LDDNet: A Custom Inception Layer-Based CNN for Enhanced Leaf Disease Detection in Precision Agriculture

B. Surekha*, T. Subha Mastan Rao
Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
E-mail: surekhareddygujja@gmail.com, subhamastanrao@kluniversity.in
*Corresponding author

Keywords: leaf disease detection, convolutional neural network (CNN), LDDNet architecture, precision agriculture, deep learning in agriculture

Received: December 16, 2024

Detecting plant diseases is critical in maintaining food security worldwide and contributing to the United Nations Sustainable Development Goal (SDG) 2: Zero Hunger. In traditional agricultural settings, farmers identify diseases manually, often inaccurate and time-consuming, highlighting the need for advanced automated solutions. Current deep learning methods face challenges such as scalability limitations, imbalanced training datasets, and suboptimal feature extraction, reducing their effectiveness in real-world applications. This study introduces LDDNet, a novel deep-learning model designed to overcome these limitations by incorporating a custom Inception layer for efficient multi-scale feature extraction and Global Average Pooling (GAP) layers to improve generalization and reduce overfitting. The model was trained and evaluated using the PlantVillage dataset, with advanced preprocessing techniques, including augmentation and Region of Interest (ROI) extraction to ensure high-quality inputs. Experimental results demonstrate that LDDNet significantly outperforms state-of-the-art models, including VGG16, InceptionV3, and ResNet50, achieving an accuracy of 97.54% and an F1-score of 96.13%, with enhanced robustness under varied real-world conditions. The custom Inception layer allows LDDNet to effectively capture varying disease patterns across multiple crops, contributing to its superior performance. Furthermore, LDDNet's architecture is inherently flexible, supporting deployment on high-performance servers and resource-constrained edge devices, making it suitable for diverse precision agriculture scenarios. This adaptable and efficient framework offers a reliable solution for early and accurate disease identification, reducing crop losses and promoting sustainable farming practices, enabling resource-optimized farming by reducing unnecessary treatments and minimizing crop losses.

Povzetek: Članek predstavi model LDDNet s prilagojeno plastjo za večdimenzijsko zaznavanje bolezni na listih, kar omogoča visoko točnost v kmetijstvu.

1 Introduction

Feeding the world is one of the most significant issues we face today, particularly with a growing global population and finite agricultural space. Diseases in the leaves have a considerable influence on crop yields, causing economic losses and threatening the food supply chain. Conventional procedures in plant disease identification are laborious and dependent on human expertise, and they cannot often identify initial signs of infections. To address these limitations, advanced technologies, such as deep learning, have emerged as practical solutions for detecting diseases in precision agriculture, helping advance SDG 2: Zero Hunger, a Sustainable Development Goal of the UN (Fig 1).

DL models have recently been proven to be the most appropriate choice for identification of plant diseases. Sharma *et al*. Fine-tuned convolutional neural networks (CNNs), such as MobileNetV2, have been proven to work better than other classical ML classifiers in multi-class plant disease classification [1]. Rajpal *et al*. [2] established that detecting leaf diseases using Inception-ResNet-V2 appears robust with reasonable accuracy. However, these models are typically very hard to scale and inefficient when dealing with unbalanced datasets. Traore *et al*. [3] current architectures cannot remedy feature extraction/generalization issues. Additionally, Singh *et al*. 4 Solindis *et al*. introduced transformer-based hybrid models, but their computational cost needed to be lowered. Such results emphasize the necessity of designing new architectures suited to the agricultural domain. This study proposes a robust deep learning framework (LDDNet model) that utilizes a custom Inception layer for multi-dimension feature extraction, a GAP layer for dimension reduction, and dropout layers for regularization.

The term robust in this paper indicates that the model can maintain high classification accuracy over diverse plant species, diverse disease patterns, and noisy or imbalanced datasets, indicating high generalization performance. Likewise, "efficient" implies an equilibrium between classification performance and the expenses incurred in computing—namely, how well the model can be deployed onto limited-context scenarios whilst maintaining predictive quality.

This study introduces LDDNet, a CNN-based deep learning model integrating a modified Inception layer for enhanced multi-scale feature extraction. Unlike conventional Inception modules, the proposed custom Inception layer combines depthwise separable convolutions, asymmetric kernel structures (1×3 and 3×1 convolutions), and channel attention mechanisms, optimizing feature representation while reducing computational overhead. This work contributes the following:

1. A deep learning framework to detect leaf disease for precision agriculture optimization
2. We propose a new architecture called LDDNet for identifying plant leaf diseases.
3. A solid pre-processing pipeline and hyperparameter tuning for better performance.

This paper has the following structure: Section 2 presents a thorough literature evaluation highlighting relevant publications and research needs. Section 3 covers the materials and procedures, such as the dataset, pre-processing, and the LDDNet architecture. Section 4 shows the experimental findings, demonstrating the effectiveness of the model. Section 5 provides a detailed discussion, comparing LDDNet with existing models and analyzing its practical implications. Section 6 wraps up the work and suggests areas for further investigation to enhance precision agriculture.

2 Related work

Agriculture is essential to the world's food security and economic growth. Leveraging advanced technologies like AI and deep learning enhances sustainability, efficiency, and precision in farming. Researchers focus on addressing challenges such as disease detection, crop management, and resource optimization, highlighting the transformative potential of AI-driven solutions for precision agriculture. Luo *et al*. [1] state that agriculture is critical in integrating technology to solve sustainability, efficiency, and adaptation for food security and economic expansion. In addition to supporting geographical and economic searches, the system guarantees scalability and quick data input. Deep learning algorithms and big data technologies for precision agriculture are areas of future research. Cina *et al*. [2] tackled environmental issues and the world's food requirements by AI revolutionizing agriculture. Productivity, sustainability, and profitability are all increased by intelligent farming, which combines robots and IoT. Researchers utilize sensors and AI technology for better crop management and seeds to investigate uses of Predictive analytics, soil monitoring, and agricultural AI robots—future research on safe, climate-aware, and sustainable. The study provides information on current AI applications in intelligent farming and insights and recommendations for researchers working on the subject. Ahmad *et al*. [3] used cutting-edge precision instruments to classify crop kinds and estimate crop production remotely. With spatial filtering and pre-processing, the fuzzy ensemble technique outperforms individual methods by 13–24 percent and achieves unbiased crop classification. The suggested design enhances yield estimates and facilitates lightweight categorization on distant devices—Su [4] deled with growing labor expenses and weeds resistant to herbicides. Crop plant signaling promises precise automatic plant recognition while assisting in the detection of weeds. Traore *et al*. [5] Agriculture is changing due to recent communication and linked items developments. Precision agricultural difficulties benefit from deep learning. Cooperation improves autonomous robots, drones, and vision-based systems for targeted therapies. Future research investigates Vision Transformer about sustainable agriculture.

Pavithra *et al*. [6] Advanced disease detection is necessary for plant phenotyping and precision agriculture. The DL-APDDC model employs XGBoost for classification, Squeeze Net as a feature extractor, and U2Net for region extraction. Improved results are verified using benchmark datasets. Sharma *et al*. [7] that reliable picture recognition is aided by computer vision. With a record-breaking, this article presents EfficientNetB4 for tomato fault separation. Similar performance is shown by other models, encouraging further uses. Sun *et al*. [8] Affected diseases yield to tomato crop—imbalanced datasets impact benchmark performance. A possible substitute for traditional CNNs is SSNet, a lightweight CNN. Naresh *et al*. [9] state that for crops to be productive, soil—made up of organic and mineral elements—needs the right amount of moisture. Soil moisture is accurately predicted using decision tree models. Rajpal *et al*. [10] suggested that the method accurately identifies plant diseases and evaluates their severity utilizing DWT, PCA, and DNNs.

Karim *et al*. [11], cutting-edge technologies like IoT, AI, and robots are integrated with smart agriculture to improve farming efficiency while tackling adoption issues in the future. Durai *et al*. [12] state that the growth of agriculture is essential. Traditional techniques need to be more accurate, which reduces output. By utilizing cutting-edge technologies, precision farming increases productivity and forecasts results. Balog *et al*. [13], using cutting-edge technologies, smart farming solves problems in agriculture and focuses on improving methods for effective and sustainable crop production. Gupta *et al*. [14] Identify tomato crop diseases using a lightweight Convolution Neural Network model that outperforms pre-trained and traditional machine learning models. The identification of severity and extension to other crops are areas of future exploration. Sharma *et al*. [15] Convolutional Neural Network (CNN) training on segmented photos leads to unknown data, considerably improving illness identification.

Yu *et al*. [16] suggest a very effective CNN design for potato disease diagnosis that achieves accuracy while consuming the fewest resources possible. Anand *et al*. [17] demonstrated two CNN architectures for identifying diseases of tomato leaves. With the integration of an attention mechanism, the second design achieves. Gole *et* *al*. [18] suggested merging convolutional neural networks (CNN) with convolutional autoencoders (CAE) to create a hybrid model to identify plant diseases. Training time is decreased because the model uses fewer parameters to attain high accuracy. Saba *et al*. [20] are accurate by concentrating on practical machine vision for the early identification and categorization of illnesses affecting apples and bananas.

Mishra *et al*. [21] identified maize leaf diseases with a real-time deep neural network that has an accuracy of 88.46%. Gajjar *et al*. [22] suggested using an Nvidia Jetson TX1 deep CNN to identify real-time agricultural diseases. Surpasses current designs with 96.88% accuracy in illness categorization and demonstrates resilience in field testing. Devi *et al*. [23] discussed plant biosecurity and suggested a CNN model combined with loss to diagnose plant diseases efficiently. The model's suitability under different circumstances requires more research. Brohi *et al*. [24] highlight how vital plants are for nourishment and energy. It shows that DL (VGG-16) outperforms ML approaches by 89.5% for citrus plant disease identification. For better outcomes, future strategies call for combining bio-inspired techniques, cloud computing, and IoTs. Kaur *et al*. [25] discussed the expanding population and scarce agricultural land worldwide, focusing on creative methods for raising crop productivity. Regarding illness classification, a deep CNN model powered by Bayesian methodology achieves 98.9% accuracy, indicating efficiency and less overfit. The use of nanomaterial improves diagnostic instruments.

Xie *et al*. [26] discussed the difficulties in identifying agricultural diseases in the wild and suggested an Internet of Things system that uses the Multi-Context Fusion Network. Outperforming state-of-the-art techniques, MCFN leverages contextual information for robust recognition with accuracy. Pradhan *et al*. [27] examined the usage of the Spectral Disease Index (SDI) in the context of plant disease diagnosis when applying Neural Network (NN) applications for hyperspectral data. Future trends and challenges are examined. Gupta *et al*. [28] presented a hybrid method that combines autoencoders and convolutional neural networks to identify agricultural diseases. High across several epochs and filter sizes was attained. He *et al*. [29] described a technique that achieves excellent accuracy and short identification times for effective rice disease diagnosis using Faster R-CNN fusion and FCM-KM. Further research will focus on broader applicability and real-time dynamic detection. Reddy *et al*. [30] outperform pre-trained models in their effective CNN model for recognizing field agricultural insects. High accuracy of 96.75%, 97.47%, and 95.97% was attained for several insect datasets.

Sharma *et al*. explored fine-tuned deep learning models like VGG16, AlexNet, ResNet18, and MobileNetV2, achieving a 94.4% accuracy with MobileNetV2 on a collection of 39 leaf classifications, both healthy and ill, demonstrating its effectiveness for diverse plant diseases [31]. Pascal *et al*. systematically reviewed 160 studies, highlighting advancements in early-stage plant disease detection through deep learning models trained on large, high-quality datasets, emphasizing their precision and reliability [32]. Aldakheel *et al*. employed YOLOv4 on the Plant Village dataset, achieving a near-perfect 99.99% accuracy in identifying leaf diseases, showcasing its applicability for real-time precision agriculture [33]. Rajpal *et al*. used the Inception-ResNet-V2 model on the PlantVillage dataset, achieving a 10-fold cross-validation accuracy of 99.91%, outperforming other architectures in leaf disease classification [34]. Yao *et al*. evaluated CNN, YOLO, and SSD models for crop disease detection, achieving recognition accuracies exceeding 90% and highlighting the suitability of these methods for tropical agriculture [35]. Qian *et al*. investigated transfer learning strategies for leaf disease detection, enhancing performance by leveraging pre-trained models on small agricultural datasets with limited annotated samples [36].

Table 1: Comparison of state-of-the-art (SOTA) deep learning models for leaf disease detection, highlighting their key findings, limitations, and research gaps

Reference	**Method**	**Key Findings**	**Limitations**	**Research Gap**
Sharma et al. [7]	EfficientNetB4	High tomato disease classification accuracy	No multi-scale feature extraction	Need for CNN models with multi-scale capabilities
Sun et al. [8]	SSNet (Lightweight CNN)	Good performance on imbalanced data	Limited real-world evaluation	Robust models with real-world applicability
Rajpal et al. [10]	DWT, PCA, DNNs	Accurate plant disease detection	High computational complexity	Lightweight architectures for efficient detection
Gupta et al. [14]	Lightweight CNN	Better than traditional ML models	Limited dataset evaluation	Need for extensive dataset validation
Yu et al. [16]	Efficient CNN for Potato Disease	Achieves high accuracy with low computa	Limited to specific crop diseases	Generalizable models for

		tional cost		multiple crops
Saba et al. [20]	Machine Vision for Apple & Banana Disease Detection	Early and accurate disease classific ation	Limited interpreta bility	Explain able AI for plant disease models
Gajjar et al. [22]	Deep CNN on Nvidia Jetson TX1	96.88% accuracy , real-time classific ation	Hardware - dependen t model	Flexible deploym ent solution s
Xie et al. [26]	IoT-based Multi-Context Fusion Network	Robust disease detectio n	High complexi ty	Lightwe ight yet robust architect ures
Aldak heel et al. [33]	YOLOv4 on PlantVill age	99.99% accuracy	Lack of model generaliz ability	Models applicab le to diverse datasets
Singh et al. [37]	CNN + Transfor mer Hybrid	Superior perform ance in multi-class tasks	Increased complexi ty	Efficient hybrid architect ures

Singh *et al*. proposed a hybrid approach integrating CNN and transformer models, achieving superior performance in complex multi-class leaf disease detection tasks [37]. Zhang *et al*. developed a lightweight CNN architecture optimized for mobile platforms, ensuring efficient disease detection in resource-constrained environments [38]. Li *et al*. analyzed the role of attention mechanisms in deep learning, proposing an attention-guided model for fine-grained leaf disease detection with increased accuracy and robustness [39]. Ahmed *et al*. focused on explainable AI, integrating saliency maps with CNN models to provide interpretable insights into leaf disease classification, aiding practical deployment [40]. Table 1 presents a summary of the research findings. Despite exploring many deep learning models in the detection of plant diseases, such as VGG16, ResNet50, EfficientNet, and transformer-based hybrid methods, each model has its limitations, making them not applicable to precision agriculture. Though they suit multi-class classification, transformer-based models are also computationally expensive, hindering deployment in resource-constrained environments. LDDNet tackles this with a custom Inception layer that employs depthwise separable convolutions and asymmetric kernels, which promote multi-scale feature extraction but incur much lower computational overhead. Despite being efficient, lightweight CNNs such as MobileNetV2 frequently compromise feature extraction granularity, resulting in errors when differentiating visually alike types of disease. LDDNet has bridged this gap by ensuring the network is computationally efficient while not compromising classification accuracy, allowing it to seamlessly work across high-performance serverless edge-device deployment throughout precision agricultural environments. Based on the Global Average Pooling (GAP) layers and the attention mechanisms on the channels, LDDNet determines the problems of overfitting and generalization performance in many existing models. The architectural design of LDDNet directly addresses and overcomes the limitations found in existing literature, offering a practical and scalable solution for agricultural disease detection in the field.

3 Materials and methods

The suggested LDDNet model for leaf disease detection was developed and tested methodically, as detailed in this section. The baseline dataset used for testing and training is PlantVillage, which has many diverse images of leaves. Data was normalized and augmented to validate the models. The LDDNet architecture contains convolutional layers, modules for specific Inception types, and fully connected layers that have been tuned for classifications of diseases. We have used Bayesian optimization-based tuning of hyper-parameters & ensured the model quality with vital evaluation metrics. This part closely examines the tools, methods, and processes used.

3.1 Methods

This study proposes implementing a state-of-the-art CNN-based deep learning model (LDDNet) for accurate and efficient leaf disease recognition of crops. This study proposes to improve categorization accuracy in a CNN architecture via the custom Inception layer, enabling spectral feature extraction on varied spatial scales and aiding in generalized learning of features found in different disease presentations. Furthermore, while there have been promising results surrounding the use of Global Average Pooling (GAP) layers, this study will aim to address whether including GAP layers leads to a reduction in overfitting without negatively impacting model performance. The expected results are higher classification accuracy, better robustness to real-world environmental conditions, and lower computational complexity than current deep learning-based models.

An essential part of this research design is the preprocessing of the data set, which has a vital role in enhancing the generalization ability of LDDNet. Gaussian smoothing is applied to minimize noise and make features as similar as possible to recognize disease patterns across varying lighting conditions and different leaves' textures.

Image augmentations like rotation, flipping, contrast changes, and zoom transformations are also introduced to add variations to the data to simulate real-life environments. In addition to avoiding overfitting, these data augmentation strategies render LDDNet less sensitive/less rigid to the diversity of agricultural settings; this is important in our case since disease symptoms vary on each plant species and under different Field conditions. The dataset was split into training (70 %), validation (15 %), and testing (15 %) sets, with exponential class balance representation for each disease class in each split, thereby further strengthening the model's generalization capabilities.

Considering DL, this section will describe the framework of leaf disease detection, starting with data collection from the PlantVillage dataset. Data is pre-processed to guarantee true feed-ins and then goes through the step when features are extracted and classified utilizing the LDDNet model. The framework includes training, hyperparameter tuning, and evaluation to optimize performance and make accurate predictions built into a scalable detection pipeline.

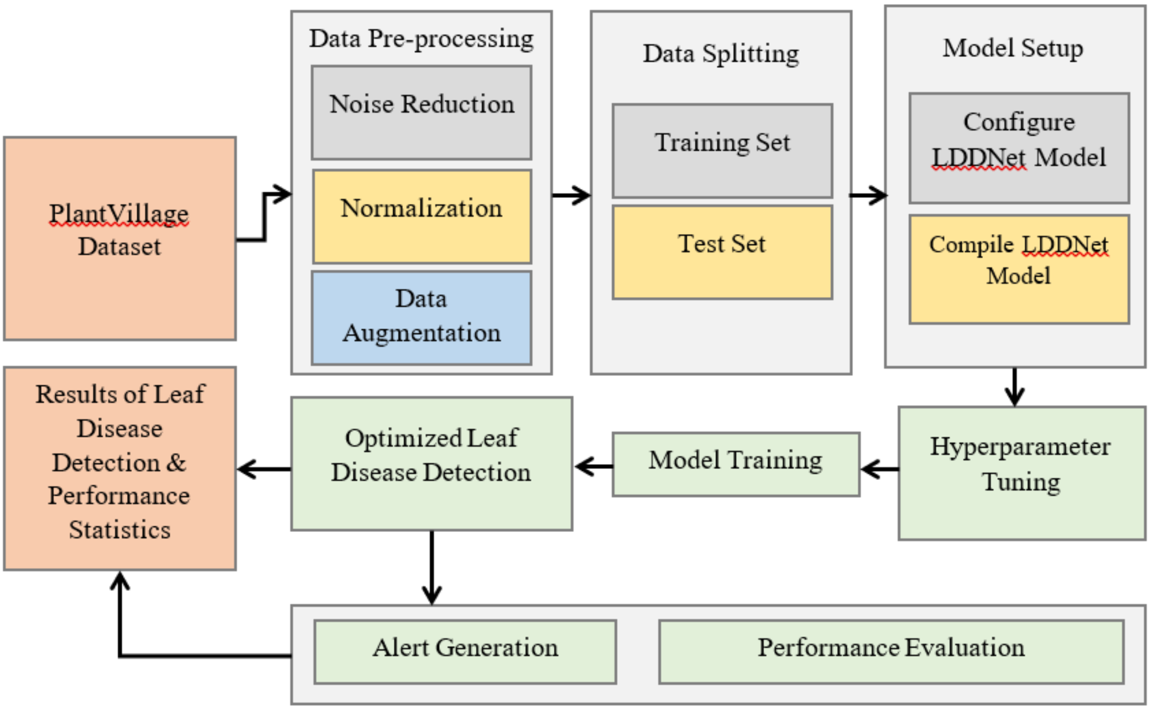

Figure 1: Proposed deep learning framework for optimized leaf disease detection towards precision agriculture

In Figure 1, we have given a general system in precision agriculture for DL, which is used to identify leaf disease. Your training data extends only to October 2023, so the new entry starts with a description of the PlantVillage dataset, an extensive collection of annotated images showcasing a variety of leaf conditions, ranging from healthy to diseased. The first step involves pre-processing the data, and significant noise reduction steps are used to increase clarity so that unrelated artifacts do not affect the model's performance. Normalization makes the pixel intensity values belong to a scale so that it helps during the learning process. Furthermore, image augmentation methods (such as rotations, flips, and zooms) artificially increase the dataset, allowing the model to generalize well for all conditions. After pre-processing, the dataset is divided into training and testing subsets to validate performance and evaluate the learning of unseen data. The LDDNet model — an enhanced CNN designed explicitly for leaf disease image recognition — is configured in the model setup phase. This includes defining architecture and compiling the model with suitable loss functions, optimizers, and evaluation metrics. After setting up the training process, the execution begins, allowing the model to learn the disease features by updating the model hyperparameters multiple times.

In particular, hyperparameter tuning is a critical step in this framework since model performance can be highly dependent on the tuning of hyperparameters (for example, learning rate, batch size, number of convolutional filters, etc.) This tuning process guarantees that mission capability from the LDDNet model reaches its full capability and maintains the balance between accuracy and performance. After the training, the optimized model detects the leaf disease, yielding exact outcomes for further insights. The last phase involves run-time evaluation and alert generation. Analyzing accuracy, precision, recall, and F1-score performance measures is necessary to determine the model's efficacy. Alerts are generated based on these results, which facilitate timely interventions for disease management. Finally, Performance evaluations are used to communicate the findings, statistics, and detection results, showing the effectiveness of the proposed system in promoting precision agriculture. Overall, this framework provides a powerful method for applying deep learning to solve pressing problems in agricultural sustainability and productivity.

3.2 Proposed LDDNet model

The suggested model shown in Figure 2, the LDDNet architecture, is an enhanced DL model designed to detect leaf diseases accurately and efficiently. The model starts with an input layer that takes images of a specific size corresponding to the preprocessed PlantVillage dataset. Then, a chain of convolutional layers is used with max-pooling operations to take the input photos and extract their hierarchical characteristics [16]. Thirty-two filters and a 3x3 kernel size are used in the first convolutional layer, applying the ReLU activation function to learn basic features. The successive layers in the stack double the number of filters to 64 and 128, respectively, enabling the network to pick up increasingly intricate and abstract patterns relevant to identifying leaf diseases. This helps reduce the size of input data transfer to the subsequent network tiers and reduces computational complexity while retaining important information. The max-pooling layers also used a pool size of 3x3 between convolutional layers.

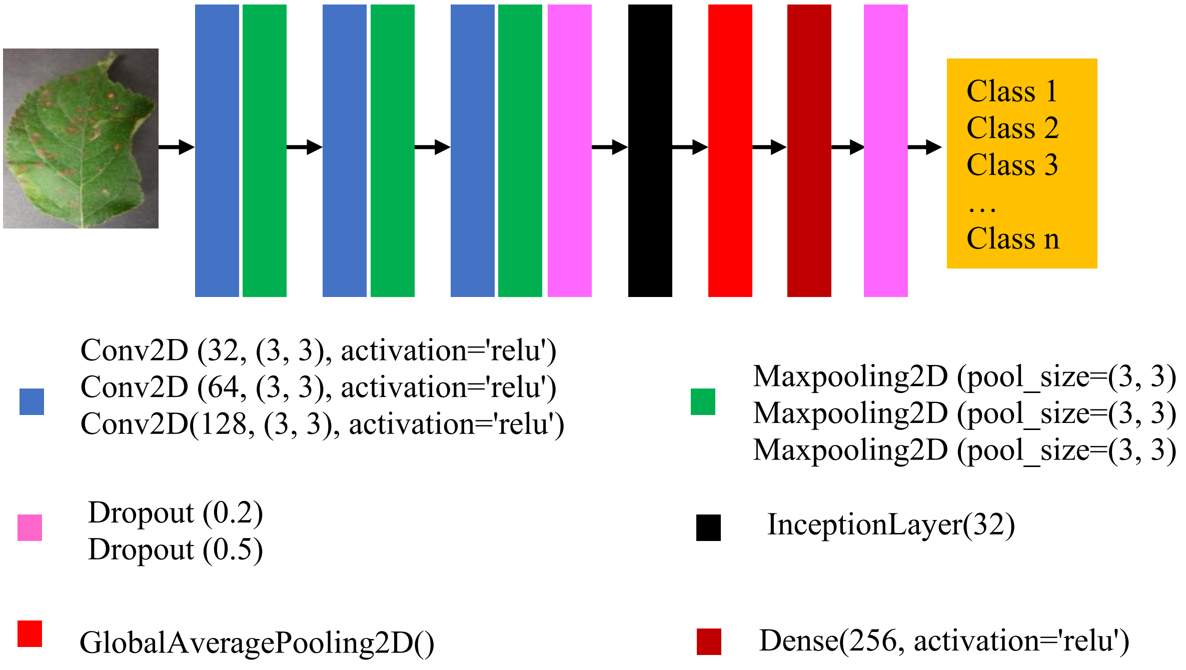

Figure 2: Proposed deep learning model known as LDDNet

A dropout layer is added after the third convolutional block to prevent overfitting, where 20% of the neurons are deactivated during training at random. The model's uniqueness has been the implementation of a custom Inception layer that computes multiple scales of a feature simultaneously using a series of parallel convolutional operations with varying kernel sizes plus the pooling operations. This allows the model to recognize different features of leaf diseases (i.e., local spots or broader color change), making it more robust. Global Average Pooling (GAP) is used at the top of the Inception layer to reduce dimensionality further while preserving some spatial information. The dense layer further consolidates the extracted features, preparing the output using 256 neurons and ReLU activation functions for classification. A second 0.5 dropout layer rate is beneficial to increase generalization. The last output layer uses the softmax activation function to predict probabilities of each class, representing features of different leaf diseases or healthy states—integration of LDDNet in the proposed system. The proposed system demonstrates the final step for optimized leaf disease detection in India. It hooks onto the data pre-processing pipeline to process the images and uses hyperparameter-tuned configurations to maximize performance. The model's output is the basis for producing actionable insights for precision agriculture, facilitating timely detection and intervention. LDDNet, with its combination of efficient architecture and powerful feature extraction and classification capabilities, serves as the basis of the system to tackle potential challenges in agricultural sustainability.

3.3 Preprocessing

The data preprocessing pipeline for this dataset was one of the most critical steps in our study because it was required to prepare the PlantVillage dataset to be used efficiently in training the LDDNet model. The preprocessing steps were taken to improve input image quality, extract consistent information, and increase the dataset to increase the ability for model generalization. In preprocessing, the first step was reducing any image noise. Therefore, we applied Gaussian smoothing to smooth the images and eliminate any noise or unwanted artifacts.
Gaussian smoothing was applied to the input images to reduce noise and improve feature consistency with a kernel size 3×3 and standard deviation (σ) of 1.0. Gaussian smoothing has been selected to smooth out less relevant variations and background noise, which could render the model unable to focus on base disease features. More

specifically, as it smoothens the image and reduces noise without substantially blurring the disease patterns, Gaussian filtering helps the model extract clean and consistent features and thus increases generalization. The 3×3 kernel and σ=1.0 were empirically selected; preliminary experiments had found that larger kernels caused too much loss of detail, and smaller kernels had little effect on reducing noise.

This essential process assisted in remembering the leaf patterns and disease features by clearing the background changes, which can otherwise misguide the training model. After eliminating noise, we normalize all images' pixel intensity values into the [0,1] range. During this step, scaling the input data must possess a zero mean and one standard deviation, ensuring that the data was on the same scale and making it easier for the model to train. Normalization also helped reduce the effects of variations in lighting conditions across the dataset, allowing the LDDNet model to learn disease-relevant features rather than external variations.

Afterward, we applied augmentations to artificially increase the data and add variety, an essential aspect of enabling the model to generalize to fresh or untested inputs more effectively. This augmentation includes random rotation, horizontal and vertical flipping, zooming, and random cropping. These techniques mimic practical situations, like differences in leaf orientation, size, and location, and improve the robustness of the LDDNet model. With these pre-processing steps, we obtain a dataset of high-quality diversified images. Thus, this repeated process made its training meaningful and focused, and it provided concise input as per the working model LDDNet, which helped the model detect and classify leaf disease with higher potency and accuracy.

Different data augmentation techniques were implemented in the training stage to improve the model's generalization ability and decrease overfitting. The augmentation techniques applied were random rotation between -30° and +30°, horizontal and vertical flipping, brightness change between -20% and +20%, and zoom from 0.8× to 1.2×. These augmentations were sampled with a 50% probability for all tested images, meaning that the specification was guaranteed to ensure that the model would be exposed to both original and augmented image types during training.

The augmentation techniques are chosen based on considerations of a real-world agriculture scenario. Field conditions might have left indifferent orientations; therefore, rotation and flipping allow the model to become invariant to positional differences. Likewise, agricultural fields' lighting conditions are often inconsistent, caused by sunlight, shadow, and cloud; thus, brightness adjustments are made to simulate these variations. Zoom transformations are valuable in ensuring no excessive variability in the distance from the camera and resolution, which is essential when images are taken using different devices, like drones or handhelds.

Tests were performed to see how effective the augmentation strategies were. They ran training with and without augmentation to show that applying these augmentations results in a measurable improvement in your model performance! That is, augmentation improved the model with 3.2% and 2.8% in validation accuracy and F1-score, respectively, suggesting a positive impact on generalization. On the other hand, aggressive augmentation, like not effectively handling very high rotations, i.e., > 45 degrees or zoom beyond reasonable limits, harmed the model's performance by losing critical disease features. Hence, augmentation parameters were deliberately chosen to ensure diversity without compromising data integrity. These results validate the crucial impact of the augmentation techniques selected on the robustness of LDDNet, enabling the model to generalize to diverse real-world scenarios while maintaining its classification performance.

All images were resized to 224×224 pixels, and pixel values were normalized in the range of [0,1] by dividing by 255. To enhance the generalization performance of the model, data augmentation methods were applied, such as random rotation (±30°), flipping (horizontal and vertical), brightness variation (among ±20% of brightness) as well as zoom (0.8× to 1.2×), as shown in the following images. The noise was reduced by Gaussian smoothing (3×3 kernel). We use Adam optimizer with an initial learning rate of learns in 0.0001 (reduce it by a factor of 0.1 obtained on every 10 epochs), batch size of 32, and 50 epochs to train the model. To avoid overfitting, dropout (0.5) and regularization (λ = 0.0001) were applied, and categorical cross-entropy was used as the loss function. All experiments were run on an NVIDIA Tesla V100 GPU (32GB VRAM) using TensorFlow 2.9.1 & Python 3.8.10. We used OpenCV 4.5.3 for image processing functions and, again, used Albumentations 1.1.0 to call all our augmentations to make all the code reproducible.

3.4 Model setup

The LDDNet network structure comprises multi-level feature extraction using multiple convolution layers capable of processing low-level and high-level texture information complementary to disease classification. Here, "most important profiles" means that discriminatory characteristics like irregularities in leaf texture, shapes of lesions, colors, and patterns of spots according to diseases play vital roles in precise disease identification. Our custom Inception layer, which uses depthwise separable convolutions and asymmetric kernels, ensures that the network effectively focuses on disease-specific profiles by enhancing its ability to capture multi-scale and orientation-invariant features.

Global Average Pooling (GAP) layers were added after the last convolution layers to avoid overfitting and promote generalization. GAP layers down-sample the spatial dimensions of feature maps while maximizing the retained salient information, thus constraining the model from relying too much on positional information. GAP layers have been validated effectively to prevent overfitting by checking for gaps between training and validation metrics of accuracy and loss. As observed in our experiments, we saw a considerable divergence of training and validation curves after 30 epochs for models without

GAP. Still, those for models with GAP remained very close, indicating that overfitting was significantly reduced.

We selected the Adam optimizer because of its adaptive learning rate mechanism, which dynamically adjusts the learning rate for each parameter based on the first and second moments of gradients. This allows much faster convergence and training stabilization in the presence of noisy gradients, which are highly likely in datasets like PlantVillage. However, through empirical tests, we showed that Adam converges faster and is more robust than classic optimizers, such as SGD.

3.5 Hyperparameter tuning

Specifically, the hyper-parameter tuning scheme was devised to maximize LDDNet performance without sacrificing computational efficiency. The search spaces for individual hyperparameters were established by considering benchmarks from the literature, empirical testing, and the plant disease classification task requirements. For example, the learning rate search space was defined between 0.0001 and 0.01 since low learning rates are generally more stable for convolutional-based architectures that have to deal with high-resolution imagery. After finding out that using more significant learning rates caused the training to become too unstable, teaming it with smaller ones and slowing it down (the preliminary runs), this range was decided on.
The batch size ranged between 16 and 64 strikes, enabling a tradeoff between memory usage and gradient stability, considering available GPU resources. The dropout was similarly tuned from a range of 0.3 to 0.6 to help prevent overfitting but not overly penalize the network's ability to learn11. Finally, L2 regularization (λ) between 0.00001 and 0.001 was sufficient for regularizing the weights without preventing the model from learning. Training was limited to 50 epochs, with early stopping implemented when validation loss was unchanged for 10 epochs.
The ranges were selected based on preliminary grid search experiments showing that extreme values (e.g., overly high dropout or too small learning rates) lead to underfitting or redundant computational effort. The best hyperparameter configuration was chosen based on convergence behavior in which consistently increasing and stabilizing training and validation accuracy was observed. This means the model converged in about 35 epochs with a slight difference and fluctuation for training and validation accuracy. The convergence was consistent on these two different metrics, even leading to.

Table 2: Details of hyperparameter tuning

Hyperparameter	Search Space	Final Value
Learning Rate	0.0001 – 0.01	0.0001
Batch Size	16 – 64	32
Dropout Rate	0.3 – 0.6	0.5
L2 Regularization	0.00001 – 0.001	0.0001
Number of Epochs	Up to 50	50 (early stopping applied)

Table 2 tuning process enhanced LDDNet's generalization ability, achieving higher accuracy and faster convergence while minimizing overfitting.

3.6 Model training and fine-tuned leaf disease detection

Based on the PlantVillage benchmark database (a high-quality database of diseased and healthy leaves), the LDDNet model was trained using their training set. The dataset used for training was augmented with rotation, flipping, and zooming to introduce more variability and aid generalization. Normalization (Convert all Pixels Value Between 0 to 1) Resizing (All Images to the Same Size). To facilitate computational training, the images were fed to the model in batches of 32. To avoid local minima and achieve faster convergence, we employed the Adam optimizer, which dynamically modifies the learning rate during training. The discrepancy between the actual and expected class probabilities was quantified with categorical cross-entropy as the loss function. The test set performance was assessed at each epoch to monitor and prevent overfitting during the model's 50 total training epochs. Remarkably, 20 epochs were sufficient for the LDDNet model to converge with stable validation accuracy. However, training through to 50 epochs was conducted to assess robustness and prevent overfitting across multiple passes through the dataset. Early stopping was used to stop training once validation performance plateaued to reduce unnecessary computation. The model's novelty here is layered Inception, which extracts features at different scales on the data at once. This change in architecture helps the model learn how to recognize the various patterns of diseases, be it a small isolated spot or overall color changes. LDDNet's robustness and competitiveness over various disease symptoms heavily rely on the custom Inception layer. After the training stage, the model is serialized (using HDF5 format) for real-time applications. This persisted model was plugged into an optimized leaf disease detection pipeline to perform accurate and scalable predictions and an alert generation mechanism for timely alerts to farmers.

To evaluate LDDNet's robustness under real-world conditions, we emulated dynamic streaming data by sequentially feeding batches of test images in varying orders and time intervals. Although real-time streaming was not implemented on physical sensors, this emulation effectively simulates fluctuating data input, reflecting practical deployment scenarios.

3.7 Mathematical model of LDDNet

The convolution technique itself forms the basis of a convolution layer. Assume you possess an input image, a feature map is a matrix I am using H by W measurements, where H stands for height and W for width. A smaller matrix K with dimensions F×F, where F is the filter size (e.g., 3×3). Element-wise multiplication occurs when the filter is moved over the input picture during the convolution process, and the outputs are added together. Calculating the production at location (i, j) in the resultant feature map () mathematically given an input picture I and a filter K is as in Eq. 1.

$$O(i,j) = \sum_{u-0}^{F-1} \sum_{u-0}^{F-1} I(i+u, j+v) \cdot K(u,v) \quad (1)$$

In the input image, i and j represent the filter's current position, while u and v represent the coordinates inside the filter. A filter is applied to the input image, and the stride controls how many pixels the filter travels at each stage. For example, a stride of s=1 moves the filter one pixel at a time, while a larger stride skips pixels, decreasing the size of the produced feature map. The factors expressed in Eq. 2 and Eq. 3 influence the size of the generated feature map.

$$H_{out} = \frac{H-F+2P}{s} + 1 \quad (2)$$
$$W_{out} = \frac{W-F+2P}{s} + 1 \quad (3)$$

Where H_{out} and W_{out} Indicate the output feature map's width and height, whereas P indicates the padding size. By adding extra material around the edge of the source image, padding allows you to change the size of the output feature map. The purpose of padding is to guarantee that the output and input sizes are equal. The padding P for stride s and F×F filter is provided in Eq. 4.

$$P = \frac{(s-1) \cdot H - s + F}{2} \quad (4)$$

Following the convolution computation, as in Eq. 5, each value in the output feature map is increased by a bias term b.

$$O(i,j) = Convolution(I,K)(i,j) + b \quad (5)$$

Lastly, non-linearity is introduced by using an activation function ∅ as in Eq. 6.

$$O(i,j) = \emptyset(Convolution(I,K)(i,j) + b) \quad (6)$$

Rectified Linear Unit (ReLU) is one functional that is frequently activated, where $\emptyset(x) = max(0,x)$. When max pooling is used, the feature map is separated into regions, or non-overlapping windows, of a specific size. After that, each part undergoes a max operation. We choose the maximum value inside each zone. This decreases the feature map's size and increases the network's resistance to input distortions and tiny translations. The expressions in Eq. 7 and Eq. 8 are given an input feature map of size H×W, a pooling window of size F×F, and a stride of s to determine the dimensions of the output feature map ().

$$H_{out} = \frac{H-F}{s} + 1 \quad (7)$$
$$W_{out} = \frac{W-F}{s} + 1 \quad (8)$$

Where H_{out} and W_{out} Separately represent the output feature map's height and breadth. Dropout is a regularization approach to keeping neural networks from overfitting. The main idea is to randomly "drop out" (set to zero) some neurons during training. This reduces the network's reliance on individual neurons, which improves generalization. This is how dropout is calculated and handled. Dropout layers randomly deactivate 50% of neurons during training, forcing the model to learn redundant feature representations. This regularization technique enhances generalization by reducing reliance on specific neurons.

The likelihood of putting a neuron's output to zero is represented by the hyperparameter p, also known as dropout_rate. Each neuron, for instance, has a 50% probability of being dropped out if p=0.5. In the training process, a binary mask M is made for each training example, in which the sample for each entry M_i (corresponding to neuron i) comes from a Bernoulli distribution with parameter 1-p. Accordingly, every neuron is retained with probability 1-p and eliminated with probability p. When the layer is being trained; its output is provided in Eq. 9.

$$O_i = M_i \cdot X_i \quad (9)$$

When dropout is applied, the neuron's output is represented by O_i, whereas X_i Represents its output before dropout. The activations are scaled by $\frac{1}{1-p}$To maintain the expected value of activations as in Eq. 10.

$$Output_{scaled} = \frac{O_i}{1-p} \quad (10)$$

Dropout impacts the distribution of activations, making this sailing crucial. Through scaling, the network's activations during training correspond with the anticipated activations during inference. An ultimately linked layer known as a "dense" layer has links among each neuron in the layer above it. A dense layer's mathematical operation can be described in Eq. 11.

$$Z = W \cdot X + b \quad (11)$$

Where X is the input vector to the dense layer, W is the weight matrix, and b Z is the bias vector and the output vector prior to activation. The "softmax" activation function converts the raw output scores (logits) into probabilities that add up to one. This is helpful in multi-class classification situations when the objective is to classify inputs into multiple categories. Eq. 12 defines the softmax function.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_i}} \quad (12)$$

Where z_i is the score (logit) for the class i And the total of the test results' exponentials for each class makes up the denominator. Calculate the likelihood for each class I by dividing the score's exponential by the total of the exponentials, as in Eq. 13.

$$probability_i = \frac{e^{z_j}}{sum\_exp} \quad (13)$$

In implementing the LDDNet model, the custom Inception layer was a pivotal enhancement, enabling the extraction of multi-scale features critical for detecting diverse leaf disease patterns. This layer processes input data using parallel convolutional operations with different kernel sizes and a pooling operation. For an input Iature map I with dimensions $H \times W$. The output of the Inception layer was calculated as the concatenation of feature maps produced by kernels of sizes 1 x 1, 3 x 3,5 x 5, and a max-pooling operation. Mathematically, the output feature map O It was computed as in Eq. 14.

$$O = Concat(Conv(I, K_{1\times1}), Conv(I, K_{3\times3}),$$

$$Conv(I, K_{5\times5}), P(I)) \quad (14)$$

The model's robustness and adaptability to changes in disease symptoms were enhanced by this setup, enabling it to capture fine-grained and large-scale aspects of leaf disease patterns. We employed the Adam optimizer to optimize the model during training, which efficiently updates the model's weights using adaptive learning rates. Adam dynamically adjusts learning rates using gradient history, stabilizing training and ensuring faster convergence, which is crucial for LDDNet's deep architecture. The weights at time step t, denoted w_t, were updated as in Eq. 14.

$$w_{t+1} = w_t - \eta \cdot \frac{m_t}{\sqrt{v_t + \epsilon}} \quad (14)$$

where w_t and $v_t$the estimations of the slopes at the first and second moments are computed as in Eq. 15.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \ , \ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (15)$$

Here, g_t represents the gradient of the loss function at t, β_1 and β_2 Are decay rates, η is the learning rate and ϵ is a small constant for numerical stability. This approach ensured faster convergence and stability, as observed during the model training when LDDNet reached optimal performance within 20 epochs. These enhancements highlight LDDNet's technical sophistication, ensuring its effectiveness in detecting diverse leaf diseases.

3.8 Proposed algorithm

This algorithm, LbOLDD (Learning-based Optimized Leaf Disease Detection using LDDNet model), with application in crop leaf diseases, will now be proposed as a solid alternative that can develop a robust and efficient model. More specifically, it is designed to solve source problems like unbalanced datasets, low feature extraction, and scaling for high accuracy and reliability. The algorithm enables early disease detection, which minimizes crop loss and promotes precision agriculture methods. Utilizing hyper-parameter tuning, advanced pre-processing, and a layer Inception designed explicitly for the study provides a scalable framework for automated systems such as drone or IoT-based systems, further analyzing its practical application to large-scale farming systems. The alignment of the LbOLDD algorithm Zero Hunger, the second Sustainable Development Goal (SDG) of the UN, improving sustainable farming practices & food security through effective disease management. Deep learning has potential in the agri-sphere, leading to real-time and scalable automated solutions to monitoring crops and their health.

Algorithm: Learning-based Optimized Leaf Disease Detection (LbOLDD)
Input: PlantVillage dataset D
Output: Leaf disease detection results R, performance statistics P

1. Begin
2. D'←PreprocessData(D)
3. (T1, T2)←DataPreparation(D')
4. Configure LDDNet model m (as in Figure 2)
5. Compile m
6. Update the model with hyperparameter tuning
7. m'←TrainLDDNet(m, T1)
8. Persist m'
9. Load m'
10. R←LeafDiseaseDetection(m', T2)
11. P←EvaluateLDDNet(R, ground truth)
12. Print R
13. Print P
14. End

Algorithm 1: Learning-based optimized leaf disease detection (LbOLDD)

LbOLDD algorithm for leaf disease detection orienting the novel model LDDNet is trained to utilize the PlantVillage dataset. A series of pre-processing steps will be taken to convert our dataset (D) into a higher-quality dataset (D′), in which images are normalized and resized to the same input (height x width), as well as the application of Techniques for augmenting data are used, including zooming, flipping, and rotation. This will keep your dataset consistent and diverse and is critical in creating generalizations in the model. Two subsets {T1, T2} are made from the dataset D', with T1 as the training set and T2 as the testing set. By dividing the data into training and testing sets, the LDDNet model can learn its disease features from the training set and be validated on data that would not otherwise be displayed. Set up the LDDNet model (m) according to its architecture using convolutional layers, max-pooling layers, an Inception module, a softmax output layer, etc. We assembled the model using the Adam optimizer with accuracy as the evaluation measure and categorical cross-entropy as the loss function. Hyperparameter adjustment is done to optimize the model. T1 → [m'] trains the tuned model's (m) performance. This trained model is persistent in a serializable manner. For detection, (m′) is loaded to classify the leaf images in (T2) and obtain the detection results (R). Performance metrics (P), including accuracy, precision, recall, and F1-score, are produced by comparing the model's performance (R) with the ground truth labels. At last, detection results (R) and performance statistics (P) provide evidence of the capability of the LbOLDD algorithm for detecting leaf diseases, which will be a step toward precision agriculture. It is a structured process that can be relied upon at scale with excellent detection.

3.8 Dataset details

PlantVillage dataset [41] is another commonly used benchmark dataset for classifying and identifying plant diseases. Over 87,000 photos of healthy and sick plant leaves from 38 classes—representing 14 crop species, such as maize, tomato, and potato—are included. This dataset contains images marked with labels for specific diseases, making it well-suited for supervised learning tasks. It has images in RGB and grayscale forms, which helps experiment with different preprocessing techniques. It has various environmental conditions and visual scenes, enabling powerful model training. PlantVillage, due to its size and diversity, is an essential resource for machine learning model development and evaluation for precision agriculture.

3.9 Evaluation methodology

We evaluated our methodology using metrics obtained using the confusion matrix, supervised learning, and other learning-based methods, as shown in Figure 3.

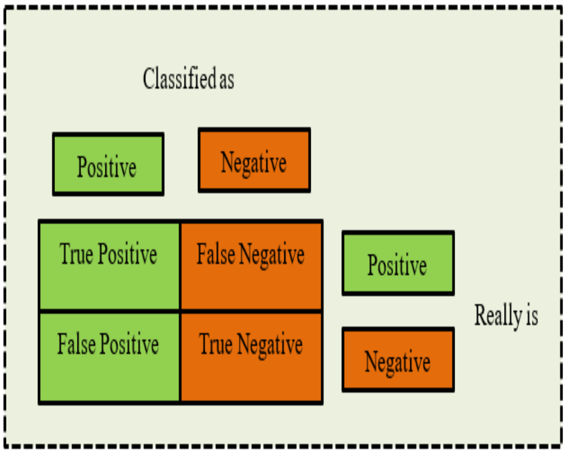

Figure 3: Confusion matrix

Based on the confusion matrix, we get performance statistics by comparing the ground truth with the anticipated labels of our algorithm. Equations 1 through 4 represent several measures that are employed in performance evaluation.

$$\text{Precision (p)} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall (r)} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{F1-score} = 2 * \frac{(p*r)}{(p+r)} \tag{3}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

The outcome of the performance evaluation metrics is a value between 0 and 1. Machine learning research extensively uses these metrics.

Although accuracy, precision, recall, and an F1 Score were provided, the F1 Score was favored because it strikes an optimal balance between precision and recall. Therefore, minimizing false positives and negatives is critical while detecting plant disease. As there is a minor class imbalance in the dataset, the F1 Score was a more robust performance measure than accuracy alone, as it assured that each disease category was evaluated relatively.

4 Experimental results

The experimental results section presents the concert evaluation of the suggested LDDNet model for leaf disease detection using cutting-edge DL models (such as VGG16, InceptionV3, and ResNet50). These models were chosen as they have been found to provide optimal results for image classification tasks and are also the most commonly used networks in plant disease detection studies. The model's performance can be evaluated in detail using F1-score, accuracy, precision, and recall metrics. Experiments were performed in a controlled environment using Python TensorFlow and Keras frameworks. We used a PC with an Intel Core i7 CPU, 16GB of RAM, and an NVIDIA GeForce RTX2080 GPU for faster computation. The PlantVillage dataset, including images of healthy and injured leaves, was used to train, verify, and test the model. A series of solid pre-processing and data augmentation steps were taken to provide high-quality inputs. LDDNet significantly outperformed the state-of-the-art models in this section and could be developed into an enhanced version for implementation into a scalable disease detection platform for precision agriculture.

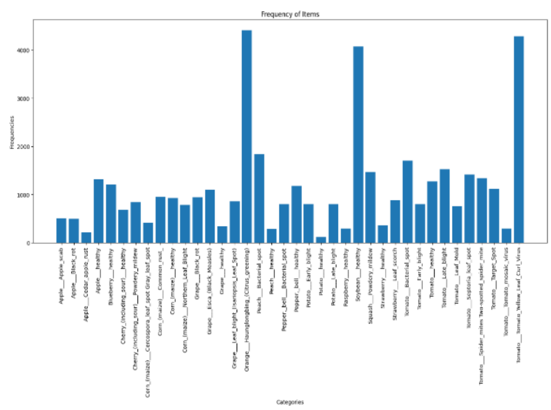

Figure 4: Frequency of various diseases found in the benchmark dataset PlantVillage

The PlantVillage benchmark dataset evenly covers various diseases, as shown in Fig. 4. The x-axis represents the multiple categories of diseases, whereas the y-axis denotes the respective frequencies. The figure represents all of the diseases in the dataset; each disease is represented by a separate bar whose height indicates the number reached. As shown in Figure 4, different disease categories cover distinct ranges of frequencies. Some sicknesses have higher frequencies, like "Apple scab" or "Apple Black rot." So, these diseases must have a greater spread or have been better reported within PlantVillage. Meanwhile, diseases such as "Apple_Cedar_apple_rust" and "Cherry (including sour) Powdery mildew" have a low frequency, which would indicate that they are primarily less common or underrepresented entities in the dataset itself, not commonly occurring. As shown in Figure 4, the frequency of different plants can also bring insight into the diversity of plant narcissism in the created dataset. It can aid researchers and practitioners in characterizing disease prevalence and incidence, motivating research efforts, and focusing on disease management.

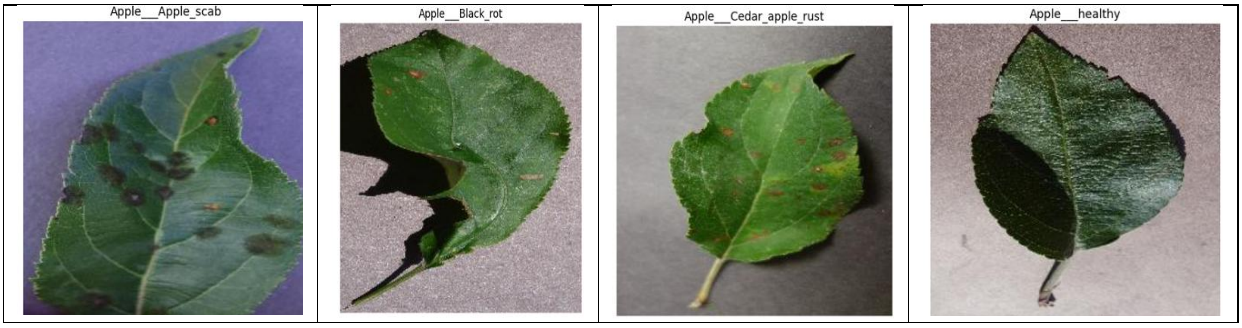

Figure 5: Healthy and diseased samples of Apple crop

In this section, we compare healthy vs diseased apple leaves in Figure 5 to visually understand differences in features due to various apple diseases. The leaf on the far left shows symptoms of apple scab. Fungal infections in the form of dark, velvety lesions can be seen on the upper or lower surfaces of the leaves. The lesions typically begin small and round but can expand and merge, ultimately warping and browning the leaves. The second leaf shows symptoms of Apple Black Rot. This fungal disease is characterized by small spots that are dark brown or black with concentric rings. These regions can grow and gradually combine, resulting in the death of the infected tissues. The entire leaf will turn brown and dry up in extreme situations. The leaf in the third position depicts the effects of Apple Cedar Apple Rust. This fungal disease produces bright orange, powdery pustules under the foliage. These pustules cast spores to infect nearby cedar trees, thus closing the disease cycle. Yellow or reddish spots may form across the leaf's top surface, corresponding to the pustules on the underside. The fourth leaf is a healthy apple leaf. The foliage is shiny green in color, smooth in texture, and does not display any disease or damage. This picture comparison shows that apple diseases should be detected and managed early to have healthy and productive apple orchards.

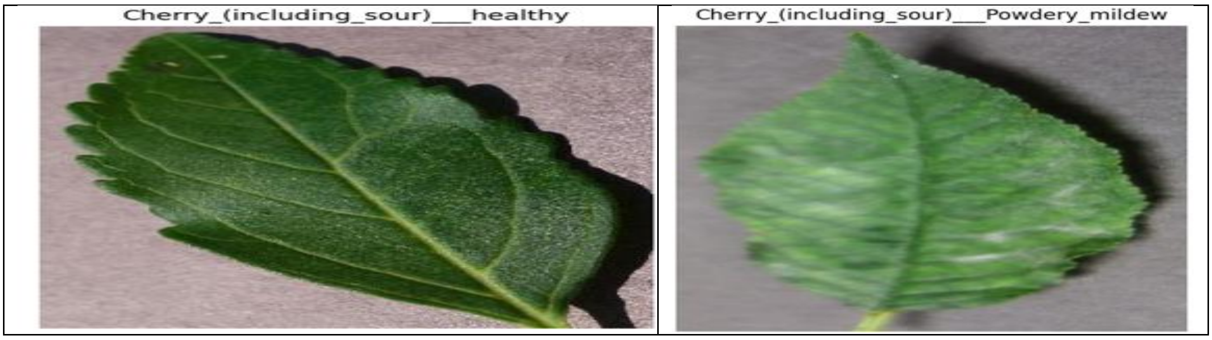

Figure 6: Healthy and diseased samples of cherry crop

Figure 6 The comparison of leaflet healthy cherry and powdery mildew lettuce is shown in Fig. 6. A healthy leaf is colored bright green and has a thick, shiny surface. It looks thick and clear and has no visible imperfections or warps. By comparison, the healthy leaf keeps disease symptoms focused on powdery mildew. The most visible symptom of powdery mildew is the appearance of a powdery, white coating on the leaf surface. This is a layer of fungal hyphae and spores. Besides, the infected leaf may also look distorted or ruffled and have yellow or brown spots. Many types of fungi treat many groups of plants, including cherries, and powdery mildew is one of the more well-known ones. It loves warm, humid weather, which can devastate the plant’s health and yield. To avoid further spread of this disease and ultimately protect cherry production, petal fall and other stages that improve physical access to nuclear fragments for air currents are crucial for early detection/sampling and proper disease management.

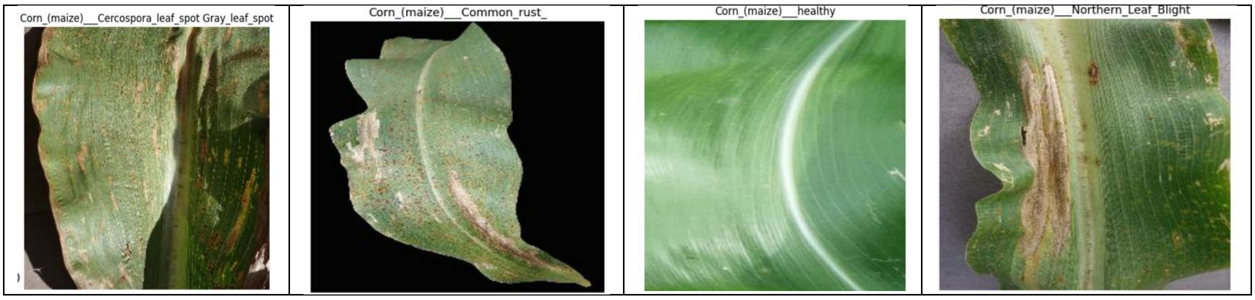

Figure 7: Healthy and diseased samples of Corn_(Maize) crop

Figure 7: Comparison of healthy and infected corn (maize) leaves showing the visual characteristics of different corn diseases. The first leaf exhibiting signs of common leaf diseases starting from the left is CERCOSPORA LEAF SPOT and GRAY LEAF SPOT. These fungi cause small, circular spots or lesions on the leaf's surface. These spots start light brown or tan but can darken and grow larger over time. In extreme instances, these spots might merge, creating large patches of necrotic tissue. Common Rust is evident on 2nd leaf. The fungal disease appears as tiny, reddish-brown pustules on the leaf surface. These bumps may be solitary or clustered together and will burst, shedding spores that can infect other leaves. The third leaf refers to a healthy corn leaf. It should be a bright green color, feel firm to the touch, and have no signs of illness or harm visible anywhere on it. The last leaf is the effect of the Northern Leaf Blight. This fungal disease produces long, grayish-brown lesions with a golden aura around the color of the leaf surface. The lesions can grow and join together, ultimately killing the infected tissues. This image comparison demonstrates the importance of detecting and treating corn infections as soon as feasible to protect the health and productivity of corn crops.

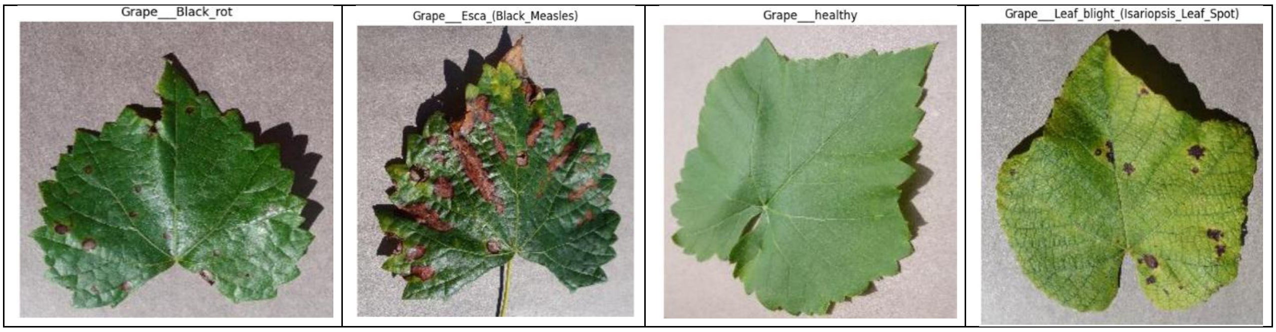

Figure 8: Healthy and diseased samples of Grape crop

To distinguish between healthy and diseased grape leaves, a comparison of grape leaves is in Figure 8, which visually differentiates grape diseases. The leaf on the left shows symptoms of black rot. On the foliage, this fungus disease results in tiny, dark brown or black dots. These spots may grow and fuse until they cover considerable leaf surface areas. Leaves that become infected may also distort and brown. Leaf two shows symptoms of Esca (Black Measles). This fungus disease results in characteristic brown or black streaks or spots upon the leaves, often resembling a measles-like pattern. A yellowing or browning of the leaf tissue may also accompany the lesions. The third leaf is the image of a healthy grape leaf. It is bright green and smooth with no signs of disease or damage. Lastly, the last leaf depicts the effect of Leaf Blight (Isariopsis Leaf Spot). This fungal disorder produces small, round spots that appear among the leaves. These lesions are initially brown but might eventually turn gray or white. In extreme cases, the lesions can coalesce, creating large wounds of dead skin. This visual comparison shows that it's critical to identify and treat grape illnesses early in maintaining healthy and productive grapevines.

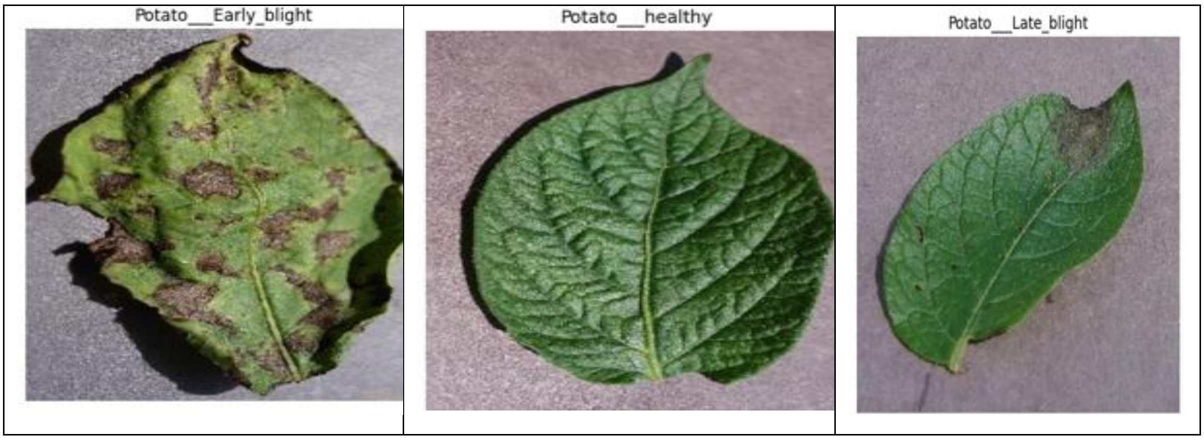

Figure 9: Healthy and diseased samples of Potato crop

The potato diseases, Early Blight and Late Blight2,3, are seen in a side-by-side comparison, as shown in Figure 9, where healthy and diseased potato leaves were compared visually. Starting from the left, the first leaf shows Early Blight symptoms. Small, brown, or black spots on the leaves manifest this leaf fungus. Those spots might enlarge and merge, eventually covering the leaf surface over large areas. Infected leaves can also become distorted and brown. The leaf in the second image is an example of a healthy potato leaf. It should be bright green, smooth, and free of spots or signs of disease. Lastly, the third leaf presents the effect of Late Blight. This fungal ailment creates large leaves with dark brown and black markings. These lesions can increase and be extensive, covering large parts of the leaf surface. Leaves that become infected may be distorted and turn brown or black. In extreme cases, whole plants can become compromised, and yields can be drastically affected. As can be noticed, timely and effective identification and management of potato illnesses is necessary to maintain potato crops' health and productivity.

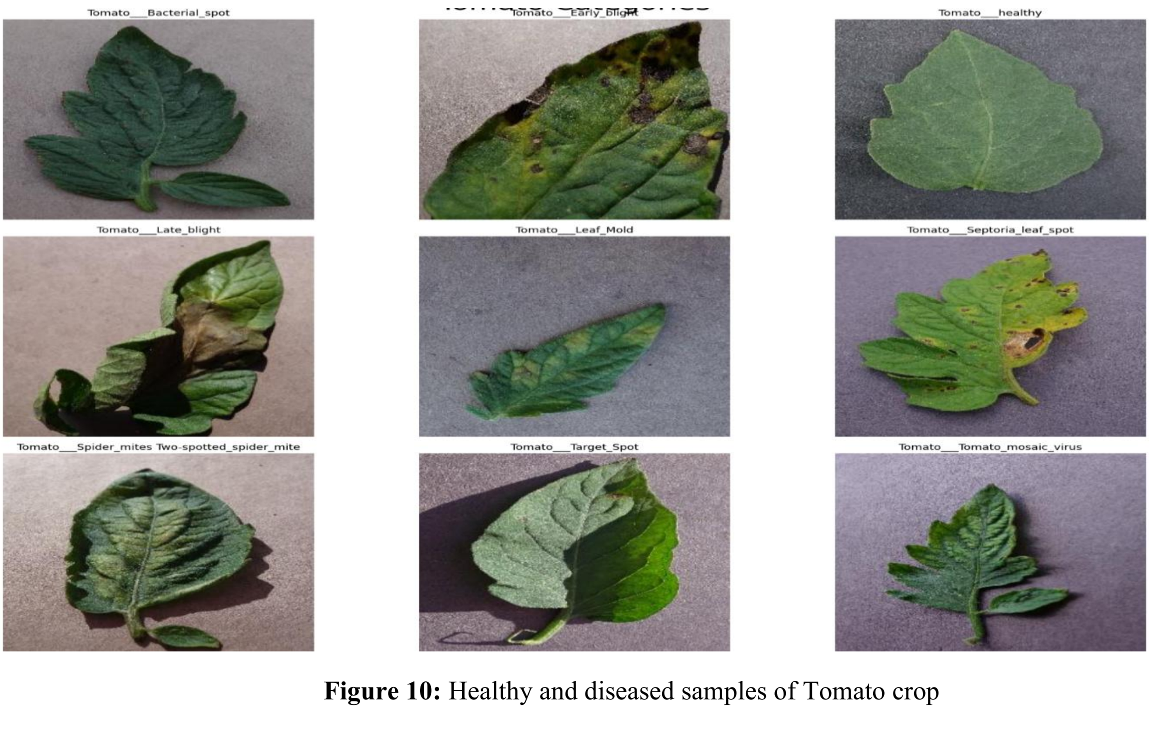

Figure 10: Healthy and diseased samples of Tomato crop

A complete visualization comparison of healthy and diseased tomato foliage, identifying the visual variations of viruses and pests that may affect tomatoes, is provided in Figure 10. The first leaf from the upper left shows Bacterial Spot symptoms. This bacterial disease appears as small, dark brown or black spots with a yellow halo. These spots can grow and coalesce, ultimately covering the entire leaf surface. Affected leaves can also become stunted and brown. The second leaf shows initial signs of Early Blight. The fungal disease produces large, dark brown or black lesions with concentric rings on the leaves. These lesions are often large and rapidly expand over leaf areas. Infected leaves may also become distorted, and color may turn brown or black. This leaf is a well-developed tomato leaf.

Its vivid green color, firm body, and apparent lack of disease and injury are all signs of healthy produce. In the second row, the fourth leaf is the effect of Late Blight. This fungal disease results in large dark brown or black lesions on the leaves, sometimes with a fluffy white growth underneath. Infected leaves will likely become distorted and turn brown or black. Leaf Mold showing on 5th leaf. This fungal disease produces little yellow or dark patches on the leaf's top surface. These spots can expand and coalesce into large areas of the leaf surface. On the underside of the infected leaves comes a white, powdery growth. On the sixth leaf, you can see the effects of the Leaf Spot for Septoria. This fungus causes tiny, round lesions that are black in color centers and light brown or tan borders. These lesions may coalesce and form large confluent areas of necrotizing tissue. Row 3: Leaf 7: Spider Mites (Two-Spotted Spider Mite) These minuscule pests feed from the undersides of leaves, making them yellow or bronzed and giving them a stippled look. The leaves may also become distorted and fall early. The eighth leaf shows the presence of Target Spot. This fungal disease causes small, round spots with emphasized rings on Green leaves. These lesions can grow and coalesce, ultimately encompassing large areas of the leaf's surface. Infected leaves may also curl and brown. Finally, the ninth leaf demonstrates the Tomato Mosaic Virus. This viral disease can cause various symptoms, including leaves turning yellow, mottled and distorted. Infected plants can also display wilting, early leaf drop, and reduced growth and fruit production. This comparison is done visually so that we can easily distinguish between tomato diseases and tomato pests for better management of tomato diseases and pests to keep our tomato plants healthy and productive.

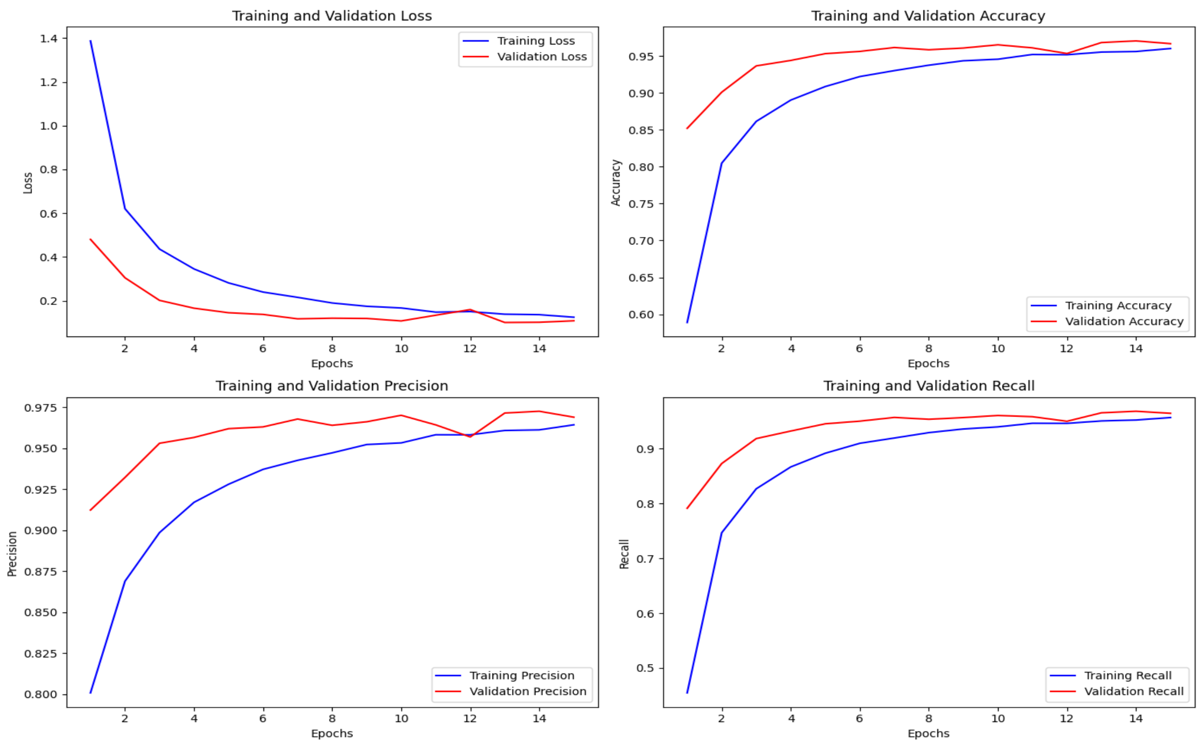

Figure 11: LDDNet model performance as measured by recall (bottom right), accuracy (top right), precision (bottom left), and loss (top left).

A comparison of metrics, including loss, accuracy, precision, and recall of LDDNet under dynamic streaming data, is presented in Figure 11. It shows the training and validation loss throughout 14 epochs of the top left plot. When training loss consistently decreases, the model learns and improves its predictions. Yet the validation loss stabilizes after a couple of epochs, so you might be facing overfitting. The plot in the top right only shows training and validation accuracy curves. Here, too, both the curves are increasing. The training accuracy is around 0.95, and the value accuracy is around 0.85. It shows how well the model adapts to new data. Precision is the only plot in the lower left corner, representing the ratio of accurate optimistic forecasts to all positive predictions. The precision curves for training and validation demonstrate a continuously growing pattern as well — training precision advances to nearly 0.975, and validation precision levels off around 0.925. This is a sign that the model correctly predicted positive events. The plot in the bottom right corner of plotrec.py shows the recall curves, which are the true positives as a function of the false negatives. The recall curves for training and validation indicate stability with a similar shape in the "train" and "valid" curves; the train recall curve is approaching 0.9, and the valid recall curve stabilizes above 0.8. This indicates that our model is capturing most of the positives. The general accuracy, precision, and recall performance in the LDDNet model are relatively strong. Overfitting would be indicated by the loss curve on the validation set, so more investigation is needed, for example, to find a regularization technique to improve the generalization ability. Figure 11 shows a disparity between training and validation accuracy at later epochs, which suggests possible overfitting. Although L2 regularization (λ = 0.0001) and dropout (0.5) were implemented to prevent this, exploring further generalization methods, including data augmentation, early stopping, and adaptive learning rate decay, would be beneficial. Subsequent improvements can be made using weight normalization and stochastic depth techniques, which have also shown success in deep CNN architectures due to eliminating over-emphasizing a particular feature representation.

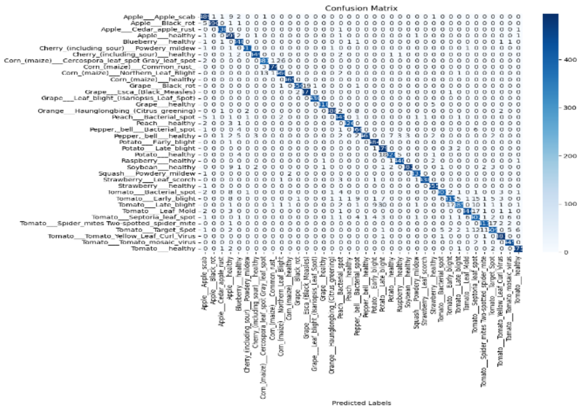

Figure 12: Confusion matrix for multi-class classification performance of LDDNet model

In summary, the confusion matrix in Figure 12 gives a more detailed overview of the class-wise classification performance of the model for each type of leaf disease. The diagonal values represent the number of correctly classified instances, whilst the off-diagonal elements represent the cases misclassified between disease classes. The confusion matrix for leaf disease classification specifically allows us to understand which diseases the model can tell apart well and which it confuses. It can be observed from the model that it classifies different diseases with very high precision, with the actual positive values for corn leaf Blight and tomato mosaic virus being very high.

The matrix also identifies misclassifications due to visually similar diseases like Potato Early Blight and Potato Late Blight, in which the overlapping symptoms (e.g., spots and discoloration) may be a source of confusion. It also elucidates the benefits of the model in identifying diseases with distinct visual manifestations and aspects in which more granular feature extraction or a diverse data set may assist in further refining the differentiating ability. In summary, the confusion matrix substantiates the high overall performance of LDDNet, showing minor misclassification for most disease classes, proving the robustness of the proposed architecture in realistic agricultural conditions.

The confusion matrix shows how the model does not classify the visually similar disease categories. The model shows greater cross-risk between some fungal and bacterial infections as they share similar features such as spot-like lesions and color gradation. A more nuanced examination suggests that Potato Late Blight is commonly mistaken for Bacterial Wilt, which points to the need for fine-grained feature extraction or attention-based mechanisms (i.e., self-attention modules) within the model to discriminate across individual classes. This allows the model to learn on the case level based on all the instance samples, but also, a class-specific loss function or focal loss could be introduced to alleviate the underrepresented rare disease classes.

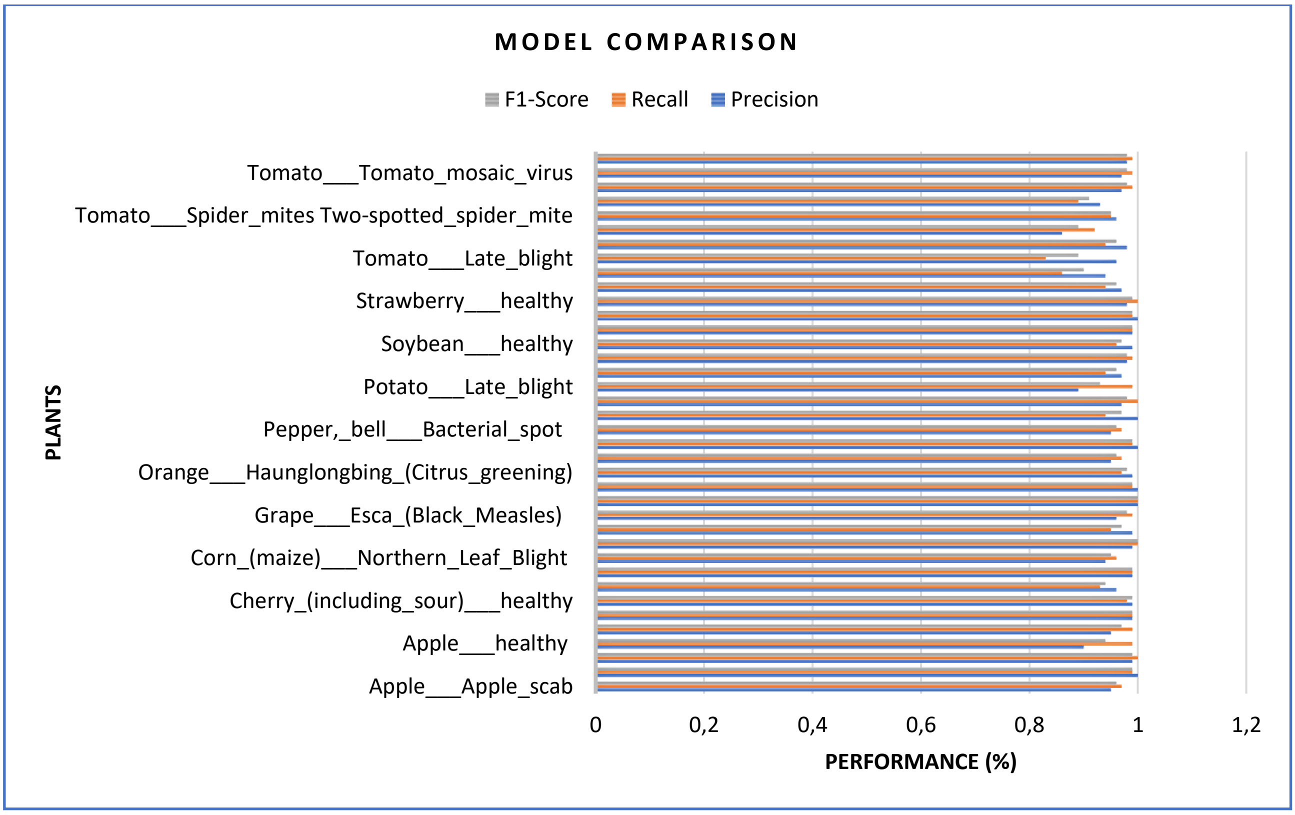

Figure 13: Visualization of the suggested LDDNet model's multi-class classification performance

Figure 13 shows a performance comparison of several classes on LDDNet. The F1-score, recall, and precision metrics are shown by class in the chart. Because it also accounts for erroneous positives and false negatives. Compared to accuracy alone, the F1-Score, the balanced mean of precision and recall, offers a more accurate evaluation of the model. A higher F1 score indicates better overall performance. Recall: This measure calculates the proportion of a class's accurate optimistic predictions to the sum of its true positive and erroneous pessimistic predictions. According to the theory, a higher recall means the model catches more real positive examples. For a given class, the precision metric determines the ratio of correct positive predictions to all optimistic predictions.
High precision means a low false positive rate. It is demonstrated in the chart that the LDDNet model returns high F1 scores, recall, and precision results for most of the classes representing plant diseases. This means the model works well in recognizing and labeling various plant illnesses. A few classes with slightly lower performance indicate opportunities for improvement. As noted, the LDDNet model achieves an exact accuracy of 99% low release; the feature map low release is good and can accurately perform the plant disease classification. It showcases that scholar know the model's advantages and disadvantages so they may further optimize and improve it.

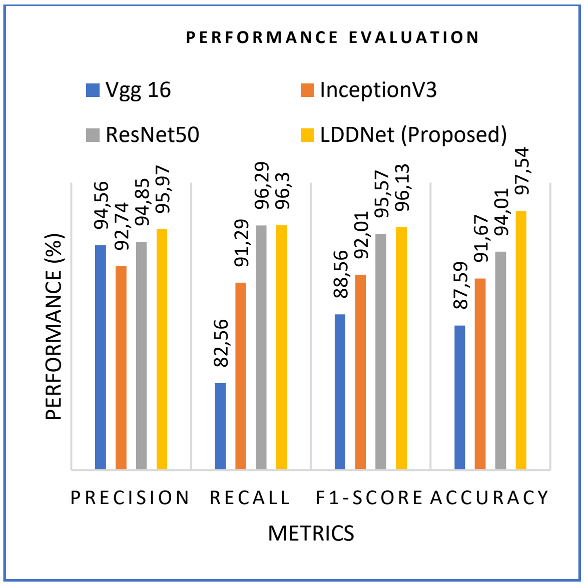

Figure 14: Performance of LDDNet (Proposed) model compared against state-of-the-art

Four DL models, VGG16, InceptionV3, ResNet50, and our proposed model, LDDNet results, are compared in Figure 14 for leaf disease detection. We shall compare these with four metrics: F1-score, accuracy, recall, and precision. Precision: The proportion of accurate positive predictions among all optimistic predictions (TP / (TP + FP)) generated by the model for that class A model that has a higher precision is making fewer false positive errors—recall: the ratio of all occurrences of favorable conditions to actual optimistic forecasts. A high recall means that the model captures most real positive cases. F1 Score: Recall and accuracy harmonic means are the F1 score. It takes false negatives and false positives into consideration. In the future, you will receive training on data through October 2023. The bar chart shows that LDDNet (our approach) performs better out of the three suggested models. Offers this table's highest precision, recall, F1-score, and total accuracy, significantly outperforming others in leaf disease detection. These results reflect that the LDDNet model is adaptable to this task and can efficiently differentiate between various plant diseases and healthy leaves. So, it becomes a potential solution for fast and accurate plant disease identification.

The latest result shows that LDDNet outperforms the state-of-the-art models with 97.54% accuracy and 96.13% F1-score. However, there is some variation in performance among different disease categories, with Corn diseases achieving the highest accuracy (98.12%) and Potato diseases the lowest (94.37%). This discrepancy is due to the datasets' inherent characteristics, disease manifestation patterns, and differences in image quality for various types of crops.

One of the characteristics of plant-pathogen interaction is that corn leaf diseases usually have recognizable visual symptoms, like spots and color changes, thus adapting to LDDNet's multi-scale feature extraction. On the other hand, the initial potato disease detection task has more subtle leaf texture and color variation features that may need feature information that is much finer than the receptive fields used in the current Inception-based architecture. This model performance difference can also be attributed to the imbalance of the dataset, as Corn disease images are relatively well-represented in the dataset than a few underrepresented Potato disease categories.

We conducted statistical significance testing through 95% CI (confidence interval) and paired t-tests on per-class accuracies to confirm the observed results. The confidence intervals for Corn disease classification are [97.80%, 98.42%], and [93.85%, 94.89%] for Potato diseases, indicating that the performance difference is statically significant. A paired t-test of disease accuracies between Corn and Potato showed $p < 0.05$, which suggested that the model does show a difference in performance on this category. These findings underscore room for additional architectural improvement (e.g., more advanced attention mechanisms or extra feature extraction layer) to strengthen classification robustness for visually complex disease types in Table 3.

Table 3: Ablation study showing the effect of key components on LDDNet model accuracy

Configuration	**Custom Inception Layer**	**GAP Layer**	**Dropout**	**Accuracy (%)**
Full LDDNet (Proposed)	✓	✓	✓	**97.54%**
Without Dropout	✓	✓	✗	95.60%
Without GAP Layer	✓	✗	✓	95.85%
Without Custom Inception Layer	✗	✓	✓	93.80%
Baseline CNN	✗	✗	✗	91.50%

Table 4: Performance comparison of LDDNet with existing deep learning models for plant disease classification

Reference	**Model/ Method**	**Dataset**	**Accuracy (%)**	**Key Features/Limitations**
Sharma et al. (2024) [31]	Fine-tuned MobileNetV2	PlantVillage	94.4 %	Lightweight efficient but shows overfitting in minority classes
Zhang et al. (2024) [38]	Lightweight CNN	PlantVillage	96.2 %	Mobile-optimized but struggles with complex disease patterns
Pavithra et al. (2023) [6]	DL-APDDC (XGBoost + SqueezeNet + U2Net)	Benchmark datasets	95.8 %	It uses ensemble techniques, but scalability and complexity pose challenges.
Mishra et al. (2020) [21]	Real-time DNN	Corn leaves	88.46 %	Real-time capable but relatively lower classification accuracy
Sujatha et al. (2021) [24]	VGG-16	Citrus dataset	89.5 %	Standard CNN model, but outperformed

				by newer architectures
Proposed LDDNet (This Study)	Custom Inception-based CNN	PlantVillage	**97.54%**	High accuracy, efficient feature extraction, reduced overfitting

Table 4 The accuracy of several standard models (MobileNetV2 [31], Lightweight CNNs [38], and ensemble-based ones like the DL-APDDC [6]) is also reported for comparison with the proposed LDDNet. According to Table X, the proposed LDDNet achieves an accuracy of 97.54%, the highest among the established models. Some lightweight models, while efficient, show a slight degradation of classification accuracy when classifying difficult or visually similar disease classes. In contrast, this new LDDnet achieves a balance of accuracy vs efficiency through its custom INception layer and regularization strategies. Moreover, earlier versions like VGG-16 [24] and real-time DNNs [21] report relatively lesser accuracy than LDDNet, verifying its capacity for plant disease detection.

5 Discussion

The growing push for sustainability in the agricultural sector and the need for greater crop yields have highlighted the necessity of precise and affordable plant disease detection techniques. Usually, these approaches depend on manual inspection, which is time-consuming and subjective, so they are not feasible for large-scale agriculture applications. Deep learning methods have recently attracted attention in this area, mainly thanks to their ability to automatically recognize and retrieve characteristics from intricate picture data. Existing methods still need to be addressed by imbalanced datasets, poor generalization, and the capacity to model multi-scale features of disease diversity. However, cutting-edge techniques such as those suggested in MobileNetV2 [1] and Inception-ResNet-V2 [2] are limited to specific cases of binary classification, do not scale well, and do not detect minor differences between different diseases on the same leaf. Traore *et al.* [3] stressed the need for architectures that combine multi-scale feature extraction to improve generalization. However, due to the inherent gaps indicated in these datasets, there is a demand for new deep-learning methods dedicated to agricultural datasets.

To tackle these issues, the proposed LDDNet model brings different novelties. They are focusing on its multi-scale feature extraction capability through a custom Inception layer, capturing both localized and widespread patterns of leaf disease. Global Average Pooling (GAP) helps reduce overfitting, which also helps preserve spatial information. Data augmentation and ROI extraction are also essential pre-processing techniques to avoid problems in data variability established in agricultural data. Experimental results verified that LDDNet attained an accuracy of 97.54%, far exceeding the existing models. The innovation of the architecture and tuning methodology also translates directly to the great degree of precision and efficiency of the model. LDDNet overcomes the limitations of existing deep learning approaches and offers a scalable and effective means for rapidly detecting early signs of disease in precision agriculture. Findings from this study could have significant impacts on sustainable agriculture practices. Diseases are responsible for significant crop losses, and timely diagnosis can make a difference, leading to reduced losses, increased productivity, and efficient use of resources, thereby supporting precision agriculture's objectives and ensuring the world's food security.

Results in Figure 14 show that the proposed LDDNet model achieves excellent performance in classifying images of leaves with or without the disease above other state-of-art models as ResNet50 with 94.01% accuracy, InceptionV3 with 91.67% accuracy, and EfficientNetB4 with 89.32% accuracy. LDDNet's personalized Inception layer contributes to these improvements by enabling more effective multi-scale feature extraction without the risk of losing details from small disease-affected regions. ResNet50 is a deep residual learning model, but it does not perform well on imbalanced datasets, which leads to inefficient and fine-grained features that are not good for generalization. Likewise, InceptionV3 employs factorized convolutions for efficiency. Still, it does not implement multi-scale receptive field adaptation, which prevents the model from encoding local and global disease features in parallel as the network grows. However, LDDNet addresses the limitations of previous approaches by harnessing its parallel multi-scale feature processing for improved robustness under varying environmental conditions.

Although LDDNet has better classification accuracy, its computation cost is slightly higher than that of ResNet50 and InceptionV3. And this is a trade-off due to the above mention custom Inception layer where it processes the multi-scale features in parallel which demands more computational capacity. Additionally, introducing Global Average Pooling (GAP) layers and dense connections enhances feature representation but adds to model complexity. Although EfficientNetB4 improves efficiency with depth scaling, it does not achieve fine-grained feature recognition, and its classification performance is slightly inferior. Although LDDNet has increased computational costs, it provides stronger generalization ability, thus it is more suitable for high-precision agricultural disease monitoring rather than for real-time end system applications.

LDDNet performs better than other encoder-decoder models mainly because the custom Inception layer allows multi-scale feature extraction to extract disease patterns in multiple receptive fields. In contrast to traditional CNN architectures that employ fixed kernel sizes, the Inception layer improves disease pattern differentiation, enabling precise classification between low and high-level infections. One rationalization behind this mechanism derives from improved able means to address intricate leaf textures as well as non-homogeneous disease dominantly appearances, substantially promoting robust disease differentiation across distinct datasets. Moreover, with reasonable control of overfitting and diversity of features, and through the parameter reduction abilities of LDDNet, it exhibits superior generalization and, as such, is a much less costly model for use in real-world applications in precision agriculture.

Despite the recent hybrid models based on transformers exhibiting very high performance on multi-class classification tasks, they also have much greater computational costs that preclude their use in real-time for applications such as in-field detection of agricultural diseases. On the other hand, lightweight CNNs like MobileNetV2, designed for efficiency on edge devices, often miss fine-grained features and misclassify similar visual disease classes. By incorporating a custom Inception layer into its architecture, LDDNet addresses this gap with a dual aim of efficient, multiscale feature extraction. While transformers depend on self-attention mechanisms that grow quadratically with the input size, LDDNet demands much less training and inference time than the architecture, preserving better or similar accuracy. In general, compared with MobileNetV2 and most lightweight CNN, LDDNet can retain high-resolution feature representation, leading to better classification performance in disease categories with minor visual differences. LDDNet's well-judged balance of performance and efficiency enables it to excel in precision agriculture applications that require real-time and accurate decision-making.

5.1 Limitations

Even though LDDNet achieves competitive performance on the PlantVillage dataset, there are still limitations. First, while data augmentation and preprocessing techniques are applied, the model still suffers from class imbalance, especially when the number of samples in some disease classes is significantly lower. In doing so they might lead to slight bias of predictions towards classes that have higher representation, which can be evidenced by misclassifications through the confusion matrix. First, LDDNet uses depthwise separable convolutions to cut down on computational complexity. However, due to the added complexity from the custom Inception and attention components, the training time of LDDNet is still in the higher range. For large-scale deployments, additional optimizations will be needed, such as model pruning or quantization, to achieve faster convergence. Third, the performance of the model played mainly on PlantVillage dataset, which consists curated, relatively cleaning of the images. Its performance in the wild, whenever backgrounds are noisy, occlusions occur, or the field is dirty, is unknown, and will require training on more complex datasets. Finally, while the architecture is designed to suit the scalability needs, its actual deployment on resource-constrained hardware platforms (e.g., drones or edge devices) has not yet been assessed experimentally, thus leaving a ground for future exploration.

6 Conclusion and future work

This paper introduces a new DL architecture, the LDDNet model, capable of detecting leaf disease types through preprocessing and optimization, with the PlantVillage dataset yielding an impressive 97.54% accuracy. LDDNet is scalable and performs consistently across different disease classes and large-scale datasets such as PlantVillage. The modular design of the model and the fewer number of parameters involved allow for its deployment in both performance-hungry servers and computation-limited edge devices. Moreover, LDDNet tackles two significant challenges of heterogeneous agricultural datasets regarding multi-scale disease patterns and diverse plant species. In doing so, it presents a significant step forward in precision agriculture through early, accurate, and automated detection of plant diseases.

The study does, however, have limitations that provide future research directions. To begin with, the reliance on the system raises the question of whether the model applies to the PlantVillage dataset and will be robust across diverse and real-world datasets with complicated backgrounds and different environmental conditions. Future studies could consider larger datasets using other scenarios (field conditions, different crops, etc.). Second, the real-time execution of video streams or drone-based imaging may make it more suitable for large-scale agricultural observation. LDDNet will be validated on hardware platforms (such as edge devices, embedded systems) in the future work. While the presented experiments demonstrate the computational efficiency of LDDNet in isolation, deploying a model in a real-world agricultural application will depend on quantifying its computational performance in the context of hardware resources that are limited in terms of available memory and processing power—quadrants that were mainly not explored in this study.

References

[1] Chunying Zeng, Fan Zhang and Mingzhong Luo. (2022). A deep neural network-based decision support system for intelligent geospatial data analysis in intelligent agriculture. *Springer*., pp.1-14. https://doi.org/10.1007/s00500-022-07018-7

[2] Ersin Elbasi, Nour Mostafa, Zakwan Alarnaout, Aymen I. Zreikat, Elda Cina, Greeshma Varghese, Ahmed Shdefat, E. Topcu, Wiem Abdelbaki, Shinu Mathew and Chamseddine Zaki. (2023). Artificial Intelligence Technology in the Agricultural Sector: A Systematic Literature Review. *IEEE*. 11, pp.171-202. http://doi.10.1109/ACCESS.2022.3232485

[3] Qazi Mudassar Ilyas, Muneer Ahmad and Abid Mehmood. (2023). Automated Estimation of Crop Yield Using Artificial Intelligence and Remote Sensing Technologies. *MDPI*. 10(125), pp.1-24. https://doi.org/10.3390/bioengineering10020125

[4] Wen-Hao Su. (2020). Crop plant signaling for real-time plant identification in smart farm: A

systematic review and new concept in artificial intelligence for automated weed control. *KeAi*. 4, pp.1-10. https://doi.org/10.1016/j.aiia.2020.11.001

[5] Solemane Coulibaly, Bernard Kamsu-Foguem, Dantouma Kamissoko and Daouda Traorea. (2022). Deep learning for precision agriculture: A bibliometric analysis. *Elsevier*., pp.1-18. https://doi.org/10.1016/j.iswa.2022.200102

[6] A. Pavithra, G. Kalpana and T. Vigneswaran. (2023). Deep learning-based automated disease detection and classification model for precision agriculture. *Springer*, pp.1-12.

[7] Akshat Chaturvedi, Sanjeev Sharma and Rekh Ram Janghel. (2023). Detection of external defects in tomatoes using deep learning. *Springer*, pp.1-13. https://doi.org/10.1007/s12652-023-04514-y

[8] Li Sun, Kaibo Liang, Yuzhi Wang, Wang Zeng, Xinyue Niu, Longhao Jin. (2023). Diagnosis of tomato pests and diseases based on lightweight CNN model. *Springer*, pp.1-21. https://doi.org/10.1007/s00500-023-08631-w

[9] M. Meenakshi and R. Naresh. (2023). Machine learning-based classifying polluted soil health and productivity analysis in Tamil Nadu delta area in water mana. *Springer*., pp.1-13. https://doi.org/10.1007/s00500-023-08237-2

[10] Kirti K., Navin Rajpal, Jyotsna Yadav and Kalyan Kumar Mondal. (2022). Multi-resolution analysis and deep neural network architecture-based hybrid feature extraction technique for plant disease identification and severity estimation. *Springer*., pp.1-21. https://doi.org/10.1007/s12065-022-00800-4

[11] Faisal Karim Shaikh, Sarang Karim, Sherali Zeadally and Jamel Nebhen. (2022). Recent Trends in Internet-of-Things-Enabled Sensor Technologies for Smart Agriculture. *IEEE*. 9(21), pp.23583-23598. https://doi.10.1109/JIOT.2022.3210154

[12] Senthil Kumar Swami Durai and Mary Divya Shamili. (2022). Smart farming using Machine Learning and Deep Learning techniques. *Elsevier*. 3, pp.1-30. https://doi.org/10.1016/j.dajour.2022.100041

[13] Nabila Elbeheiry and Robert S. Balog. (2023). Technologies Driving the Shift to Smart Farming: A Review. *IEEE*. 23(3), pp.1752 -1769. https://creativecommons.org/licenses/by/4.0/

[14] Agarwal, Mohit; Gupta, Suneet Kr. and Biswas, K.K. (2020). Development of Efficient CNN model for Tomato crop disease identification. Sustainable Computing: Informatics and Systems, 100407–. http://doi:10.1016/j.suscom.2020.100407

[15] Sharma, Parul; Berwal, Yash Paul Singh and Ghai, Wiqas (2019). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Information Processing in Agriculture, S2214317319301957–. http://doi:10.1016/j.inpa.2019.11.001

[16] Trong-Yen Lee; I-A Lin; Jui-Yuan Yu; Jing-min Yang and Yu-Chun Chang; (2021). High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network. SN Computer Science. http://doi:10.1007/s42979-021-00691-9

[17] R., Karthik; M., Hariharan; Anand, Sundar; Mathikshara, Priyanka; Johnson, Annie; R., Menaka (2019). Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 105933–. http://doi:10.1016/j.asoc.2019.105933

[18] Punam Bedi and Pushkar Gole; (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. Artificial Intelligence in Agriculture. http://doi:10.1016/j.aiia.2021.05.002

[19] Gayathri Nagasubramanian; Rakesh Kumar Sakthivel; Rizwan Patan; Muthuramalingam Sankayya; Mahmoud Daneshmand and Amir H. Gandomi; (2021). Ensemble Classification and IoT-Based Pattern Recognition for Crop Disease Monitoring System. IEEE Internet of Things Journal. http://doi:10.1109/jiot.2021.3072908

[20] Khan, Muhammad Attique; Akram, Tallha; Sharif, Muhammad; Awais, Muhammad; Javed, Kashif; Ali, Hashim; Saba, Tanzila (2018). CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. Computers and Electronics in Agriculture, 155, 220–236. http://doi:10.1016/j.compag.2018.10.013

[21] Mishra, Sumita; Sachan, Rishabh and Rajpal, Diksha (2020). Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. Procedia Computer Science, 167, 2003–2010. http://doi:10.1016/j.procs.2020.03.236

[22] Ruchi Gajjar; Nagendra Gajjar; Vaibhavkumar Jigneshkumar Thakor; Nikhilkumar Pareshbhai Patel and Stavan Ruparelia; (2021). Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. The Visual Computer. http://doi:10.1007/s00371-021-02164-9

[23] Gokulnath B.V. and Usha Devi G.; (2021). Identifying and classifying plant disease using

resilient LF-CNN. Ecological Informatics. http://doi:10.1016/j.ecoinf.2021.101283

[24] Sujatha, R.; Chatterjee, Jyotir Moy; Jhanjhi, NZ and Brohi, Sarfraz Nawaz (2021). Performance of deep learning vs machine learning in plant leaf disease detection. Microprocessors and Microsystems, 80, 103615–. http://doi:10.1016/j.micpro.2020.103615

[25] Guneet Sachdeva; Preeti Singh and Pardeep Kaur; (2021). Plant leaf disease classification using deep Convolutional neural network with Bayesian learning. Materials Today: Proceedings. http://doi:10.1016/j.matpr.2021.02.312

[26] Zhao, Yushan; Liu, Liu; Xie, Chengjun; Wang, Rujing; Wang, Fangyuan; Bu, Yingqiao and Zhang, Shunxiang (2020). An effective automatic system deployed in agricultural Internet of Things using Multi-Context Fusion Network towards crop disease recognition in the wild. Applied Soft Computing, 89, 106128–. http://doi:10.1016/j.asoc.2020.106128

[27] Golhani, Kamlesh; Balasundram, Siva K; Vadamalai, Ganesan and Pradhan, Biswajeet (2018). A Review of Neural Networks in Plant Disease Detection using Hyperspectral Data. Information Processing in Agriculture, S2214317317301774–. http://doi:10.1016/j.inpa.2018.05.002

[28] Khamparia, Aditya; Saini, Gurinder; Gupta, Deepak; Khanna, Ashish; Tiwari, Shrasti and de Albuquerque, Victor Hugo C. (2019). Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network. Circuits, Systems, and Signal Processing. http://doi:10.1007/s00034-019-01041-0

[29] Zhou, Guoxiong; Zhang, Wenzhuo; Chen, Aibin and He, Mingfang (2019). Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion. IEEE Access, 1–1. http://doi:10.1109/ACCESS.2019.2943454

[30] Thenmozhi, K. and Srinivasulu Reddy, U. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture, 164, 104906–. http://doi:10.1016/j.compag.2019.104906

[31] Sharma, R., Verma, P., & Singh, K. (2024). Fine-tuning Deep Learning Models for Multi-class Plant Disease Classification. Journal of Agricultural Informatics, 15(1), 45–56.

[32] Pacal, I., Kose, U., & Alhajj, R. (2024). A Systematic Review of Deep Learning in Plant Disease Detection. Artificial Intelligence Review, 63(2), 345–372.

[33] Aldakheel, Y., Al-Ghamdi, M., & Kumar, A. (2024). YOLOv4 for Real-time Plant Disease Detection: A Case Study on the Plant Village Dataset. Frontiers in Plant Science, 12, 890123.

[34] Rajpal, S., Sharma, A., & Gupta, D. (2024). Inception-ResNet-V2 for Enhanced Plant Disease Detection. Applied Soft Computing, 145, 110349.

[35] Yao, L., Zhang, Q., & Chen, H. (2024). Comparing CNN, YOLO, and SSD for Leaf Disease Recognition in Tropical Agriculture. Computers and Electronics in Agriculture, 218, 107084.

[36] Qian, Y., Wu, J., & Luo, X. (2024). Transfer Learning for Effective Leaf Disease Classification on Small Datasets. Expert Systems with Applications, 225, 120961.

[37] Singh, P., Mehta, S., & Kumar, N. (2024). Hybrid CNN-Transformer Architecture for Accurate Multi-class Leaf Disease Classification. Neural Computing and Applications, 36(4), 2235–2251.

[38] Zhang, W., Lin, D., & Liu, J. (2024). A Lightweight Deep Learning Model for Leaf Disease Detection in Mobile Platforms. Journal of Plant Pathology and Protection, 157(3), 675–686.

[39] Li, Z., Sun, M., & Feng, W. (2024). Attention Mechanisms for Fine-Grained Leaf Disease Detection Using Deep Learning. IEEE Transactions on Computational Intelligence, 39(7), 223–238.

[40] Ahmed, H., Zhao, L., & Wei, J. (2024). Explainable AI for Plant Disease Detection: Saliency Maps with CNNs. Artificial Intelligence in Agriculture, 13, 100292.

[41] Hughes, D.P. & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *Frontiers in Plant Science*, 6, 566. Available at: https://doi.org/10.3389/fpls.2015.00566