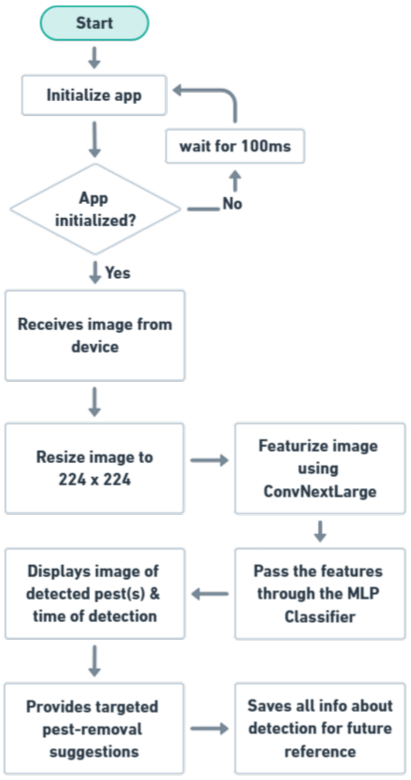

Start
Initialize app
wait for 100ms
App initialized?
No
Yes
Receives image from device
Resize image to 224 x 224
Featurize image using ConvNextLarge
Pass the features through the MLP Classifier
Displays image of detected pest(s) & time of detection
Provides targeted pest-removal suggestions
Saves all info about detection for future reference

Figure 5. Application flowchart.

IV. RESULTS

We have experimented with various ML techniques using 2 approaches. In the first, we performed transfer learning on MobileNetV2, and in the second, we used the ConvNeXtLarge model to featurize the images and then used features with ML techniques, such as KNN, Random Forest, XGBoost, and MLP, to build various models.

A. MobileNetV2

A total of 25 experiments were performed by varying the learning rates from 0.0001 to 0.01 and the epochs from 10 to 50. The best validation accuracy of 55.32% was achieved at a learning rate of 0.0005 with 40 epochs. The variation in validation accuracies with epochs for different learning rates is presented in Figure 6.

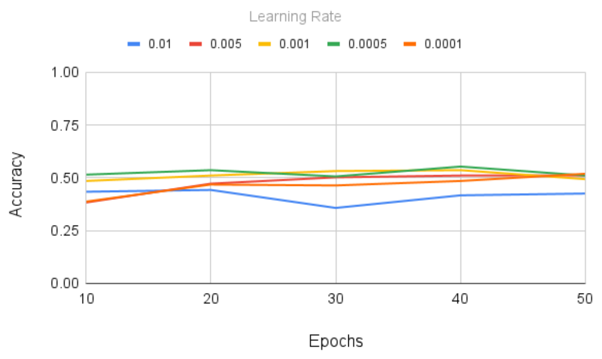


Figure 6. MobileNetV2: Validation accuracy vs. epochs for different learning rates.

B. KNN

A total of 13 experiments were performed by varying the K values from 2 to 14. The best validation accuracy of 66.81% was achieved at a K value of 8. The variation in validation accuracies with the K values can be found in Figure 7.

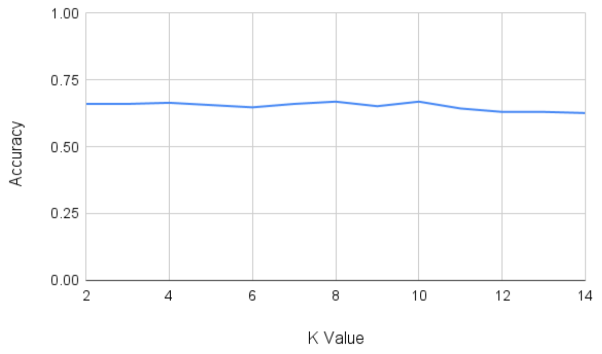


Figure 7. KNN: Validation accuracy vs. K values.

C. Random Forest

A total of 70 experiments were performed by varying the depth from 1 to 7 and the number of trees from 10 to 100. The best validation accuracy of 62.98% was achieved at a depth of 7 with 70 trees. The variation in validation accuracies with the number of trees for different depths is presented in Figure 8.

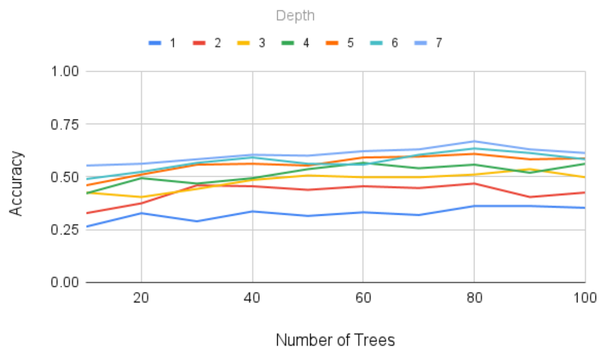


Figure 8. Random Forest: Validation accuracy vs. number of trees for different depths.

D. XGBoost

A total of 70 experiments were performed by varying the depth from 1 to 7 and the number of trees from 10 to 100. The best validation accuracy of 71.91% was achieved at a depth of 4 with 50 trees. The variation in validation accuracies with the number of trees for different depths is presented in Figure 9.

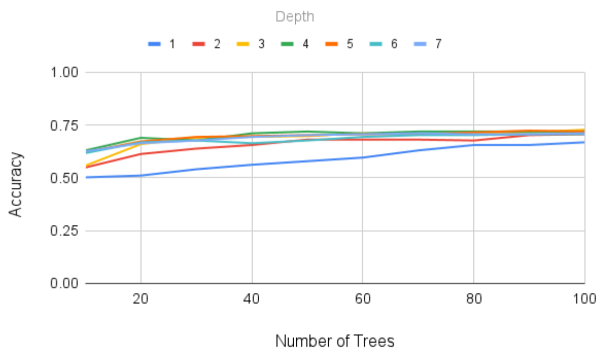


Figure 9. XGBoost: Validation accuracy vs. number of trees for different depths.

E. MLP

A total of 50 experiments were performed by varying the learning rates between 0.0001 and 0.01, and epochs between 10 and 50. The best validation accuracy of 78.72% was achieved at a learning rate of 0.001 with 30 epochs. The variation in validation accuracies with epochs for different learning rates is presented in Figure 10.

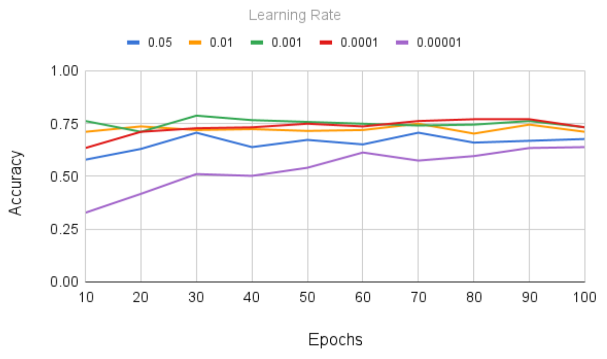


Figure 10. MLP: Validation accuracy vs. epochs for different learning rates.

F. Results summary

TABLE II. VALIDATION ACCURACIES FOR THE BEST MODELS.

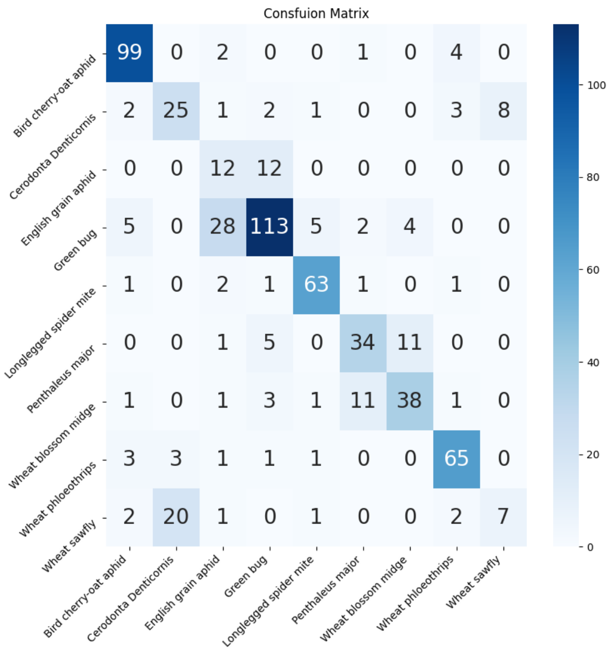
ML Model	Validation Accuracy	Precision	Recall	F1 Score
MobileNetV2	55.32%	46%	47%	46%
KNN	66.81%	72%	67%	67%
Random Forest	62.98%	64%	63%	62%
XGBoost	71.91%	74%	73%	73%
MLP	78.72%	77%	77%	77%

Out of all the ML techniques, MLP gave the best validation accuracy of 78.72%. The final model was then tested with the test dataset and achieved an accuracy of 74.51%.

As per the confusion matrix in Figure 11, the model has performed well for most of the categories except "Wheat sawfly" and "English grain aphid". To make it perform well for those 2 categories as well, we might need to add more dataset and clear images for those categories.

G. Real-world test results

The model was deployed with a web application in Streamlit Community Cloud. The device was connected to the application using a port forwarder. Once the application is ready, it takes ~6 seconds to perform inference.



	Bird cherry-oat aphid	Cerodonta Denticornis	English grain aphid	Green bug	Longlegged spider mite	Penthaleus major	Wheat blossom midge	Wheat phloeothrips	Wheat sawfly
Bird cherry-oat aphid	99	0	2	0	0	1	0	4	0
Cerodonta Denticornis	2	25	1	2	1	0	0	3	8
English grain aphid	0	0	12	12	0	0	0	0	0
Green bug	5	0	28	113	5	2	4	0	0
Longlegged spider mite	1	0	2	1	63	1	0	1	0
Penthaleus major	0	0	1	5	0	34	11	0	0
Wheat blossom midge	1	0	1	3	1	11	38	1	0
Wheat phloeothrips	3	3	1	1	1	0	0	65	0
Wheat sawfly	2	20	1	0	1	0	0	2	7

Figure 11. MLP: Confusion matrix of test results.

V. Discussion

For MobileNetV2, at a learning rate of 0.01, the accuracy varied with no consistent improvement as epochs increased, indicating possible instability, likely due to the higher learning rate causing weight oscillations. Conversely, at a lower learning rate of 0.0005, a more stable accuracy was achieved, peaking at 0.5531 with 40 epochs, suggesting better gradual convergence despite minor fluctuations at 30 epochs. The learning rate of 0.001 emerged as particularly effective, demonstrating a continual improvement across epochs, achieving a maximum accuracy of 0.5362 at 40 epochs.

For KNN experiments, as the K value increased beyond 10, a decrease in precision was observed, with K = 11, 12, 13, and 14 producing lower precision, reaching a minimum of 0.6255 at K = 14. This trend suggests that larger K values might oversmooth the decision boundary, leading to underfitting.

Regarding Random Forest, increasing both the number of trees and the depth contributes to improved accuracy, but with varying degrees of effectiveness. At a depth of 1, performance is limited in all tree counts, with the highest accuracy reaching only 36.17% at 80 and 90 trees. This underperformance is expected due to the insufficient depth to make complex decisions. The results suggest that deeper trees offer better performance, provided that there is a sufficiently large number of trees to offset the variance associated with deeper models. However, increasing the number of trees beyond 70 generally yields diminishing returns with high-depth models, likely due to saturation in ensemble benefits.

Regarding XGBoost, the influence of depth is prominent at levels 4 and 5, where accuracies consistently approach or exceed 71.91% beyond 50 trees, indicating that deeper trees can effectively tackle complexity and provide robust performance as the ensemble size grows. However, with depths 6 and 7, the incremental gain in accuracy decreases, suggesting a potential overfitting tendency or saturation, where additional depth does not necessarily translate to substantial improvements.

Regarding MLP, at a higher learning rate of 0.05, performance was moderate with fluctuating accuracies across epochs, peaking at 70.64% at both 30 and 70 epochs. This suggests that while a higher learning rate allows for rapid convergence early on, it may cause instability, leading to non-consistent improvements. As the learning rate decreases to 0.001, we observe some of the highest performance metrics, achieving a maximum accuracy of 78.72% at 30 epochs. This learning rate allows the model to explore the solution space, leading to consistent performance and better generalization. Lowering the learning rate further to 0.0001, the model maintained stability, with accuracies consistently high, peaking at 77.02% at both 80 and 90 epochs. This stability reflects the advantage of smaller learning rates, although it requires more epochs to reach effective solutions.

Overall, the best accuracy achieved out of all the experiments was from the MLP technique, suggesting the effectiveness of DL techniques with complex image datasets. However, the performance can improve with architectural changes or experimenting with other CNN models for featurization.

VI. Conclusion and Future Work

A DL model is developed for wheat pest detection, as well as a web application for real-time pest identification and targeted pest-removal suggestions. The best model achieved the best test accuracy of ~75%, and once deployed with a device, it can help mitigate crop loss in the early stages of pest infestation. The device offers low-cost, easy-to-use, efficient, convenient, and remote monitoring capabilities, eliminating the need for manual pest monitoring. The device and web app can lead to better pest control, leading to less economic loss and improved food security. However, there are limitations and areas for future study. The current image resolution is not ideal, and better images could improve model accuracy. Additionally, multiple devices (cameras) are needed to cover a large wheat field; future studies may explore the use of drone cameras with higher resolution. Notification features can also be added to the app so that the user can be notified when a pest is detected, as well as building a mobile app to enhance functionality.

References

[1] *Wheat Production by Country 2025 — worldpopulationreview.com*, https://worldpopulationreview.com/country-rankings/wheat-production-by-country, [Accessed 08-21-2025].

[2] *Wheat — agmrc.org*, https://www.agmrc.org/commodities-products/grains-oilseeds/wheat, [Accessed 08-21-2025].

[3] S. P. A. S. Lori Tyler Gula, *Researchers Helping Protect Crops From Pests*, https://www.nifa.usda.gov/about-nifa/blogs/researchers-helping-protect-crops-pests, [Accessed 08-21-2025], 2023.

[4] J. Yang and Y. Zhou, "Efficient pest classification using lightweight neural networks for sustainable pest control", in *2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI)*, IEEE, 2024, pp. 119–123.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[6] Z. Liu *et al.*, "A convnet for the 2020s", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[7] R. G. Mundada and V. V. Gohokar, "Detection and classification of pests in greenhouse using image processing", *IOSR Journal of Electronics and Communication Engineering*, vol. 5, no. 6, pp. 57–63, 2013.

[8] Y. Shi, W. Huang, J. Luo, L. Huang, and X. Zhou, "Detection and discrimination of pests and diseases in winter wheat based on spectral indices and kernel discriminant analysis", *Computers and electronics in agriculture*, vol. 141, pp. 171–180, 2017.

[9] W. Haider, A.-U. Rehman, N. M. Durrani, and S. U. Rehman, "A generic approach for wheat disease classification and verification using expert opinion for knowledge-based decisions", *IEEE Access*, vol. 9, pp. 31 104–31 129, 2021.

[10] T. Kasinathan, D. Singaraju, and S. R. Uyyala, "Insect classification and detection in field crops using modern machine learning techniques", *Information Processing in Agriculture*, vol. 8, no. 3, pp. 446–457, 2021.

[11] F. I. Abbas, N. M. Mirza, A. H. Abbas, and L. H. Abbas, "Enhancement of wheat leaf images using fuzzy-logic based histogram equalization to recognize diseases", *Iraqi Journal of Science*, pp. 2408–2417, 2020.

[12] H. Kang *et al.*, "A novel deep learning model for accurate pest detection and edge computing deployment", *Insects*, vol. 14, no. 7, p. 660, 2023.

[13] L. Xu *et al.*, "Wheat leaf disease identification based on deep learning algorithms", *Physiological and Molecular Plant Pathology*, vol. 123, p. 101 940, 2023.

[14] L. Liu *et al.*, "Pestnet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification", *Ieee Access*, vol. 7, pp. 45 301–45 312, 2019.

[15] N. Chamara, G. Bai, and Y. Ge, "Aicropcam: Deploying classification, segmentation, detection, and counting deep-learning models for crop monitoring on the edge", *Computers and Electronics in Agriculture*, vol. 215, p. 108 420, 2023.

[16] X. Wu, C. Zhan, Y. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition", in *IEEE CVPR*, 2019, pp. 8787–8796.

[17] A. Bhat, *GitHub - anikaaa22/Pest-Identification — github.com*, https://github.com/anikaaa22/Pest-Identification, [Accessed 08-21-2025].