Journal of Information Systems and Telecommunication
Special Issue, Winter,2026, PP 25-38
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Maize Leaf Disease Detection using Deep Learning Models and a DenXNet Ensemble Model

Meghna Gupta[1*], Sarika Jain[1] , Manoj Kumar[2]

[1].Amity Institute of Information Technology, Amity University Noida, Uttar Pradesh, India
[2].Faculty of Engineering & Information Sciences, University of Wollongong, Dubai, United Arab Emirates

Received: 28 Jul 2025/ Revised: 04 Dec 2025/ Accepted: 06 Jan 2026

Abstract

Maize(Corn) is considered an important crop worldwide for global production after wheat and rice. It provides food, ethanol, carbohydrates, vitamins, and other resources, making it essential to human civilization. However, it does face numerous difficulties, including pest infestations, deteriorating soil, scarce water supplies, and climate change, resulting in various yield losses. This research introduces an efficient deep learning framework for the accurate identification of maize leaf disease. Four convolutional neural network architectures, DenXNet- MobileNet, Xception, DenseNet169, and DenseNet201- were trained and evaluated using both original and augmented datasets. To ensure fairness and eliminate data leakage, the original data is divided into train, validation, and test sets, and then augmented, whereas a stratified five-fold cross-validation strategy was applied to non-augmented data. A comprehensive ablation study was conducted to compare model performance with and without augmentation and across different ensemble configurations. The study explored soft ensemble modelling using combinations of two and four base models. Among all configurations, the four-model ensemble, DenXNet, achieved the highest accuracy of 98.46% and consistency across folds, outperforming individual and partial ensembles. The proposed method demonstrates improved precision, reduced overfitting, and strong adaptability for real-world agricultural disease detection tasks.

Keywords: Deep Learning; DenseNet169; DenseNet201; Xception; Mobilenet; Ensemble Model.

1- Introduction

The agriculture sector in every country must upgrade its methods of cultivation to make farmers' lives easier and to strengthen infrastructure. A significant percentage of people across the world rely upon farming for their food and to keep their economic systems stable. Farmers are always dependent on observations and historical data, like crop yield statistics as well as climate patterns, to make important choices that assist the planet stay healthy in the short and long term. For thousands of years, agricultural activity has been a key part of human development, making sure that everyone has enough to eat **[1]**. To keep supplies of food safe, protect the natural world, and improve the health of agricultural populations, it is important to promote farming practices which prove beneficial for the environment. The fitness of plants matters for keeping the natural world in balance and for guaranteeing that agricultural activity goes smoothly at the same time. But occasionally, conventional methods of agriculture fail to produce results, and these wastes resources and boosts food prices **[2]**. Pests that affect crops are one of the most significant issues that hurt the quality and range of farm products. The most apparent part of plants is the leaves that cover them. They are also tasked with photosynthesis and the creation of chlorophyll. Numerous observations about the plant can be made by examining them **[3]**. Different algorithms have been utilized for predicting diseases in different plants, and the conclusions are made generally in terms of accuracy **[4]**. Still, one of the biggest problems is figuring out the right way to classify small datasets. It may also take a long time and a lot of money to get labeled agricultural data. The seriousness of crop diseases is an important variable that causes the decline in agricultural production. Consequently, quick detection as well as reaction are essential for safeguarding crops from spoiling while rendering farmers' lives easier. Proper detection enables it practicable to deploy chemical fertilizers and pesticides with caution **[5]**. Deep Learning (DL) models have shown outstanding ability at quickly scanning large amounts of data, particularly images, to find small signs of

✉ **Meghna Gupta**
drmeghnaphd@gmail.com

plant diseases **[6]**. Models based on deep learning have made it feasible to digitally and in real time, analyze pictures in the field. Manual inspection, on the contrary hand, is tedious and may result in blunders **[7-8]**. Once DL is combined with agricultural precision, specific changes can be made. These kinds of modifications make farming more beneficial for the environment and make ecosystems associated with agriculture stronger. Maize (Zea mays L.) is a member of the family of Poaceae and represents one of the major cereal crops in the world. Farmers cultivate it all over nations for several purposes, among which are food, livestock feed, bioethanol, and commercial applications. In recent decades, the production of maize has grown a lot in response to new technologies, greater crop yields, and improved global demand **[9]**. It is a crucial component of global agri-food systems because it can be utilized in numerous ways.

Indeed, deep learning has made significant progress in diagnosing plant diseases; however, several problems remain that need to be addressed. Most current research depends on specific CNN architectures, which occasionally fail to perform well in various conditions, like when the lighting, leaf orientation, or background noise fluctuates. Also, datasets that are restricted and not uniformly distributed make these models unreliable, which leads to biased performance during classification. Only a small number of studies have investigated ensemble-based CNN frameworks that combine the most effective elements of several distinct architectures to make predictions that are more stable and accurate. Additionally, the impact of data augmentation on maize crops has not been thoroughly examined. Consequently, there is a necessity for a systematic investigation that includes data augmentation, cross-validation, and soft averaging ensemble learning in order to improve the reliability of disease detection methodologies. The proposed research attempts to address this gap by developing and validating a hybrid ensemble model which utilizes both enhanced and ordinary datasets in order to boost accuracy, minimize overfitting, and offer an extensive framework to recognize early maize leaf diseases.

The remainder of this paper is organized as follows: Section 2 contains the related work done in this area, Section 3 describes the methodology used for the framework, Section 4 discusses the results, and Section 5 draws the conclusion of the study.

2- Related Work

Deep learning techniques have been widely employed by researchers to point out and categorize plant or agricultural diseases using machine learning algorithms, CNN, and Ensemble Modelling **[10-12]**. Xiaolin Sun *et al.* **[13]** suggested an image recognition technique for maize disease using convolutional neural networks and transfer learning. This technique saves a significant amount of training time and enhances classifier performance by using the specifications of the Inception-v3 and Inception-v4 models that have been learned on ImageNet as the start values of training. They have applied the model on 8 classes of Maize (Puccinia polysora general, Maize dwarf mosaic virus, Corn healthy, Cercospora zeaemaydis tehon and daniels general, Puccinia polysora serious, Cercospora zeaemaydis tehon and daniels serious, Corn curvularia leaf spot fungus general, and Corn curvularia leaf spot fungus serious) containing 3503 images. The authors emphasized that for deep learning models, the collection of datasets is very crucial.

Plant diseases are a broad category of diseases brought on by a variety of pathogens, like fungi, bacteria, phytoplasmas, viruses, and nematodes. These diseases can cause symptoms including wilting, discoloration, lesions, and abnormalities in the leaves, stems, roots, and fruits of the plant, among other symptoms **[14]**. Distinct approaches are currently in use for plant disease detection, using image processing **[15]**. It can be done by extracting color features (HIS model, YcbCr Model) **[16]**, shape features **[17]**, and texture features**[18]**. Enquhone Alehegn **[19]** used 7 textures, 6 colors, and 9 morphological features, a total of 22 features, for the recognition and classification analysis of images using k-nearest neighbor and Artificial Neural Network, and achieved an accuracy of 82.5% and 94.4%, respectively. Sumita Mishra et. al **[20]** offer a real-time, three-class, 88.6% accurate deep convolutional neural network technique for recognizing maize leaf disease. By modifying the pooling combinations and hyperparameters on a GPU-equipped machine, they have enhanced the network. Basavaraj et. al **[21]** created a deep CNN framework using the VGG16 model to automatically identify stressed paddy crop images taken during the booting development stage. A 92.89% accuracy rate was achieved by the model on five distinct types of paddy with varying stress levels.

Mohit Agarwal et. al. **[22]** have proposed a new CNN model with 8 hidden layers. They have compared this model with pre-trained deep learning models by testing it on the publicly available Plant Village dataset. After image augmentation, they increased the brightness of the image for image processing by a random value to increase accuracy. They have used the convolution layer, max pooling layer, dropout rate, network weight, activation function, learning rate, momentum, epochs, and batch size as hyper parameters. K. P. Panigrahi et al. **[23]** applied the supervised algorithms Naïve Bayes, Decision Tree, K-nearest neighbor, Support Vector Machine, and Random Forest on the Maize dataset on 3823 images and obtained a higher accuracy for Random Forest, i.e., 79.23% in comparison to other algorithms used in their study. They had compacted the size of the pictures to 100 x 100.

S. Pudumalar et al. **[24]** proposed an ensemble model based on CNN and the VGG16 model to improve the accuracy by removing the bias at each layer. They considered the real-time dataset of cotton crops consisting of 6 classes, from Thadikombu village, Dindigul District, Tamil Nadu. Due to limited dataset, data augmentation techniques were used to avoid overfitting issues. The proposed model attained an accuracy of 95% using the softmax function and ReLU optimizer. However, the authors were concerned about the quality of images used for classification. We found that a single model cannot cover all aspects, resulting in bias and overfitting issues. Researchers are working on the buildup of ensemble models for improving the precision of crop estimations, thus dealing with the challenges brought about by specific models **[25].**

Nabende and Murindanyi **[26]** also discuss how diseases can harm maize crops, emphasising the importance of having accessible, easy-to-use, and accurate diagnostic tools. They compared classical artificial intelligence models, custom CNNs, transfer learning methods using InceptionResNetV2, MobileNetV2, and Vision Transformers. MobileNetV2 had the best classification accuracy at 97%. Deep learning ensemble models have shown great potential in classifying plant diseases, although there are still some research gaps that make them hard to use in real-world farming situations.

Table 1: Details of different classification techniques applied to recognize different diseases in crops

Classification Model	***Crop***	***Dataset Size (images)***	***Parameters***	***Diseases Categories***	***Accuracy***	***Research Paper***
VGG16	Pearl Millet	124	Automatic Feature Detection	2	95%	[28]
CNN	Lotus	2640	Colour	11	99.54%	[10]
CNN	Maize	AI Challenger dataset	Automatic Feature Detection	3503 pictures, 8 categories	81%	[13]
KNN & ANN	Maize	800	7 textures, 6 colours, and 9 morphological features	4	82.5%(KNN), 94.4%(ANN)	[19]
Deep CNN	Corn	4382	Automatic feature detection	3	88.6%	[20]
VGG16	Paddy	30000	colour	5	92.89%	[21]
SVM, Naïve Bayes, KNN, Decision Tree, Random Forest	Maize	PlantVillage dataset	Automatic feature detection	3823 images with 4 classes	77.56 %, 77.46%, 76.16%, 74.35%, 79.23 % respectively	[23]
Ensemble Model(CNN&VGG16)	Cotton	15600	Automatic feature detection	6	95%	[15]
Resnet50	Maize	2309	Colour features	5	98.52%	[29]
CNN & MobileNet	Grape_Esca_ (Black_Measles), Tomato_Early_blight etc	87000	Automatic feature detection	38	89% & 96%	[30]
MobileNetV2, InceptionResNetV2, EfficientNetB0, ResNet50, InceptionV3,	Maize	38571	Automatic feature detection	6	97%,96%,77%,95% and 94% respectively.	[26]

Current datasets do not have adequate variety and variability compared to the real world, which makes it hard for models to work in real-world farm conditions. Additionally, the lack of explainable AI integration makes it harder for farmers to understand and trust automated systems. Also, most of the datasets that are already out there have a lot of class imbalance, which renders effectiveness biased across disease categories **[27]**. To make real, field-ready crop disease detection systems, it's necessary to deal with these challenges by using different ways to collect data, explainable computational design, and balanced training strategies. **Table 1** exhibits a summary of the different models applied to different crops and the accuracy achieved.

3-Methodology

Diseases that impact maize plants render the crop less nutritious and less available. So, it's extremely essential to identify and manage maize leaf diseases quickly and correctly to reduce losses, keep the quality of the crop, and help farmers make more money. Convolutional Neural Networks (CNNs) have become useful for diagnosing plant diseases because they can easily pull out complex features from leaf images using sequential convolution and optimization operations.

Many research studies have proven that CNN-based methods have been effective at correctly diagnosing and categorizing plant diseases. Fig. 1 shows the whole procedure of the designed model, with the main steps for finding and stopping maize leaf disease.

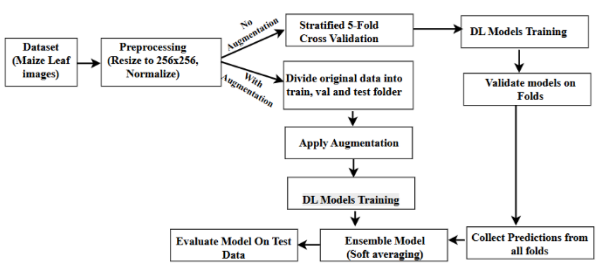


Fig. 1. Workflow of the proposed Ensemble model. Source-Self

The suggested approach incorporates different deep learning architectures into a single one using a soft averaging ensemble strategy to render maize disease classification additionally precise and trustworthy. It employs lightweight and high-performing CNN models comprising MobileNet, Xception, DenseNet169, and DenseNet201 as base learners. Each model is trained independently to get different and beneficial traits from photographs of maize leaves. MobileNet and Xception are effective at recording fine-grained spatial features because they are efficient and do not require a lot computational power. The DenseNet variants, on the other hand, enhanced feature reuse and make it easier for gradients to flow through their dense connectivity.

3-1 Data Gathering

We used the Kaggle dataset (https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset) for our experiments. At first, 4188 pictures of four different types—Common Rust, Corn Blight, Gray Leaf, and Healthy Leaf were included in the gathered dataset. 1306 pictures of common rust, 574 pictures of gray leaf,1146 pictures of corn blight, and 1162 pictures of healthy leaves were initially included in the dataset. The pictures are cropped, and dimensions are adjusted to the required input size of 256 x 256 pixels and normalized to the [0,1] range. To evaluate both generalization and overfitting resistance, two experimental setups were designed: one using original images without augmentation and another applying data augmentation to the training samples only. Overfitting is one of the greatest challenges in deep learning. It occurs when a model works superbly on training data but fails to perform well on test data that it hadn't encountered before. This is particularly frequent in tasks involving plant disease detection that require analyzing images, where the visual features might not be obvious, the datasets might not be balanced, and there might not be numerous photos per class. In order to get around this, we used a carefully selected collection of data augmentation techniques on the baseline dataset. The aforementioned comprised random rotation (between +40 and -40 degrees), width and height shifts (0.2), shear distortion (0.2), zoom (0.8-1.2), brightness variation (0.8-1.2), and horizontal flipping. In order to prevent the data from leaking, it was initially organized into train, val, and test folders, after which it was added to. To keep splitting and reconstructing under control, every single improved photo is quite distinct from the original ones. We implemented these changes because we observed that real leaf pictures possess a lot of natural contrast. Rescaling permits the model come together more rapidly during training. Rotation demonstrates the model how to keep its equilibrium intact while moving in a circle. The width alteration range shows how leaf shapes move in landscape photos, the height shift range shows how leaves are arranged in the field, and the horizontal flip shows how leaves interpret each other to avoid bias in direction. You can make your training dataset appealing without adding new images through the application of these methods. This not only protects the model from overfitting, but it also makes it easier to incorporate when new data comes in. We discovered that models trained with augmentation achieved superior performance on validation tests and displayed reduced accuracy variation between training and testing phases. It shows that they were better at making generalizations. Data augmentation is very important for

DenXNet, our ensemble model. With class imbalance in the dataset, augmentation helped artificially increase the presence of minority class samples**,** making the model less biased and more equitable in its predictions. Table 2 shows the no of enhanced images, and Fig. 2 shows sample of leaf images taken in to consideration.

Table 2: Dataset after Data Augmentation

Maize Classes	***No of original images***		***Train***	***Validation***	***Test***
Common Rust	1306		5224	653	653
Corn Blight	1146		5224	653	653
Gray Leaf	574		5224	653	653
Healthy	1164		5224	653	653

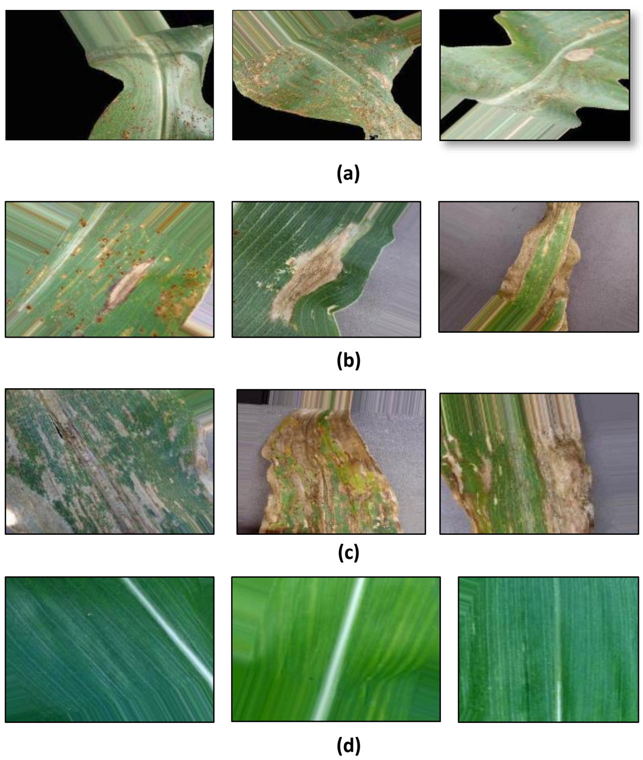


Fig. 2. Maize Leaf Dataset (a) Common Rust Diseased leaves (b) Corn Blight Diseased Leaves (c) Grey Spot Diseased Leave(d) Healthy Leaves, Source: Kaggle dataset

3-2 Data Cleaning

The quality of the dataset directly influence the efficacy and credibility of machine learning models trained on it, making data cleaning a crucial step making data cleaning a crucial step in the data preparation pipeline. When a dataset is carefully cleaned, it can yield precise and meaningful results in a variety of data-driven studies, such as the classification of diseases that affect Maize leaves and other analytical activities. Dividing the whole dataset into training, testing, and validation, the model can be trained on the training data, and after training, it will be tested on the remaining testing and validation data. We have used the standard method of splitting the data into an 80:10:10 ratio and followed the stratified 5-fold cross validation in case of non-augmented data. Cross validation helps to avoid data leakage and ensure that every fold has same proportion of images in each class . Images sent as input are converted to JPEG images and reshaped to size (256 x 256) for all used models. Following the transformation of the image data into numerical Numpy arrays, the range 0 to 1 was normalized. In the end, an unseen test image was used to validate and test the models to determine their degree of generalization.

3-3 Modelling of the proposed Ensemble model

The procedure of upgrading learning in a new process by using earlier acquired knowledge from a related task is known as transfer learning. Several things may be transferred from the earlier trained model to the newly selected task in transfer learning. The pre-trained model has already learned useful feature representations from the source data it was trained on. These feature representations in the lower and middle layers of the network can potentially be reused and fine-tuned for the new target task, avoiding having to learn them from scratch. **Table 3** summarizes the working principles and the key techniques used by different deep learning models. Among different ensemble strategies, the average ensemble offers a straightforward yet effective approach to integrate multiple heterogeneous models. In contrast to stacking methods, it eliminates the need for an additional meta-learner, which helps lower computational cost and minimizes the likelihood of overfitting. The averaging strategy differentiates itself from boosting or bagging because it isn't contingent on the model sequence or architecture type. This renders it an excellent means to integrate various frameworks like CNNs. This method improves the model's outputs more uniformly and consistently by averaging the prediction probabilities of the base models. It also assists in the model generalize more successfully when applied to fresh data.

The proposed ensemble model (DenXNet) is computationally modelled using four deep learning models: Xception, MobileNet, DenseNet169, and DenseNet201. These models were selected because they had better performance on their own when trained. First, predictions are generated from individual models using cross validation, and then these predictions are aggregated to get the final ensemble prediction. Predictions are made individually for each model in the ensemble, M_i, for a given input picture X. By averaging the prediction probabilities across multiple models, an ensemble blending technique helps mitigate bias and variance. Averaging the outputs of each model enhances the system's overall ability to extrapolate and stability since each model reflects various aspects of its data distribution. $P_i(X)$, which represents each of these unique predictions, is a measure of the model's confidence ratings for each of the categorization categories. The ensemble prediction is obtained by a weighted averaging procedure once the predictions from each model have been generated. The Ensemble Prediction(X) is calculated as the mean of the individual model forecasts by combining the predictions from each of the constituent models, M_i.

$$Ensemble\ Prediction\ (X) = \frac{1}{N}\sum_{i=1}^{N} P_i(X) \quad (1)$$

In this equation (1), N represents the total number of models in the ensemble. The resulting ensemble prediction

Table 3. Working principles and techniques used by Deep Learning model

Classification Model	Working Principle	Key Techniques	Advantages	Disadvantages
VGG16	A deep learning model with 16 layers arranged sequentially for extracting hierarchical features.	Small 3×3 convolutional kernels, Max pooling layers for downsampling, and ReLU activation functions.Dense, fully connected layers at the output.	Straightforward architecture, easy to implement, Strong baseline for classification tasks., Pre-trained models are widely available.	Consumes significant memory and computational power, Risk of overfitting on a small dataset. Inefficient use of parameters.
VGG19	An extension of VGG16, featuring 19 layers to enable deeper feature learning.	The same techniques as VGG16 with additional layers for enhanced capacity and uniform design structure.	Marginally better performance than VGG16, Simple and systematic architecture, Accessible pre-trained weights.	Increased resource requirements compared to VGG16, Slower training, and inference times.
InceptionV3	Processes feature using parallel convolutional paths in "Inception modules," enabling multi-scale feature extraction.	Decomposed convolutions to reduce the computational cost, Auxiliary classifiers for better gradient propagation., Batch normalization to stabilize training and parallel multi-scale processing.	Efficient computation with high accuracy., Effective for large and diverse datasets, it captures features at multiple resolutions.	More complex to design and implement. It requires careful hyperparameter tuning.
EfficientNetB0	A compact architecture that adjusts depth, width, and resolution systematically for optimized efficiency.	Compound scaling balances network dimensions and depth-wise separable convolutions to save computation	Scalable across different tasks, Lightweight for smaller datasets.	May underperform on very complex tasks. Scaling parameters need to be fine-tuned for the best results.
EfficientNetB7	A larger variant of EfficientNetB0 with increased depth, width, and resolution for improved representation capacity.	Same as EfficientNetB0 but scaled up, incorporates compound scaling to expand dimensions proportionally.	Exceptional performance on challenging benchmarks, maintains efficiency even at larger scales.	Demands substantial computational resources and is slower to train.
DenseNet169	Utilizes dense connectivity, where each layer receives inputs from its preceding layers, enhancing feature reuse and gradient flow.	Dense layer connections to improve feature propagation, Transition layers compress feature maps, and encourage parameter efficiency through reuse.	Reduces parameter redundancy, Mitigates vanishing gradients, and performs well on moderately sized datasets.	With high memory usage due to dense connections, Computational overhead grows with depth.
DenseNet201	A deeper variant of DenseNet169 is designed to extract more complex and detailed features through additional layers.	Similar techniques to DenseNet169 with increased depth, Improved transition, and compression strategies.	Excels in extracting detailed representations, Strong performance on complex tasks, and Efficient parameter utilization.	Higher computational and memory demands than DenseNet169. Training becomes slower as the network deepens.

Xception	A refinement of Inception that replaces standard convolutions with depthwise separable convolutions for greater efficiency and flexibility.	Depthwise separable convolutions split spatial and channel-wise filtering, and Skip connections improve gradient flow.	Competitive accuracy across datasets reduces redundant computations in convolution operations.	Requires careful tuning to achieve optimal performance and demands higher computational resources despite being efficient.
MobileNet	A lightweight model designed for mobile and embedded systems, focusing on reducing computational and memory overhead.	Depthwise separable convolutions to minimize computation, Width and resolution multipliers adjust the model size, Global average pooling prevents overfitting.	Highly efficient for real-time applications, optimized for resource-constrained devices.	Lower accuracy on very complex datasets, Limited capability to handle tasks requiring deep feature extraction.
Ensemble Model	Combines predictions from multiple architectures (e.g., Xception, MobileNet, DenseNet169, DenseNet201) to enhance overall performance and robustness.	Weighted averaging assigns importance to each model, Stacking uses outputs of individual models as inputs for a meta-classifier, and Bagging improves prediction stability.	Aggregates strengths of diverse models, improves accuracy and reduces bias, and increases robustness against overfitting.	Higher computational and memory requirements due to multiple models, Complex implementation, and integration process.

provides a consolidated estimate of the classification outcome, harnessing the collective insights from multiple models to enhance accuracy and reliability. The algorithm for the proposed model is as follows:

Step 1: Load the pre-trained models (Xception, MobileNet, DenseNet169, DenseNet201) from their respective paths.

Step 2: Modify the loaded models to remove the last layer (Softmax layer), as we want to use them as feature extractors.

Step 3: Define an input tensor for our ensemble model with the shape (256, 256, 3).

Step 4: Pass the input tensor through each pre-trained model to get their outputs.

Step 5: Combine the outputs of the pre-trained models by taking their average.

Step 6: Define the ensemble model using the input tensor and the averaged outputs.

Let

- I will be the input image tensor with dimensions (256, 256, 3)
- M_1 be the Xception model.
- M_2 be the MobileNet model.
- M_3 be the DenseNet169 Mode
- M_4 be the DenseNet201 model
- O_1 be the output tensor of M_1 given input I
- O_2 be the output tensor of M_2 given input I.
- O_3 be the output tensor of M_3 given input I.
- O_4 be the output tensor of M_4 given input I.
- E be the ensemble model.

The operations performed in the code can be mathematically modeled as follows:

1. Load Pre-Trained Models: M_1, M_2, M_3, M_4 are pre-trained models.
2. Remove Last Layer and Modify Models: Let $fn_1(\cdot)$, $fn_2(\cdot)$, $fn_3(\cdot)$ and $fn_4(\cdot)$ be the functions representing the modified Xception, MobileNet, DenseNet169 and DenseNet201 models, respectively, after removing their last softmax layer

$$fn_1\ (I) = Modified\ \ Xception\ (I) \tag{2}$$

$$fn_2\ (I) = Modified\ \ MobileNet\ (I) \tag{3}$$

$$fn_3\ (I) = Modified\ \ DenseNet169\ (I) \tag{4}$$

$$fn_4\ (I) = Modified\ \ DenseNet201\ (I) \tag{5}$$

3. Ensemble Model: The ensemble model, E, takes I as input and computes the average of the outputs of $f_1, f_2, f_3,$ *and* f_4:

$$O_{ensemble} = \frac{fn_1\,(I) + fn_2\,(I) + fn_3\,(I) + fn_4\,(I)}{4} \tag{6}$$

where, $O_{ensemble}$ is the final average probabilities

Therefore, the mathematical model for the ensemble model E can be exemplified as:

$$E\ (I) = O_{ensemble} = \frac{1}{4}\sum_{i=1}^{4} fn_i\ (I) \tag{7}$$

Then, the final predicted class label $\hat{y}$ is:

$$\hat{y} = \arg\max_{k\,\in\{1,\ldots,4\}} E\ (I)\ [k]\ ) \tag{8}$$

where, $\hat{y}$ is the predicted class label from the ensemble. Each base model is trained using categorical cross-entropy loss:

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y_k \, . \, \log \widehat{y_k} \quad (9)$$

Here, K is the number of classes,
$y = [y_1, y_2, \ldots, y_K]$ be the one-hot encoded ground truth label vector, where $y_k = 1$ if the sample belongs to class k, and 0 otherwise .
$\hat{y} = [\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_k}]$ be the predicted probability from the model E(I).

3-4- Model Configuration & Architecture

The pre-trained individuals used for the suggested job are VGG16, VGG19, InceptionV3, EfficientNetB0, EfficientNetB7, DenseNet169, DenseNet201, Xception, Mobilenet, and an Ensemble model using Xception, MobileNet, DenseNet169, and DenseNet201 CNN models using Adam Optimizer for the classification of 4 classes of Maize. To categorize images of maize crop fields, the model is developed using an elaborate Python library called Keras, which is used as a backend atop an open-source deep learning framework called TensorFlow. The input is processed through a heap of convolutional layers with a convolution filter size of 3×3 and convolution strides in x and y directions (1,1) pixels. The image dimension is set to 256 × 256 pixels with depth 3 (RGB channels). After convolution, the spatial size is maintained with hyper-parameter padding 1. All hidden layers have taken into account the activation function "ReLU," and the last layer applies the "softmax" function to guarantee that the projected probability output values fall between 0 and 1. The network uses 25 epochs and a batch size of 32 with a learning rate of 0.001 and a dropout rate of 0.5. The network has been optimized through the use of the logarithmic loss function and gradient descent optimization technique with categorical cross-entropy. The input of the model consists of 256 × 256-pixel images of maize crops. After that, it has a sequence of convolutional and pooling layers that extract features, an output layer, and a fully connected layer that interprets the information. Four neurons make up the output layer, which is the actual number of classes that the image being processed must be categorized into. The models were run on a desktop PC that was set up with an Intel Core i7-8565U CPU @ 1.80 GHz, a 64-bit operating system, x64-based processor, 16.0 GB (15.9 GB usable) RAM, and one NVIDIA GeForce GTX 1070 GPU.

4-Results and Discussion

Ensemble modelling helps to amalgamate the predictions of multiple models and provides more precise results with high accuracy and precision **[31-34]**. It is a robust and efficient means of integrating multiple heterogeneous models without requiring an additional meta-learner.

By averaging the probabilistic outputs of base model, it minimizes overfitting and yields more stable, generalized prediction across diverse datasets. In this study, nine individual models- VGG19, VGG16, InceptionV3, EfficientNetB0, EfficientNetB7, DenseNet169, Densenet201, Xception and MobileNet were implemented and evaluated. Additionally, two ensemble models combining MobileNet with DenseNet169 and MobileNet with Xception along with proposed ensemble model, DenXNet based on four models were developed to analyze the impact of hybridization on performance. However experimental results revealed that the individual four model configurations achieved superior performance compared to two model ensembles, demonstrating their stronger feature extraction and generalization capabilities. Using the corresponding images of the maize crop, the models were trained and evaluated, taking into account the pre-learned weights of ImageNet dataset in each layer until convergence. **Table 4** shows the training accomplishments of various models for the Maize diseased leaf images dataset using the Adam optimizer. The experimental results demonstrate how different deep learning models perform when data augmentation is utilised and when it isn't. DenseNet201 had the best testing performance of 0.9878 with augmentation among the individual architectures. DenseNet169 (0.969) and InceptionV3 (0.9696) were the next closest. Since their convolutional layers are somewhat densely connected, these models were capable of predicting well and extracting features efficiently. The results further indicate that data augmentation renders all models considerably more accurate by making them more robust to variations in lighting, orientation, and leaf conditions. For example, MobileNet's testing accuracy improves from 0.9334 (without enhancement) to 0.948 (with enhancement).

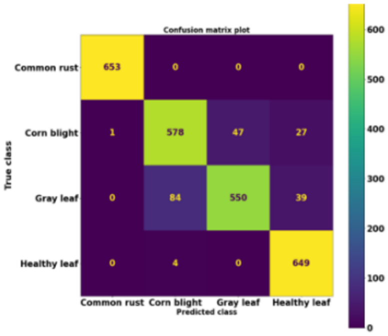
Table 4. Training performance of models for Maize Leaf disease images Source: Anaconda

	Accuracy Score With Augmentation			***Accuracy Score Without Augmentation***		
Model name	***Training***	***Validation***	***Testing***	***Training***	***Validation***	***Testing***
VGG19	0.9389	0.936	0.940	0.910	0.905	0.910
VGG16	0.955	0.960	0.9484	0.925	0.9324	0.922
InceptionV3	0.954	0.962	0.9696	0.925	0.935	0.935
EfficientNetB0	0.9767	0.974	0.9818	0.950	0.965	0.955
EfficientNetB7	0.9307	0.968	0.9666	0.905	0.935	0.935
DenseNet169	0.986	0.976	0.969	0.9606	0.9606	0.9577
DenseNet201	0.988	0.988	0.9878	0.9376	0.9486	0.9479
Xception	0.9272	0.9420	0.9575	0.9028	0.9046	0.9024
MobileNet	0.94	0.95	0.948	0.93	0.9272	0.9334
Ensemble Model: MobileNet, Densenet169	0.995	0.987	0.98	0.9675	0.9646	0.965
Ensemble Model: MobileNet , Xception	0.985	0.960	0.955	0.92	0.9354	0.9354
Ensemble Model: Xception, MobileNet, DenseNet169,DenseNet201	0.9946	0.9812	0.9896	0.9787	0.9742	0.97

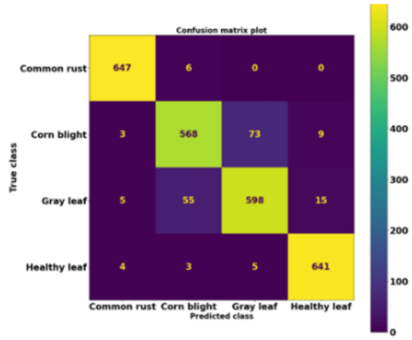
It shows that augmentation works to safeguard models from overfitting while additionally rendering them adjustable. Ensemble learning had been implemented to investigate into potential advantages of integrating models. The MobileNet-DenseNet169 ensemble had an impeccable test accuracy of 0.980 (with augmentation), which is far greater than a majority of individual models. The MobileNet-Xception setup, on the contrary hand, hadn't been as accurate. Thus demonstrating that model integration counts for ensemble success. The DenXNet four-model ensemble had its highest testing accuracy, 0.9846, without as well as with the enhancement. It executed better than any of the other setups. The enhancement shows just how beneficial it is to integrate numerous architectures that can record multiple kinds of feature hierarchy and spatial representations. The ensemble model is more dependable, indicating that it can extrapolate more accurately, stay stable, and cope with new data.

The study illustrates that individual CNN-based models, comprising DenseNet201 and InceptionV3, have been very accurate. However, hybrid ensemble architectures provide the most precise and trustworthy outcomes. These traits allow them to be useful for practical applications, such as discovering diseases in crops as well as maintaining an eye on their health. **Fig.3** shows the incorrect predictions through the creation of confusion matrices compared to all four unique models and the proposed Ensemble Model. All of these were tested on 653 maize leaf images that had never been seen before and were made using data augmentation. The individual models function quite well, with DenseNet201 and DenseNet169 correctly recognizing the most samples in all four disease classes: Common Rust, Corn Blight, Grey Leaf Spot, and Healthy. Gradient-weighted Class Activation Mapping (GRAD-CAM) was likewise employed to illustrate the attention regions of different models to make them easier to understand. The heat maps in **Fig. 4** illustrate how various architectures focused on the diseased portions of maize leaves. Mobilenet produced broader and less defined attention zones, whereas DenseNet169 and Xception showed improved localization around infected areas. The MobileNet-DenseNet169 ensemble performed an outstanding job of identifying diseased areas while minimizing background noise. The Xception-DenseNet169 ensemble performed a superior task of finding lesions with minimal background noise. DenXNet demonstrated that the majority of biologically significant and broadly distributed activations, indicating that the ensemble approach enhanced both classification accuracy and the ability to understand frameworks.Still, there were certain ones misclassifications, especially between disease categories that are similar, like Common Rust and Grey Leaf Spot. This can happen given that the symptoms of each of these illnesses are similar, rendering it rendering it challenging to separate them. **Fig. 5** illustrates how the model inaccurately anticipated Blight with a confidence score of 29.8%, given that the true label for the maize leaf

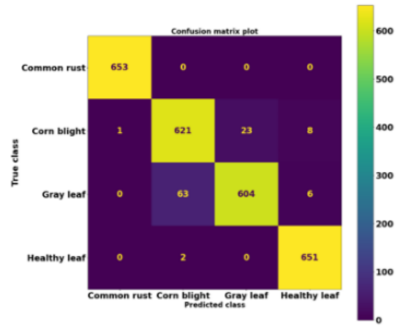
Journal of Information Systems and Telecommunication
Special Issue, Winter,2026, PP 25-38
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773



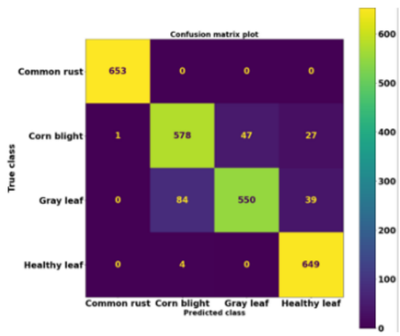
(a) DenseNet201



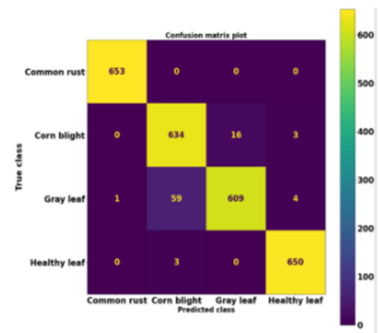
(b) DenseNet169



(c) MobileNet



(d) Xception



(e) DenXNet

Fig 3. Confusion matrices for four pre-trained deep learning models and the proposed ensemble model

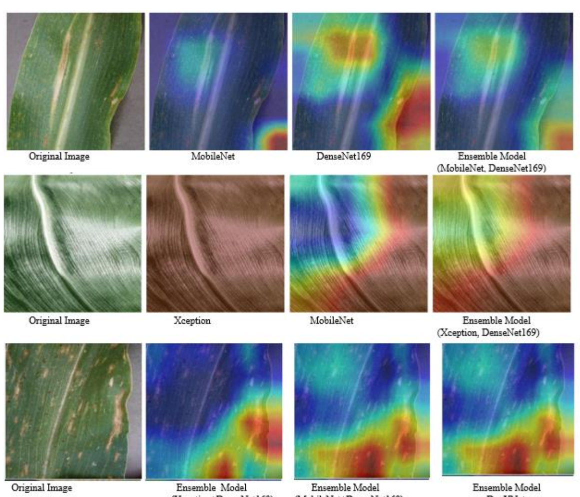


Fig. 4. Grad Cam Visualizations showing model attention on Maize Leaf lesion

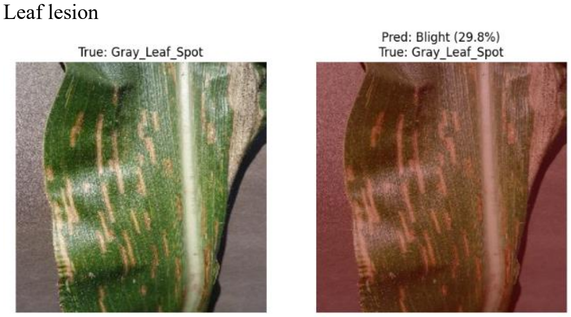


Fig.5 Error and Misclassification between Gray leaf and Blight

sample is Grey Leaf Spot. The Grad-CAM visualisation illustrates that the model's focus had been spread over both diseased and healthy leaf areas. This indicates that it was hard to pinpoint the subtle lesion patterns that are unique to Grey Leaf Spot. This misclassification is predominantly due to the fact that Blight and Gray Leaf Spot look similar. Both have long, brownish lesions with parallel veins and similar texture gradients. These patterns become harder to tell differentiate when the conditions and background change, leading to the model mix up their spatial and color features. Also, the overlapping feature representations that are learned during convolutional processing can make it harder to differentiate the difference between classes. These cases demonstrate the importance of fine-grained extraction of characteristics and attention modification in distinguishing diseases with similar visual traits more effectively. Future enhancements could include incorporation of multi-scale attention modules, spectral feature learning, or contrast reduction optimization to augment the model's discriminative ability. **Table 5** lists the performance metrics calculated from the confusion matrices, and includes classification accuracy, recall, and precision for each class. A more comprehensive look at the results metrics shows that every single deep learning model encompasses its own strengths, and weaknesses when it applies to classifying the maize leaf disease. The Common rust category had the most variances across models. As an example, VGG19 had a high recall of 0.94 but a very low precision of 0.08, which

✉ **Meghna Gupta**
drmeghnaphd@gmail.com

indicates that it regularly misclassified other diseases as Common Rust. The texture and colour deviations introduced by enhancement may have prompted the model to overfit generalised rust-like patterns, thereby failing to capture deeper, discriminative details. On the contrary hand, more complicated architectures like DenseNet201 and EfficientNetB7 got nearly perfect accuracy (0.99 and 0.95, respectively), showing that they were considerably better at detecting structural features that were specific to diseases. Another problem that continued to arise was incorrectly identifying Grey Leaf and Healthy Leaf because they matched so much alike in colour and texture. Models consisting of InceptionV3 and EfficientNetB0 achieved F1-scores of approximately 0.84–0.92 for Grey Leaf, indicating that the models were only slightly confused. DenseNet201 and MobileNet did better than others with these classes, showing a good equilibrium among precision and recall. MobileNet is a lightweight model that had a significant recall rate, making it a good choice for actual-time or mobile apps. However, it did over-detect healthy samples a little bit. Ensemble methods gave the best overall results. Models that combined MobileNet, DenseNet169, Xception, and DenseNet201 got almost perfect F1-scores, which fixed errors in each model. This shows that ensembles work more successfully when various architectures have different strengths, rendering them more resilient and reliable.

5- Conclusion

The present research focused on developing an automated framework for identifying the presence of maize leaf illnesses employing both individual deep learning architectures and ensemble CNN architectures. We trained and evaluated nine revolutionary CNN architectures: VGG19, VGG16, InceptionV3, EfficientNetB0, EfficientNetB7, DenseNet169, DenseNet201, Xception, and MobileNet. We employed two distinct methods: one with enhancement and one without any enhancement. Data augmentation significantly enhanced the framework's capacity to generalize by making its features more unique and less likely to overfit. The proposed DenXNet model, which is formed up of DenseNet-169, DenseNet-201, Xception, and MobileNet, performed the best overall out of all the models that have been tested. It enjoyed the best testing accuracy, better stability, and stronger strength than those with one-model or two-model ensembles. DenXnet combined different architectures so that the model could use the extensive connectivity of DenseNets, the depthassumptions across a wide range of pictures conditions.

Table 5: Training Performance of Model for Maize Leaf with augmentation

Model	**Class name**	**Precision**	**Recall**	**F1-Score**
VGG19	Common Rust	0.8	0.04	0.08
	Corn Blight	0.58	0.94	0.71
	Gray Leaf	0.89	0.81	0.85
	Healthy Leaf	0.69	0.97	0.81
VGG16	Common Rust	0.97	0.43	0.59
	Corn Blight	0.61	0.97	0.75
	Gray Leaf	0.94	0.78	0.86
	Healthy Leaf	0.86	0.98	0.92
InceptionV3	Common Rust	0.99	1	1
	Corn Blight	0.8	0.92	0.85
	Gray Leaf	0.93	0.76	0.84
	Healthy Leaf	0.95	0.97	0.96
EfficientNetB0	Common Rust	1	0.42	0.9
	Corn Blight	0.73	0.93	0.82
	Gray Leaf	0.87	0.97	0.92
	Healthy Leaf	0.83	0.99	0.9
EfficientNetB7	Common Rust	0.95	1	0.97
	Corn Blight	0.85	0.92	0.88
	Gray Leaf	0.94	0.81	0.87
	Healthy Leaf	0.97	0.97	0.97
DenseNet169	Common Rust	0.98	0.99	0.99
	Corn Blight	0.90	0.87	0.88
	Gray Leaf	0.88	0.89	0.89
	Healthy Leaf	0.96	0.98	0.97
DenseNet201	Common Rust	0.99	1	0.99
	Corn Blight	0.87	0.89	0.88
	Gray Leaf	0.92	0.82	0.87
	Healthy Leaf	0.91	0.99	0.95
Xception	Common Rust	1	1	1
	Corn Blight	0.87	0.89	0.88
	Gray Leaf	0.92	0.82	0.87
	Healthy Leaf	0.91	0.99	0.95
MobileNet	Common Rust	1	1	1
	Corn Blight	0.91	0.95	0.93
	Gray Leaf	0.96	0.90	0.93
	Healthy Leaf	0.98	1	0.99
Ensemble Model: MobileNet, Densenet169	Common Rust	0.98	0.98	0.98
	Corn Blight	0.92	0.95	0.93
	Gray Leaf	0.91	085	0.88
	Healthy Leaf	1	1	1
Ensemble Model: MobileNet , Xception	Common Rust	0.97	0.97	0.97
	Corn Blight	0.84	0.93	0.90
	Gray Leaf	0.84	0.79	0.82
	Healthy Leaf	0.99	0.99	0.99
Ensemble Model : (Xception, MobileNet, DenseNet169,DenseNet201)	Common Rust	1	1	1
	Corn Blight	0.91	0.97	0.94
	Gray Leaf	0.97	0.90	0.94
	Healthy Leaf	0.98	1	0.99

dependent separable convolutions of MobileNet, and the effective retrieval capacity of Xception. This combination of characteristics lets DenXNet decide simultaneously fine-grained and global disease features, which boosts the accuracy of classification and its capacity to make

Grad-CAM representations also showed that the ensemble model properly focused on areas affected by disease. This made the model easier to comprehend while providing it more credibility as a reliable tool. The model was much better at generating predictions when augmentation techniques were employed because they reduced overfitting and boosted feature diversity. The ensemble configurations were more accurate and stable than the individual models, which shows how integrating different feature visualizations can be helpful. By showing how the models found affected by the disease areas on the leaf surface, Grad-CAM representation made the models easier to understand. The aforementioned visual representations showed the fact that the ensemble models not only made predictions that were more accurate, but they also identified more specific and biologically relevant areas of feature activation. However, there were still a few minor errors in categorising diseases that looked similar, like Blight and Grey Leaf spot. This was probable because their symptoms were similar and the visual differences were small. In general, the study shows that incorporating data augmentation and ensemble modelling can be carefully done to substantially enhance the classification of illnesses in maize. In addition, CNN-based and ensemble models have shown immense potential, there nevertheless remain some research gaps which require to be filled. The research overwhelmingly emphasized on convolutional architectures and did not incorporate highly sophisticated temporal or spatial mechanisms such as BiLSTM, Self-attention, or Transformer networks, which could improve contextual feature extraction and inter-class discrimination. Later studies could enhance this study through the inclusion of Transformer-based hybrid architectures to further clarify long-range dependencies and intricate variations within corresponding disease categories integrating explainable AI (XAI) methodologies beyond Grad-CAM, which include Layer-wise Relevance Propagation or SHAP, could increase interpretability. Finally, deployment on edge or mobile devices enables real-time, low-cost field diagnosis, directly assisting farmers in precision agriculture systems.

References

[1] A. H. Ali, A. Youssef, M. Abdelal, and M. A. Raja, "An ensemble of deep learning architectures for accurate plant disease classification," *Ecol. Inform.*, vol. 81, 102618, 2024, doi: 10.1016/j.ecoinf.2024.102618.

[2] S. Vallabhajosyula, V. Sistla, and V. K. K. Kolli, "A novel hierarchical framework for plant leaf disease detection using residual vision transformer," *Heliyon*, vol. 10, no. 9, 2024, doi: 10.1016/j.heliyon.2024.e29912.

[3] R. Lumbantoruan, N. Rajagukguk, A. U. Lubis, M. Claudia, and H. Simanjuntak, "Two-step convolutional neural network classification of plant disease," *IAES Int. J. Artif. Intell.*, vol. 14, no. 1, pp. 584–591, 2025, doi: 10.11591/ijai.v14.i1.pp584-591.

[4] L. Miao *et al.*, "A high-precision automatic diagnosis method of maize developmental stage based on ensemble deep learning with IoT devices," *Comput. Electron. Agric.*, vol. 227, 109608, 2024, doi: 10.1016/j.compag.2024.109608.

[5] W. H. Zeng, H. Li, G. Hu, and D. Liang, "Identification of maize leaf diseases by using the SKPSNet-50 convolutional neural network model," *Sustain. Comput. Informatics Syst.*, vol. 35, **100695**, 2022, doi: 10.1016/j.suscom.2022.100695.

[6] I. Attri, L. K. Awasthi, T. P. Sharma, and P. Rathee, "A review of deep learning techniques used in agriculture," *Ecol. Inform.*, vol. 77, 102217, 2023, doi: 10.1016/j.ecoinf.2023.102217.

[7] M. Yang, A. Sekhari Seklouli, L. Ren, Y. He, X. Yu, and Y. Ouzrout, "A new mobile diagnosis system for estimation of crop disease severity using deep transfer learning," *Crop Prot.*, vol. 184, 106776, 2024, doi: 10.1016/j.cropro.2024.106776.

[8] S, Jenifer, M J, Carmel Mary Belinda. Convolutional Neural Networks for Medical Image Segmentation and Classification: A Review. Journal of Information Systems and Telecommunication (JIST) 2023;11(44):347-358.10.61186/jist.37936.11.44.347

[9] O. Erenstein, M. Jaleta, K. Sonder, K. Mottaleb, and B. M. Prasanna, "Global maize production, consumption and trade: trends and R&D implications," *Food Secur.*, vol. 14, no. 5, pp. 1295–1319, 2022, doi: 10.1007/s12571-022-01288-7.

[10] T. Phattaraworamet, S. Sangsuriyun, P. Kutchomsri, and S. Chokphoemphun, "Image classification of lotus in Nong Han Chaloem Phrakiat Lotus Park using convolutional neural networks," *Artif. Intell. Agric.*, vol. 11, pp. 23–33, 2024, doi: 10.1016/j.aiia.2023.12.003.

[11] M. A. Jabed and M. A. Azmi Murad, "Crop yield prediction in agriculture: A comprehensive review of machine learning and deep learning approaches, with insights for future research and sustainability," *Heliyon*, vol. 10, no. 24, 2024, doi: 10.1016/j.heliyon.2024.e40836.

[12] A. Pal and V. Kumar, "AgriDet: Plant Leaf Disease severity classification using agriculture detection framework," *Eng. Appl. Artif. Intell.*, vol. 119, 105754, 2023, doi: 10.1016/j.engappai.2022.105754.

[13] X. Sun and J. Wei, "Identification of maize disease based on transfer learning," *J. Phys. Conf. Ser.*, vol. 1437, no. 1, 2020, doi: 10.1088/1742-6596/1437/1/012080.

[14] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Comput. Intell. Neurosci.*, 2016, doi: 10.1155/2016/3289801.

[15] X. Lv, X. Zhang, H. Gao, T. He, Z. Lv, and L. Zhangzhong, "When crops meet machine vision: A review and development framework for a low-cost nondestructive online monitoring technology in agricultural production," *Agric. Commun.*, vol. 2, no. 1,

100029, 2024, doi: 10.1016/j.agrcom.2024.100029.

[16] P. Chaudhary, A. K. Chaudhari, A. Cheeran, and S. Godara, "Color Transform Based Approach for Disease Spot Detection on Plant Leaf," *Int. J. Comput. Sci. Telecommun.*, vol. 3, no. 6, pp. 65–71, 2012,

[17] S. B. Patil and S. K. Bodhe, "Leaf disease severity measurement using image processing," *Int. J. Eng. Technol.*, vol. 3, no. 5, pp. 297–301, 2011.

[18] M. U. Ahmad, S. Ashiq, G. Badshah, A. H. Khan, and M. Hussain, "Feature Extraction of Plant Leaf Using Deep Learning," *Complexity*, 2022, doi: 10.1155/2022/6976112.

[19] E. Alehegn, "Maize Leaf Diseases Recognition and Classification Based on Imaging and Machine Learning Techniques," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 12, pp. 1–11, 2017.

[20] S. Mishra, R. Sachan, and D. Rajpal, "Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition," Procedia *Comput. Sci.*, vol. 167, pp. 2003–2010, 2020, doi: 10.1016/j.procs.2020.03.236.

[21] B. S. Anami, N. N. Malvade, and S. Palaiah, "Arti fi cial Intelligence in Agriculture Deep learning approach for recognition and classi fi cation of yield affecting paddy crop stresses using fi eld images," *Artif. Intell. Agric.*, vol. 4, pp. 12–20, 2020, doi: 10.1016/j.aiia.2020.03.001.

[22] M. Agarwal, S. K. Gupta, and K. K. Biswas, "Development of Efficient CNN model for Tomato crop disease identification," *Sustain. Comput. Informatics Syst.*, vol. 28, 100407, 2020, doi: 10.1016/j.suscom.2020.100407.

[23] K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms," *Adv. Intell. Syst. Comput.*, vol. 1119, pp. 659–669, 2020, doi: 10.1007/978-981-15-2414-1_66.

[24] S. Pudumalar and S. Muthuramalingam, "Hydra: An ensemble deep learning recognition model for plant diseases," *J. Eng. Res.*, 2024, doi: 0.1016/j.jer.2023.09.033.

[25] S. K. Subramanya and N. Bettahalli, "Enhanced detection of tomato leaf diseases using ensemble deep learning: INCVX-NET model," *IAES Int. J. Artif. Intell.*, vol. 13, no. 4, pp. 4757–4765, 2024, doi: 10.11591/ijai.v13.i4.pp4757-4765.

[26] J. Nakatumba-Nabende and S. Murindanyi, "Deep learning models for enhanced in-field maize leaf disease diagnosis," *Mach. Learn. with Appl.*, vol. 20, 100673, 2025, doi: 10.1016/j.mlwa.2025.100673.

[27] Maiti, Abdallah, Hanini, Mohamed, Abarda, Abdallah. Resolving Class Imbalance in Medical Classification: Technique Comparison and Performance Evaluation. Journal of Information Systems and Telecommunication (JIST) . 2025;13(51):177-188. .61882/jist.49725.13.51.177

[28] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, 2019, doi: 10.1016/j.compind.2019.02.003.

[29] G. Wang, H. Yu, and Y. Sui, "Research on Maize Disease Recognition Method Based on Improved ResNet50," *Mob. Inf. Syst.*, 2021, doi: 10.1155/2021/9110866.

[30] M. M. Khalid and O. Karan, "Deep Learning for Plant Disease Detection," *Int. J. Math. Stat. Comput. Sci.*, vol. 2, pp. 75–84, 2023, doi: 10.59543/ijmscs.v2i.8343.

[31] A. Daza, C. F. Ponce Sánchez, G. Apaza-Perez, J. Pinto, and K. Zavaleta Ramos, "Stacking ensemble approach to diagnosing the disease of diabetes," *Informatics Med. Unlocked*, vol. 44, 2024, doi: 10.1016/j.imu.2023.101427.

[32] M. Aqil *et al.*, "Deep learning based stacking ensembles for tropical sorghum classification," *J. Agric. Food Res.*, vol. 21, 101931, 2025, doi: 10.1016/j.jafr.2025.101931.

[33] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, "A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients," *Decis. Anal. J.*, vol. 7, 100242, 2023, doi: 10.1016/j.dajour.2023.100242.

[34] A. Asil, H. Alipour, S. Mojtahedzadeh, and H. Asil, "Ensemble learning of Ada-boosting Based on Deep Weighting for Classification of Hand-written Numbers in Persian (With the doctors' (With the doctors' prescription approach)," *J. Inf. Syst. Telecommun.*, vol. 12, no. 2, pp. 162–169, 2024, doi: 10.61186/jist.41053.12.46.162.