Received: April 24, 2024. Revised: June 16, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations

Shaik Thaseentaj[1] **S. Sudhakar Ilango**[1]*

[1]*School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India*
*Corresponding author: sudhakar.ilango@vitap.ac.in.

Abstract: Growing mangoes is an important part of life in southern India and a major economic driver for the area. Nevertheless, several leaf diseases often impede mango tree development and production, substantially affecting harvest output and quality. Detecting and identifying mango leaf diseases early can be challenging due to the diverse crop varieties, climatic circumstances, and numerous disease signs. While deep-learning methods have been developed to address this problem, they generally need help to detect illnesses across geographies and crop types. To tackle this difficulty, this research offers a transfer learning model that uses Explainable Artificial Intelligence (XAI) characteristics to identify and categorize leaf diseases. This research proposes MLTNet (Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations) in this study. Our study utilized a dataset from Southern India comprising 1,275 high-quality images of mango leaves affected by diseases like rust and powdery mildew, augmented to 11,480 images across 14 classes to enhance model training and robustness.This novel model utilizes Explainable Artificial Intelligence (XAI) techniques such that leaf disease detection and categorization may achieve higher levels of accuracy. The research work lies in the development of the MLTNet model, which integrates Explainable Artificial Intelligence (XAI) techniques with the ResNet50 architecture to enhance classification accuracy in mango leaf disease detection. This model uniquely employs advanced data pre-processing methods like Error Level Analysis and incorporates Grad-CAM for feature localization and visual explanations. We compared MLTNet's performance with state-of-the-art models like ResNet-50, VGG-16, and InceptionV3, focusing on accuracy, interpretability, and computational efficiency. MLTNet demonstrated superior performance, achieving a training accuracy of 94.3% and a test accuracy of 86.3%, which notably surpasses other models under similar conditions. This success is attributed to the model's ability to leverage complex features from the augmented dataset and the added interpretability provided by XAI techniques.

Keywords: Mango leaf disease, Transfer learning, ResNet50, Error level analysis, Feature localization, Visual explanations.

1. Introduction

Mango (Mangifera indica), known as the "king of fruits," holds significant agricultural importance globally, with India being a leading producer, contributing approximately 40% of the world's mangoes. However, various leaf diseases can substantially impact mango crop productivity, resulting in estimated yield losses of 30-40% [1]. Timely detection and management of these diseases are crucial to prevent crop losses and maintain fruit quality. This study addresses the need for automated disease detection systems, given the challenges of manual inspection, which is time-consuming and error-prone [2]. Focusing on the South Indian climate, the research aims to develop an advanced transfer learning framework to accurately identify and categorize mango leaf diseases. The initiative involves creating a comprehensive dataset encompassing prevalent disease classes, such as Anthracnose and Leaf Blight, subjected to rigorous pre-processing for optimal input to the transfer learning model [3, 4]. Recognizing the interpretability gap in deep-learning models, particularly in agriculture, the research introduces the

MLTNet model, integrating feature localizations and visual explanations with transfer learning. This innovative approach seeks to enhance precision in leaf disease classification, promoting broader acceptance in real-world agricultural scenarios [5].

The main objective of this study is to enhance mango leaf disease classification accuracy in leaf disease classification. The pivotal importance of the problem is mango leaf diseases significantly impact crop yield and quality, leading to substantial economic losses. Current methods for disease detection are labour-intensive, error-prone, and inefficient. Our system, MLTNet, addresses these challenges by employing a transfer learning framework, enhancing accuracy and explainability in disease identification. This system is crucial for providing rapid, precise, and cost-effective disease identification technology, assisting farmers and agronomists in early disease management. MLTNet, with 54 layers, is built upon the widely used ResNet50 architecture, a deep convolutional neural network renowned for its capacity to uncover intricate details from images [6]. To ensure optimal data quality, we employ Error Level Analysis as a data pre-processing technique, followed by TensorFlow-based image processing to standardize the image processing across the dataset. To improve the precision of our models, we use dropout techniques and increased layer density inside the ResNet50 framework [6]. To further understand the model's reasoning and promote openness, we also employ the Grad-CAM method for feature localization and visual explanation [7]. This allows us to identify critical areas in the input images that substantially impact the model's classification outcomes, making the process more trustworthy and interpretable. Throughout our research, we optimize the MLTNet model's parameters by training it over many epochs. A training accuracy of 94.3% and a test accuracy of 86.3% demonstrate remarkable precision in the outcomes. To prove MLTNet's efficacy, we test it on 14 distinct disease classes of mango leaves and compare its results to those of other popular models used for leaf disease detection and classification. When compared to other methods, our suggested model proves to be superior in detecting and categorizing mango leaf illnesses.

The significant contributions to the research are

- Investigate the effectiveness of MLTNet, an enhanced transfer learning model, in improving the accuracy of mango leaf disease classification accuracy.
- Design and fine-tune MLTNet, Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations network architecture optimized for mango leaf disease classification for predicting classification accuracy.
- Evaluate the proposed approach using a comprehensive augmented South Indian mango dataset with 14 types of diseases.
- Compare the performance of the proposed MLTNet approach with state-of-the-art works for mango leaf disease classification.

The work is presented as follows: Section II provides preliminaries and literature related to the classification of mango leaf diseases. Section III offers the initial work conducted and outlines the proposed methodology. Section IV presents the results and discussions of the proposed techniques. Finally, Section V summarizes the findings, presents the conclusion, and provides recommendations for future research.

2. Basic preliminaries and related works

Detecting and understanding agricultural leaf diseases is a top priority for machine vision researchers. Using machine vision technology to take photos of the leaves, this approach may identify illnesses that affect them. [8]. The sector has been at the forefront of adopting computer vision-based technology for crop diagnosis and identification. This technology has partially replaced conventional farming and classifying leaf diseases.

AI, notably deep learning, is pivotal in leaf disease detection [36]. In medical applications, it enhances colonic polyp detection [36] and transforms bladder cancer diagnosis, improving tumor detection precision [37]. In security, introduces WARDIC for highly accurate weapon detection using Convolutional Neural Networks [38]. Also in medical maretials research [39] explores deep learning, fostering collaboration between disciplines [39]. [40] addresses online voting challenges with a mobile-based application incorporating deep learning-based face detection [40]. For oral cancer, [41] highlights VGG19's efficacy in early detection using pre-trained Convolutional Neural Networks [41].

2.1 Current literature on transfer learning models for identifying leaf diseases

Recent advancements in agriculture underscore the shift towards computer vision for crop disease identification, departing from traditional methods. Techniques like Convolutional Neural Networks (CNNs) facilitate image-based detection, addressing challenges like poor contrast and subtle lesion differences, promising more accurate results [9, 10].

Deep transfer learning, especially in CNN architectures, has gained prominence for diagnosing leaf diseases, enhancing accuracy in automated systems [11]. Comprehensive studies, such as [12], highlight the efficacy of advanced image processing in detecting various plant diseases. Ongoing research emphasizes the crucial role of advanced image processing in elevating crop yield and mitigating economic losses in agriculture.

A "Transfer Learning" approach takes an existing model and uses it as a foundation for a new model, this time for a different goal. Researchers have been using pre-trained networks for disease classification on mango leaves to extract feature representations that are adept at distinguishing between different types of leaf diseases. The research team then embarked on an evaluative phase, contrasting the results garnered from their methodology with those obtained through alternative tactics which incorporated the use of transfer learning and notable CNN models like AlexNet [13], VGG16, ResNet-50, and MobileNet [14-16]. Furthermore, the team proposed the utilization of a Modified CNN for the precise classification of mango leaves, Anthracnose. This endeavour was grounded on the analysis of a robust dataset comprising 1070 samples of mango tree leaves documented in real-time, encompassing both healthy and infected specimens. Upon comparison with existing cutting-edge solutions, the proposed MCNN model manifested superior classification accuracy.

This study investigates the use of pre-trained deep-learning models like VGG16 and ResNet50 for tomato plant disease and leaf recognition [17]. The study compares these models' ability to diagnose different diseases [18]. We test the use of transfer models VGG16 and InceptionV3 to classify leaf diseases. Accurate illness categorization requires transfer learning, fine-tuning, and feature extraction from these pre-existing models. We thoroughly test pre-trained CNN models like VGG16, ResNet50, and InceptionV3 for leaf disease detection [19]. We test their ability to identify healthy and diseased leaves. Pre-trained deep learning models like VGG16 and InceptionV3 are extensively used to diagnose and classify plant diseases [20]. This study shows how pre-trained models, transfer learning, and fine-tuning can improve disease diagnosis.

Transfer learning for leaf disease categorization using pre-trained models like VGG16 and ResNet50 is examined in this paper [21]. We examine numerous transfer learning methods, including attention mechanisms in CNN models like VGG16 and InceptionV3, using plant disease datasets. Our data show that this method accurately diagnoses and classifies leaf diseases. In [22], we use deep learning to accurately diagnose and classify mango tree diseases. We describe the process, which includes model training on a carefully selected mango illness dataset and deep-learning architecture evaluation. These findings show that deep learning can improve agricultural practices and crop health by detecting mango illnesses. This study shows how deep learning systems accurately identify and classify mango illnesses.

The research delineated in [26], pioneered the deployment of a highly sophisticated neural network ensemble (NNE) in their analytical model to facilitate the precise identification of diseases in mango leaves. This innovative system, a confluence of multiple neural networks, has been meticulously calibrated to distinguish the nuances between healthy and diseased leaves, showcasing a remarkable accuracy rate of 87.5% in identifying healthy specimens. This notable success rate not only signifies the model's robustness but also marks a pivotal advancement in agricultural technology.

Both manual and automated planting methodologies currently face constraints in surveying expansive land masses and offering crucial insights at initial stages for well-grounded decision-making. As a result, there is an urgent need to forge ahead in crafting automated mechanisms that stand as applicable, dependable, and economically viable, to monitor vegetation health while conveying pertinent data for administrative use. A notable evolution has been witnessed in the domains of pattern discernment and classification, largely attributed to the AI, leveraging the capabilities of Convolutional Neural Networks (CNNs) in these systems. This integration has notably enhanced the precision of diagnosing diseases affecting mangoes, as evidenced by the application of renowned frameworks like AlexNet-GoogLeNet [24] and InceptionV3 (cited in [25]. However, despite these strides in image analysis technology, various hurdles continue to obstruct the path to pinpointing ailments afflicting mango foliage and fruits with complete accuracy.

In the domain of mango leaf disease, [42] have provided substantial contributions through their research on Deep Convolutional Neural Networks (CNNs). Their study, explores the potential of CNNs to accurately classify diseases affecting mango leaves in South India. They report high accuracy levels, underscoring the effectiveness of deep learning in agricultural applications. However, their work also identifies a significant limitation related to the initial dataset, which was relatively small and thus, might not fully represent the variability found in real-world scenarios. They suggest that data augmentation could be crucial in overcoming these limitations by artificially expanding the dataset to include more varied examples without the need for additional real images. This recommendation for augmentation aligns with broader research trends emphasizing the importance of robust datasets in training deep learning models. Our current study builds upon their findings by implementing a rigorous data augmentation process, thereby enhancing the dataset's diversity and size.

The research paper delineated in reference [5] ventures into the realm of explainable AI (XAI), focusing on its integration with deep learning models to enhance their transparency and interpretability. It sheds light on various methods facilitating model interpretability, including feature visualization and saliency maps, and extends into discussing the pivotal role of XAI in significant sectors. Meanwhile, the application of Grad-CAM stands as a cornerstone in mango leaf disease identification, offering visual elucidations from deep networks and thereby pinpointing the disease-afflicted regions on leaves [7]. By highlighting the critical areas in leaf images that significantly influence disease classification, recent research leveraging Grad-CAM not only ensures the precision of the model but also nurtures trust in automated systems for disease detection, demonstrating the extensive potential and real-world applications of XAI and Grad-CAM in advancing automated disease diagnosis systems [25].

2.2 Issues in the conventional techniques

Our literature on classification of mango leaf diseases in Southern India, presenting the first of its kind study focused exclusively on this region's specific agricultural context. Traditional methods for managing mango leaf diseases have significant limitations. Manual inspection, while common, is highly subjective and can lead to inconsistent results and management delays. Traditional machine learning approaches require handcrafted features and extensive domain knowledge, making them difficult to adapt and scale across different regions and varieties of mangoes. Moreover, standard deep learning models, such as basic CNNs, often suffer from overfitting and lack the transparency needed for decision-making, reducing their reliability and acceptability among end-users such as farmers and agricultural experts. In contrast, our approach utilizes a novel model that leverages advanced AI techniques to overcome these challenges, providing a robust and scalable solution tailored to the unique needs of mango cultivation in Southern India. The table 1, presents the comparison study on the existing works.

Table 1. Mango leaf disease detection and classification comparison study of existing works

Ref	Methodology / model	drawbacks
[9, 10]	Basic CNN	Poor contrast handling, difficulty in detecting subtle lesion differences
[11]	Deep Transfer Learning in CNNs	Accuracy levels are very less.
[12]	Advanced Image Processing	Concentrated on specific diseased region identification to improve the classification.
[13-16]	AlexNet, VGG16, ResNet-50, MobileNet	Specific to Anthracnose, may not generalize well to other diseases or conditions
[17-18]	VGG16, ResNet50	Requires fine-tuning and extensive feature extraction, which can be resource-intensive
[19-20]	VGG16, InceptionV3	Extensively used for diagnosis
[21]	VGG16, ResNet50 with attention mechanisms	Accurate diagnosis and classification. Datasets with 8 classes it is giving good accuracies.
[22]	Deep Learning Models	Improved agricultural practices and crop health. Requires careful dataset curation and model training
[26]	Neural Network Ensemble (NNE)	87.5% accuracy in identifying healthy specimens. Complex setup, high calibration needs
[42]	Deep CNN Models	Accuracy is good. But the dataset is new and limited images. Augmentation is needed.

The recent literature showcases a growing inclination towards the fusion of transfer learning and Grad-CAM methods for enhanced classification of diseases in mango leaves. This collaborative technique capitalizes on the feature identification abilities of transfer learning and the visualization advantages of Grad-CAM, fostering greater accuracy and reliability in disease detection. Utilizing evolved deep transfer learning models can notably amplify the efficiency and precision of disease identification processes, alleviating challenges faced in the dynamic environments of leaf disease detection and classification. The existing studies not only act as a repository of benchmarks and insights for forthcoming innovations like the MLTNet approach but also underscore the novelty and importance of this research within the expansive field of agricultural applications and advanced learning techniques.

2.3 Problem definition

This study's main problem is mango leaf disease detection and categorization, which are prevalent in Southern India and significantly impact the agricultural economy. Manual disease diagnosis in mango leaves relies on skilled visual examination, which is time-consuming, laborious, and error-prone, especially given the diversity of disease symptoms influenced by various climatic conditions.

3. MLTNET: mango leaf disease classification using transfer learning and visual explanations

Transfer learning methods are widely explored for mango leaf disease identification, facing challenges like potential misdiagnoses due to diverse leaf characteristics and environmental influences. Despite the importance of early detection, rural locales often experience delays due to time constraints and a shortage of agricultural expertise. Convolutional neural networks (CNNs) offer autonomous feature selection, reducing the need for labour-intensive image preprocessing in image-based recognition tasks. However, their effectiveness is hindered by the scarcity of large, diverse datasets for training. The proposed MLTNet (Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations), depicted in Fig. 1, addresses these challenges by combining transfer learning and visual explanations to enhance disease classification accuracy.

MLTNet, built on the ResNet50 architecture with Error Level Analysis for data preprocessing, incorporates various dense layers, dropout mechanisms, and regularization to improve accuracy. Integrating Grad-CAM for interpretability, MLTNet visually highlights important regions in input images influencing predictions. After several epochs of training, the model's accuracy on training, validation, and test sets outperforms state-of-the-art models and shows promise for agricultural leaf disease prediction and classification.

3.1 Dataset description

The south Indian area is home to a wealth of mango plant illnesses, and this collection of high-quality images captures them all. Leaf images with rust, bacterial spots, and powdery mildew are among the diseased leaves included in the collection. There

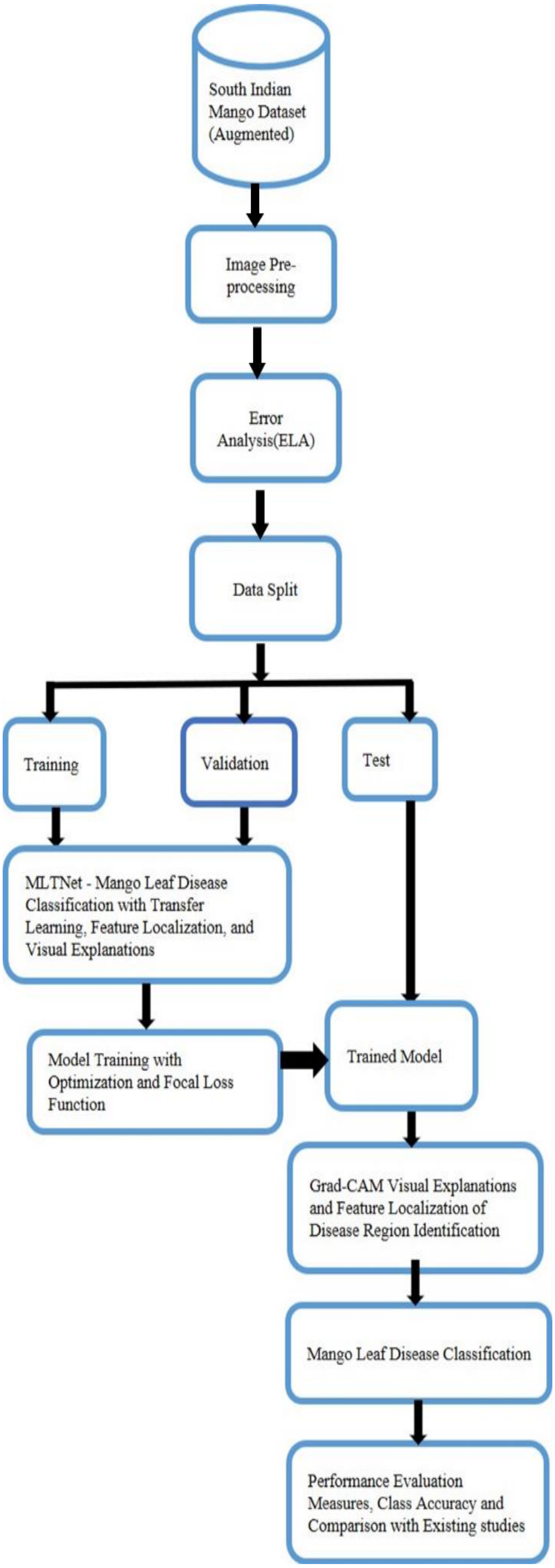
South Indian Mango Dataset (Augmented)
Image Pre-processing
Error Analysis(ELA)
Data Split
Training
Validation
Test
MLTNet - Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations
Model Training with Optimization and Focal Loss Function
Trained Model
Grad-CAM Visual Explanations and Feature Localization of Disease Region Identification
Mango Leaf Disease Classification
Performance Evaluation Measures, Class Accuracy and Comparison with Existing studies

Figure. 1 MLTNet - Mango leaf disease classification with transfer learning, feature localization, and visual explanations

are a total of 1,275 photos representing 14 distinct illnesses in the collection.

As the dataset is imbalanced, data augmentation is performed and the dataset count is increased to 820 in each class. Now the augmented South India dataset contains 11480 images with 14 disease classes.

3.2 Error level analysis (ELA)

As our proposed model predicts correctly classified images under visual predictions, it is customary to perform Error Level Analysis(ELA)[26]. ELA is a forensic image analysis technique used to detect digital image tampering and identify regions of an image that might have been altered or manipulated. It works on the principle that when an image is saved or compressed multiple times, the error levels introduced during the compression process are not uniform across the entire image. ELA calculates the error difference between the original picture and a compressed version to show these differences. Areas with significant modifications like cloning, retouching, or superimposition tend to exhibit higher error levels than unchanged regions.

ELA's primary purpose is to emphasize regions of an image that may have been compressed to varying degrees in comparison to the remainder of the image. The prospective areas of interest for future research may be indicated by the compression discrepancy, considering the possibility that these locations have been edited or altered. The ELA method evaluates image alteration and compression anomalies across a variety of JPEG quality levels. For improved visual presentation, the 'convert_to_ela_image' function generates an enhanced ELA image. In contrast, the 'compute_ela_cv' function calculates the precise discrepancy between the original and compressed images. Using the 'q' argument as control, the 'random_sample' function generates a grid of images depicting ELA effects at various compression levels. An increase in compression levels is denoted by a reduction in 'q' values, which visually emphasize compromised regions to facilitate forensic image integrity analysis.

$$ELA = ||I\_original - I\_compressed|| \tag{1}$$

A pixel's error level (ELA) is the difference in intensity values between the original and compressed picture. I_original : pixel intensity in the original picture. I_compressed : pixel's intensity in the compressed picture. This is how the ELA process is carried out. I take the original picture and duplicate it. ii Use a lossy compression technology, such JPEG, to compress the duplicate image to a predetermined compression quality. Subtract the intensity values of each matched pixel in the original and compressed photos to get each pixel's error level.Apply a color map to see mistake levels in color. This will highlight locations with varying error levels and point out possible modification areas.

3.3 Image processing

Pre-processing techniques were applied to enhance the final images in the Mango leaf disease dataset, aiming to improve classification accuracy and feature extraction. The transfer learning model's training involved multiple iterations, necessitating a large-scale image dataset to prevent overfitting and ensure reliable results.

For the South India mango leaf Disease Classification task, image pre-processing techniques played a vital role in isolating diseased leaves from background information before inputting them into deep learning models. This step significantly enhanced the accuracy of recognizing diseased leaves, contributing to improved performance. Image processing is a critical aspect of effective Disease classification of mango Leaf Using Transfer Learning, where various techniques can be applied to preprocess and enhance raw leaf images, improving the accuracy of transfer learning models [23]. Additionally, image compression techniques, including Lossless and Hybrid compression, were employed in our proposed work to achieve low-resolution images for practical deep-learning applications, facilitating efficient storage, transmission, and faster processing in leaf disease identification systems.

3.3.1. Lossless compression

Huffman and Run-Length Encoding are lossless compression methods. (RLE), preserve all the image's original information while reducing the file size [27]. These methods exploit redundancy in the image data to achieve compression without sacrificing details. Lossless compression is suitable when an exact image representation is required for accurate leaf disease identification.

3.3.2. Hybrid compression

Hybrid compression techniques combine the strengths of both lossless and lossy-compression methods. They employ a lossless compression algorithm for preserving critical image information, such as disease-specific features, while applying lossy compression to less important image components. Image quality and compression efficiency are balanced [28]. The training and test photos were pre-processed to boost contrast and decrease file size to 224 by 224 pixels before analysis. We employ picture cropping in conjunction with the conventional techniques for rescaling (Eqs. 3 and 4), and closest neighbour interpolation (Eq. 2) to shrink the images.

$$NewPixelValue(x',y') = PixelValue(round(x), round(y)) \quad (2)$$

$$NewWidth = ScaleFactor * OriginalWidth \quad (3)$$

$$NewHeight = ScaleFactor * OriginalHeight \quad (4)$$

Leaf images captured in RGB are converted to grayscale. Edge of Caution To recognize the edges in a leaf image and alleviate the irritation, unambiguous evidence is utilized. The external designs in leaf images are equal in how they are perceived from the edge. When the upper shape is taken as (p,q), the breadth and the level are (r,s), and these four centres do not settle the bobbing. Each member of the upright hopping square is still a work in progress. The return on investment region is removed using the primary RGB leaf image's coordinates (p+r, q+s). Finally, the dreaded leaf symbol may be put to rest.

3.4 Data split into training, validation, and testing

Split ratios of 80%,10%, and 10% were used to separate the mango leaf disease dataset into training, validation, and testing sets, respectively. Deep learning model predictions were improved using Adam optimization with forward and backpropagation. Thus, deep learning model output accuracy was ensured. The validation and testing sets contained twenty percent of the training set, which included images of 14 classes of mango leaves. On the training dataset, MLTNet classified test dataset images and predicted class labels.

3.5 CNN layer

CNNs, initially designed for grid-based data like photos [30], interpret information in a grid-arranged manner, where pixels determine color and brightness. Neurons in each CNN layer recognizes simpler patterns before progressing to more complex ones, mirroring the brain's visual interpretation process. CNNs have input, hidden, and output layers. CNNs combine normalization, pooling, convolution, and fully-connected layers [30]. Image processing makes use of ReLU, and the convolutional layer filters provide feature maps for classification [27]. This paper introduces a robust CNN model for fine-grained leaf disease detection, involving pre-processing to reduce leaf image size and a second-level illness identification from images with a convolutional neural network-based transfer learning model [30]. The CNN model architecture includes Conv2D, Batch Normalization, Max Pooling, and Activation layers. Algorithm 1 presents the generalized pseudocode (MLTNet) for leaf disease classification, utilizing explainable transfer learning models for multi-class accuracy prediction. The hierarchical representations created by neural networks contribute to accurate categorization.

3.6 Constructing MLTNet model for south indian mango leaf disease classification using ResNet50 model

MLTNet is built as 54 layers to forecast leaf diseases and classification using a base model called ResNet50. A deep convolutional neural network architecture called ResNet50, or Residual Network with 50 layers, was first introduced as a member of the ResNet family [6]. ResNet50 is recognized for its innovation in enabling the training of extraordinarily deep neural networks with hundreds of layers, overcoming the difficulties of disappearing gradients and accuracy loss as networks go deeper. ResNet50's main innovation is the addition of skip connections, also known as residual connections, which let the network learn residual mappings rather than attempting to learn the underlying input-to-output mappings directly. To do this, shortcut connections that bypass one or more levels are added, the input is immediately propagated to deeper layers, and gradient flow is made more accessible when training. The network can learn the residual between the input and the intended output thanks to the skip connections, which makes it simpler for the model to learn and optimize incredibly deep structures. Due to its exceptional performance in several computer vision applications, such as picture segmentation, object detection, and classification, and its extensive use, ResNet50 has cemented its place as a critical deep learning architecture.

ResNet50 contains 16 residual blocks. Fig. 2 shows the MLTNet architecture. The simplified description of ResNet50 is as follows. The architecture of ResNet50 is divided into multiple stages, and each stage includes a varying number of residual blocks.

– Stage 1: The initial convolutional layer and max-pooling layer

- Stage 2: a total of 3 residual blocks, with 3 convolutional layers
- Stage 3: a total of 4 residual blocks, with 3 convolutional layers
- Stage 4: a total of 6 residual blocks, with 3 convolutional layers
- Stage 5: a total of 3 residual blocks, with 3 convolutional layers

After ResNet50, the model was optimized by adding a batch normalization a dense layer of 256, and a dropout of 40%. Finally, a dense layer with 14 represents the class labesl In the dense (256) layer, an L2 regularizer with 0.016 as kernel regularizer and an L1 regularizer of 0.06 is taken as both activity and bias regularizer [31].

The model leverages a pre-trained ResNet50 architecture as a base. This architecture includes 16 residual blocks that facilitate the learning of residual mappings, greatly lowering deep network vanishing gradients. Fig. 3 shows the Residual Block architecture used in ResNet50.

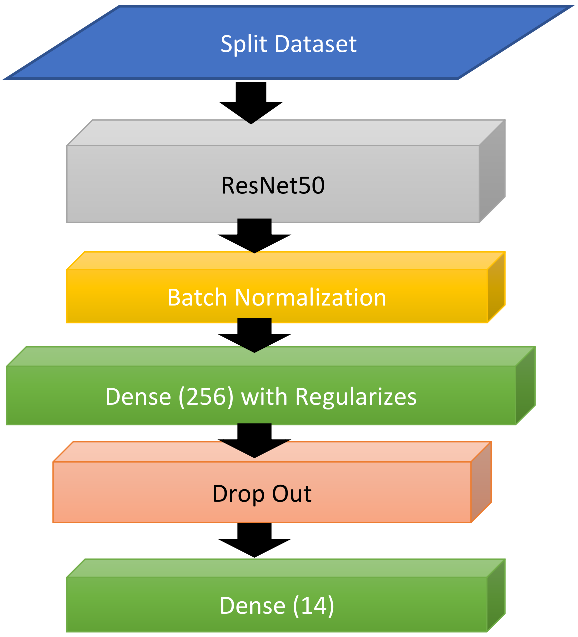

Figure. 2 MLTNet architecture

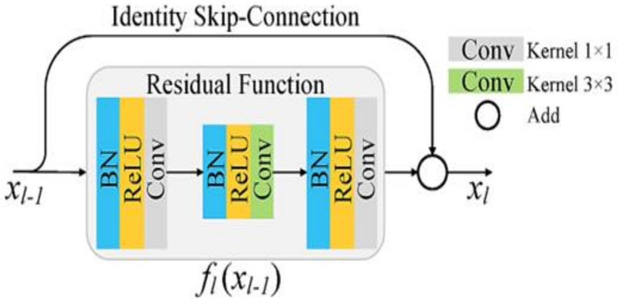

Figure. 3 Residual block architecture [6].

To preserve pre-trained features, ResNet50 layers are frozen ({model. trainable = False}) and loaded with ImageNet weights. We first applied Batch Normalization to the ResNet50's top layers, followed by a dense layer of 256 units. To prevent overfitting, dropout layers are added, and the final dense layer predicts the exact number of classes in our dataset. Using this strategy, the model can change to our classification aim while still utilizing the pre-trained ResNet50 features. The model's durability is enhanced via data augmentation techniques, and the Adamax optimizer manages the learning rate. The training process is monitored using a learning rate scheduler and early termination.We hope this clarifies the Transfer Learning strategy employed in our work. The MLTNet model incorporates modifications to the base ResNet50 architecture, including various sizes of dense layers and dropout mechanisms for accuracy enhancement, as well as the integration of Grad-CAM for interpretability.

The integration of Grad-CAM for feature localization and visual explanations in MLTNet is another significant advancement. This allows for better interpretability on decision-making, aiding in the identification of critical regions influencing classification decisions. Such insights are not only vital for understanding the model's behavior but also for improving trust in automated disease detection systems, particularly in agricultural applications. The enhanced data pre-processing, including Error Level Analysis and hybrid compression techniques, further contribute to improved data quality and model performance. Collectively, these modifications in MLTNet are designed to enhance classification accuracy and interpretability over the original ResNet50 structure.

3.7 Computer-Aided visualizations

Convolutional Neural Networks (CNNs) are integral to addressing complex computer vision challenges, such as image classification, detection, segmentation, and captioning. However, their intricate architecture poses a significant issue in model interpretation, hindering explainability—a crucial aspect of the evolution of AI [7]. Achieving explainability is particularly challenging in accurate deep learning models due to their layered structure and end-to-end training, complicating interpretability, especially in healthcare applications. To address this, techniques like Class Activation Maps, including Grad-CAM (Gradient-weighted Class Activation Mapping), have gained popularity for enhancing the explainability of computer vision models, revealing critical prediction areas in images.

3.7.1. Grad-CAM

In Convolutional Neural Networks (CNNs), researchers propose deeper architectures for higher-level visual representations. While convolutional layers retain spatial information, fully connected layers often result in its loss. However, CNN final layers balance geographical information with high-level semantics, with neuron units seeking class-specific semantic details. Grad-CAM [32] enhances interpretability by utilizing gradients within the last convolutional layer, highlighting the decision-making process for a specific area of interest through assigned prominence values to individual neurons. Grad-CAM's versatility extends to feature activation maps of any convolutional layer. To explain the activations of any layer, gradients are "global averaged pooled" over height and width dimensions. Our focus lies on the decision of the output layer, obtaining the localization map of the class through score gradient calculations.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_{i}\sum_{j}}^{\textit{global average pooling}} \quad \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\textit{gradients via backprop}} \qquad (5)$$

The equation (.5) represents a component of the Grad-CAM technique, which is used to visually explain the decisions made by convolutional neural networks. In this equation, α_k^c denotes the weights assigned to the neurons in the kth feature map for class *C*. These weights are calculated by globally averaging the gradients of the class score y^c—output before the softmax layer—relative to the feature map activations A_{ij}^k at each spatial location (i,j) which are obtained via backpropagation. The process highlights how much each part of the feature map contributes to the final decision for class c, enabling the creation of a heatmap that visualizes the important regions in the input image for predicting that class. This heatmap helps in understanding which features of the image lead the model to its classification decision, emphasizing areas with a strong positive influence on the class-specific output.

The final equation of the Grad-CAM is as follows,

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\textit{linear combination}} \qquad (6)$$

The generalized workflow of Grad-CAM has the following phases.

- Train ResNet50 for mango leaf disease classification.
- Obtain final convolutional layer output and predicted class scores from the trained model.
- Compute gradients of predicted class scores concerning final convolutional layer feature maps using Eq. (5).
- Weight feature maps by gradients to derive importance scores for each spatial location using Eq. (6).
- Generate a heatmap by combining weighted feature maps, highlighting the model's decision-making process and key input elements for the projected class.

Fig. 4 shows the Grad-CAM workflow in the CNN.

Grad-CAM is useful in localizing the regions, but it needs to explain the reasons for the model's prediction [7]. model's prediction of the regions the model looked at to make the final decision. Grad-CAM for improved feature localization and visual explanations, aiming not only to boost accuracy but also to enhance model interpretability and trustworthiness. Grad-CAM, a powerful technique in neural networks, is utilized for visualizing important regions using estimated gradients for each feature to locally visualize eq. (5) the model predictions. After training, it calculates gradients from intermediate layers, generating a heatmap that highlights crucial regions in input images.

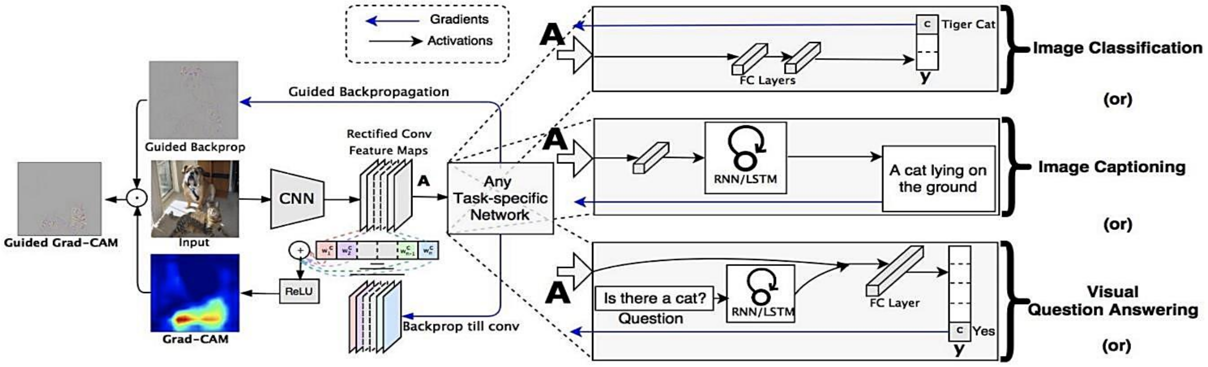

Figure. 4 The generalized workflow of GRAD-CAM [7]

MLTNet differs with other models, as MLTNet significantly advances agricultural AI by merging transfer learning with Explainable Artificial Intelligence (XAI) to address the limitations of manual and traditional automated methods. Unlike conventional deep learning models that start from scratch, MLTNet leverages pre-trained models specifically adapted for mango leaf disease contexts, enhancing learning efficiency and accuracy. It also integrates Explainable AI techniques such as Grad-CAM, which provides visual explanations of the model's decisions, boosting transparency and trust among users—a crucial factor for adoption in agricultural settings. Additionally, by incorporating advanced pre-processing techniques like Error Level Analysis, MLTNet adeptly manages variations in image quality and environmental conditions typical of field-collected data. This approach not only ensures high accuracy but also fosters user trust and interpretability, positioning MLTNet as a substantial improvement over existing methods in plant disease detection.

3.8 Optimizers

During the development phase of the model, the performance was assessed using three different optimizers as delineated in reference [33]. The specific optimizers utilized are as follows:

– Stochastic Gradient Descent (SGD)
- Adam
- RMSprop

3.8.1. Adam

Adam is a prevalent optimization strategy utilized to modify weights in neural networks throughout the training phase, as stated in reference [33]. Adam is the consequence of two SGD developmentsUsing the "Adaptive Gradient Algorithm" (AdaGrad), parameter learning rate is modified, resulting in enhanced performance in computer vision and natural language processing, where gradients with sparse distributions are prevalent.

3.8.2. RMSProp

Another optimization approach, root-mean-square propagation (RMSProp), has a learning rate that is modified per the parameters mentioned in [33]. The calculation for the 'running average' is conducted as follows:

$$(w, t) \coloneqq (w, t-1) + (1-\gamma)(\nabla\ Q_i(w))^{2}. \tag{7}$$

Eq. (7) describes the update rule for RMSProp, a type of adaptive learning rate method; it adjusts the weights w based on a decaying average (controlled) by of squared gradients, enhancing the convergence speed particularly in deep learning scenarios

The updates to the learning parameters are conducted as follows:

$$w \coloneqq w - \frac{\eta}{\sqrt{v(w,t)}} \nabla\, Q_i(w) \tag{8}$$

Equation (8) describes an update step in RMSProp or Adam-like optimizers, where the model weights w are adjusted by subtracting a scaled gradient; this scaling factor is the learning rate η divided by the square root of a moving average of squared gradients $v(w, t)$, stabilizing and improving the optimization process by adapting the learning rate to the parameters.

3.8.3. Stochastic gradient descent (SGD)

This approach employs an 'iterative method' for optimizing the loss function, which showcases differentiable properties as outlined in reference [33]. A central objective of machine learning is to fine-tune the loss or objective function. From a mathematical standpoint,

$$Q(W)=\frac{1}{n}\sum_{i-n}^{n} Qi(W) \tag{9}$$

Equation (9) defines the average loss function $Q(W)$ over n data points, where $Qi(W)$ represents the loss for the i-th data point given the model parameters W, providing a measure of the overall performance of the model across the entire dataset.

In this scenario, "w" is the estimated variable aimed at minimizing Q. Given its iterative nature, this method undertakes the following cycles to lessen the objective function:

$$w \coloneqq w - \eta \nabla Q(w). \tag{10}$$

where η is the learning rate.

Equation (10) represents the basic update rule in gradient descent optimization, where the model parameters w are iteratively adjusted by subtracting a product of the learning rate η and the gradient of the loss function $\nabla\ Q(w)$ to minimize the loss and improve model performance.

3.8.4. Loss function

Deep learning repeatedly refines its model by measuring the error between predictions and actual values using a loss function to reduce loss. Depending on the task and model behavior, cross-entropy, mean absolute error, and mean square error are common loss functions. Categorical or weighted cross-entropy were evaluated for extremely unbalanced data. Weighted categorical cross-entropy provides classes with fewer samples greater weight, although "focal loss" [34], meant to address class imbalance, worked better. Define the binary focal loss as,

$$FL(pt) = -(1 - pt)\,\gamma\log(pt) \tag{11}$$

Where $\gamma \geq 0$ and is termed as "focusing parameter" and

$$Pt = \left\{ \begin{matrix} p, y = 1 \\ 1 - p, Otherwise \end{matrix} \right\}$$

Incorrect identification of an input image has no effect on the loss function, as the adjustment factor increases to 1 and the term pt becomes 0. Successful categorization of the input image will result in Pt = 1 and the adjustment factor = 0. The loss value will "downweight" the input image as it approaches zero. The focusing parameter **P** controls the de-emphasis of easy-to-classify input pictures.

3.9 Evaluation metrics

The original dataset was enhanced by incorporating images modified through various alterations, as cited in reference [31, 32]. These alterations included horizontal flips, adjustments with a rotational scope of up to 90 degrees, and maintaining a zoom scope at 0.2. Furthermore, before their integration into the model, a rescaling process was performed on the images.

3.10 Regularization

3.10.1. L1 regularization

Lasso regularization (Least Absolute Shrinkage and Selection Operator), is a technique used in machine learning to prevent overfitting and to create more parsimonious models, which are models that achieve a good predictive performance with fewer variables [31]. The mathematical representation of this regularization term is given by:

$$L_1 = L_{Originl} + \lambda||W||_1 = \sum_{i=1}^{n} |w_i| \tag{12}$$

where:

- L_1: is the L1 regularization term
- λ: is the regularization parameter, a non-negative tunable parameter That scales the contribution of the L1 term in the loss function.
- Σ|wi|: is the summation of the absolute values of the weights

3.10.2. L2 regularization

L2 regularization serves as a preventive measure against overfitting in models, where overfitting is characterized by a rising validation loss despite a decreasing training loss. This discrepancy indicates that the model accurately represents the training data but struggles to generalize effectively on validation data, impacting its real-world applicability as emphasized in [31]. Consequently, the modified objective of the machine learning model post-regularization implementation is,

$$minimize(Loss(Data|Model) + complexity(Model)) \tag{13}$$

Eq. (13) expresses the fundamental goal of regularization in machine learning, which aims to optimize a model by balancing the fitting of training data (as measured by a loss function) against the model's complexity (typically quantified by regularization terms) to prevent overfitting and enhance generalizability to new data.

In the study undertaken, the complexity of the deployed models was mitigated through the implementation of L2 regularization. The optimization technique is encouraged to discover reduced weight values by representing this approach as the square of the weight coefficient magnitude. This approach discourages overfitting by forbidding complicated models that match the noise in the training data, resulting in a more generic and resilient model for unknown data.

$$L_2 \ regularization \ term = \| \boldsymbol{w} \|_2^2 = w_1^2 + w_2^2 + \cdots + w_n^2 \tag{14}$$

The L2 regularization term in (Eq. (14)) penalizes the sum of the squares of the model parameters, encouraging smaller, more diffuse weight values across the model to avoid overfitting and promote model simplicity. Within the configured models, a dual layer of L2 regularization was implemented preceding the ultimate SoftMax layer.

Algorithm 1 : Generalized Pseudo code MLTNet for Mango Leaf disease classification utilizing feature localization and visual explanations
Input: South India Mango Dataset (Augmented)

Output: Mango Leaf Disease Classification among 14 classes

Step 1: Acquire the mango leaf images with diseases like Anthracnose, Apoderus, Leaf Webber, Mango Sooty, Phoma, Powdery, etc.

Step2: Loading the data

Step 3: Put the correct labels on the pictures of the leaf images.

Step : Perform Error Level Analysis()

Step 4: Sort images into categories using the available class labels from the training and testing datasets.

```
Preprocess the images and perform Data Augmentation
Algorithm: Define function CreateImageDataGenerator
1. Start
2. Import ImageDataGenerator from keras. preprocessing.image
3. Define function CreateImageDataGenerator with setting different attribute parameters.
  Create an ImageDataGenerator instance with the parameters and Return the
       ImageDataGenerator instance
            4. End
```

Divide the data into training and testing.

Step 5: Create MLTNet model()
Initialize the parameters image size, epochs, batch size, and train and test image labels.

Step 7: Evaluate the trained model using a separate testing dataset.
alculate the test loss and accuracy of the model.

Step 8: Perform Grad-CAM for visual explanations and feature localization to determine significant regions.

Step 9: Check the accuracy of the proposed models, and see how they stack up against the rest of the other models out there.

Algorithm 1 : Error Level Analysis()

Input : sample image in the dataset

Output : Compression Quality of the JPEG leaf image

```
1. Import required libraries.
2. Define a function to compute Error Level Analysis (ELA) using OpenCV.
   def compute_ela_cv(path, quality):
       orig_img = cv2.imread(path)
       orig_img = cv2.cvtColor(orig_img, cv2.COLOR_BGR2RGB)
       cv2.imwrite('temp_file_name.jpeg',orig_img, [cv2.IMWRITE_JPEG_QUALITY, quality])
       compressed_img = cv2.imread('temp_file_name.jpeg')
       diff = 15 * cv2.absdiff(orig_img, compressed_img)
       return diff
3. Define a function to convert an image to ELA image using PIL.
   def convert_to_ela_image(path, quality):
    ela_image = ImageChops.difference(image, temp_image)
    ela_image = ImageEnhance.Brightness(ela_image).enhance(scale)
       return ela_image
4. Define a function to randomly sample an image from a directory.

5. Use the functions in a script or interactively for ELA analysis.
   selected_image_path = random_sample('/path/to/images')
   quality_level = 90  # Adjust the quality level as needed
   ela_result_cv = compute_ela_cv(selected_image_path, quality_level)
   ela_result_pil = convert_to_ela_image(selected_image_path, quality_level)
for i in range(1, columns * rows + 1):
    quality = init_val - (i - 1) * 8
    compute_ela_cv( ) and show the computed ele images.
```

Algorithm2 : MLTNet()

Input : Split dataset

Output : Accuracy Measures

```
1. Set image size, channels, and create image shape.
   img_size = (224, 224)
   channels = 3
   img_shape = 224x224
```

```
2. Determine the number of classes for the dense layer.
   class_count = 14.
3. Create a pre-trained ResNet50 model with weights from ImageNet.
4. Set the ResNet50 model as non-trainable.
   model.trainable = False

5. Construct the MLTNet model using Sequential API.
   MLTNet = Sequential([
       model,
       BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001),
       Dense(512, kernel_regularizer=regularizers.l2(l=0.018), activation='relu'),
       Dropout(0.5),
       Dense(256, kernel_regularizer=regularizers.l2(l=0.018), activation='relu'),
       Dropout(rate=0.5, seed=123),
       Dense(14, activation='softmax')
   ])
6. Define data augmentation using ImageDataGenerator.
   data_generator = ImageDataGenerator(
       rotation_range=20,
       width_shift_range=0.2,
       height_shift_range=0.2,
       horizontal_flip=True
   )
7. Compile the MLTNet model with specified optimizer, loss function, and metrics.

MLTNet.compile(Adamax(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
8. Display the summary of the MLTNet model.
   MLTNet.summary()
9. Set up callbacks for early stopping and learning rate scheduling.
   EarlyStopping(monitor='val_loss', patience=10)
   def scheduler(epoch, lr):
       return lr if epoch < 10 else lr * tf.math.exp(-0.1)
   lr_scheduler = LearningRateScheduler(scheduler)
10. Set batch size and number of epochs for training.
    batch_size = 32
    epochs = 20
11. Train the MLTNet model on the training generator with validation data and callbacks.
    history = MLTNet.fit(
        x=train_gen,
        epochs=epochs,
        verbose=1,
        validation_data=valid_gen,
        callbacks=[early_stopping, lr_scheduler],
        validation_steps=None,
        shuffle=False
    )
End
```

Algorithm 3: Implement Grad-CAM

```
1. Start
2. Import required libraries: os, random, numpy, tensorflow, cv2,
    matplotlib.pyplot
3. Define the data directory 'data_dir' containing subdirectories for each class
5. Create a list 'class_dirs' of paths to each class directory in 'data_dir'
6. Load the MLTNet() model with ImageNet weights
7. Define the target convolutional layer name 'target_conv_layer'
8. Define function 'apply_gradcam' with parameters img_path, model,
    last_conv_layer_name:
   a. Load and preprocess the image
   b. Create a model 'grad_model' with outputs from last_conv_layer_name and
       the final output
   c. Use GradientTape to compute gradients concerning the convolutional
       output
   d. Pool gradients and multiply with the output from the convolutional layer to
       get heatmap
   e. Normalize and apply a jet colormap to the heatmap
   f. Superimpose the heatmap on the original image
   g. Return the superimposed image, predicted class index, and predicted
       probability
```

```
9. Define function 'apply_random_gradcam' with parameters class_dir, model,
   target_conv_layer:
  a. Select a random image from the class directory
  b. Apply 'apply_gradcam' to the selected image
  c. Display the image with prediction information
10. Apply 'apply_random_gradcam' to random images from each class in 'class_dirs'
11. End
```

4. Results and discussions

Python libraries, including TensorFlow [35] and Keras, were employed to implement the research, focusing on transfer learning models. When training, we employed the Adam optimizer, which has an integrated loss function and a fixed learning rate. The experimental analysis took place an i5 processor and 8GB RAM with GPUenabled server.

The key outcomes of the MLTNet model encompassed several aspects:

- Categorizing mango leaf images into various diseases and predicting groups such as Anthracnose, Apoderus, Leaf Webber, Mango Sooty, Phoma, Powdery, etc.
- Evaluating the MLTNet model's performance across southern India Data set on mango leaf diseases using both the training and testing sets.
- Identifying and assessing feature localization, visual explanations, and significant regions contributing to mango leaf diseases through Grad-CAM visual predictions.
- Comparing the outcomes with other cutting-edge contemporary networks.
- Analyzing the results of experiments using explainable transfer learning for leaf disease identification in contrast to previous research.

4.1 Dataset description

A meticulous effort was dedicated to curate a comprehensive dataset featuring high-resolution images of mango leaves displaying diverse diseases. The images were carefully gathered from various locations across South India, capturing both healthy leaves and those afflicted with ailments such as bacterial spots, powdery mildew and rust. The original dataset comprises a total of 1,275 images, categorized into 14 distinct disease classes. The data augmentation technique is used to enhance a dataset that includes photographs of different plants and their associated illnesses, as part of a difficult data preparation assignment. The code uses Keras's `ImageDataGenerator` class to specify a series of real-time modifications, including 30 degrees of rotation, 20% width and height shifts, 20% shear transformation, 20% zoom, and horizontal flipping. Additionally, it uses the "nearest" neighbour approach to fill in any missing pixels. Every image is reviewed, checked for correctness, downsized to 256x256 pixels, and added to a list after being turned into an array using the provided procedure. These images are then divided by 255.0 to normalize to [0, 1]. After preprocessing, photos are augmented with alterations. A directory for each illness category has 820 pictures, including the originals. Plants and diseases remain hierarchical. The initial and enhanced picture quantities are shown on the monitor to track the advancement. Consequently, this meticulously augmented South Indian dataset now boasts a total of 11480 images, each meticulously categorized into one of the 14 disease classes. This augmentation not only balances the dataset but also enhances its diversity and coverage, thereby providing a more comprehensive and robust foundation for subsequent data analysis, experimentation, and machine learning model development.

The goal is to learn how to prepare pictures for transfer learning models. Because tensors are designed to store knowledge rather than actual data, they can be compared with multidimensional arrays. The necessary libraries are imported, and the dataset is loaded. This phase must be included in the data analysis process. There are many different image sizes and forms [35]. The first step in data pre-processing is to size each image evenly. New pictures with a resolution of 224x224 pixels are created using many enhancing approaches applied to the gathered photos using

4.2 Error level analysis

Error Level Analysis (ELA) identifies areas in an image that may have undergone alterations or different compression levels. In the provided code for a mango disease dataset, ELA is applied, displaying images with varying compression levels ("q" values). This visual inspection helps assess the impact of compression on image quality, determining an optimal level for subsequent image processing tasks. ELA aids in identifying any corrupted or altered images in the dataset, allowing for adjustments to the compression level during training. This process ensures data quality and integrity, crucial for subsequent image processing and machine learning tasks. Fig. 6 shows ELA analysis on the Apoderus leaf image.

Received: April 24, 2024. Revised: June 16, 2024.

4.3 MLTNet performance on south india mango leaf disease dataset

MLTNet trained on Leaf disease has reduced precision and caused loss. Table 2 displays the epochs, extra parameters, and MLTNet configurations. The dataset contains the MLTNet model, which adjusts internal parameters to enhance training performance.

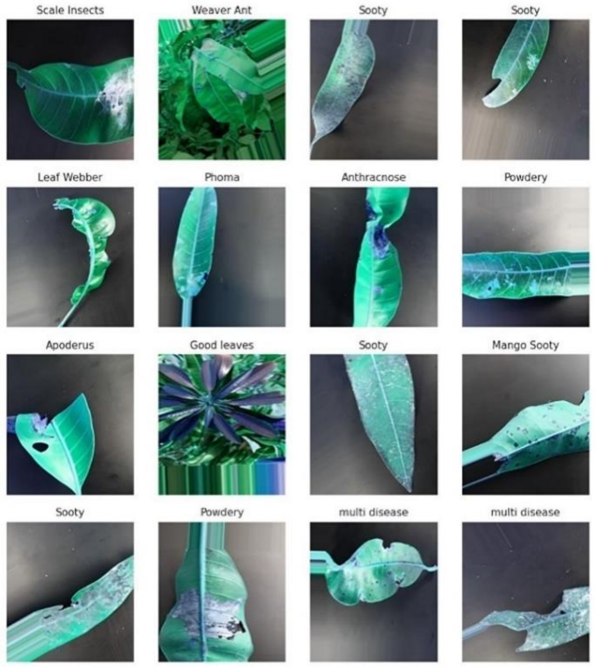

Figure. 5 Pre-processed mango leaf image (224x224).

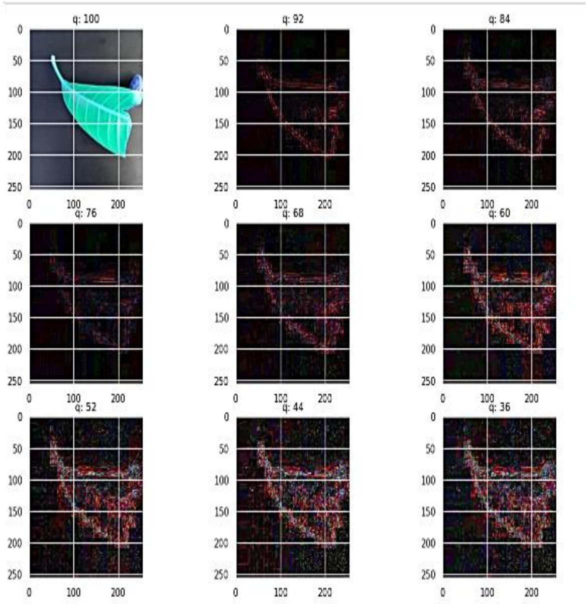

Figure. 6 The ELA analysis on the apoderus leaf image

Table 2. Explainable transfer-learning-model-training parameters

S.No	Parameter used	Value
1	Training Epochs	20 and 100
2	Optimizer	Adam
3	Learning Rate	0.001
4	Batch Size	32
5	Drop out	0.45
6	Image size	224x224
7	Image Depth	3
8	Early stopping	Yes
9	Reduce LR	Yes
10	Data Shuffle	True
11	No of Classes	14
12	Callback	True on model checkpoint
13	Loss function	Cross entropy
14	Activation Function	ReLU, Softmax
15	Regularization	L2

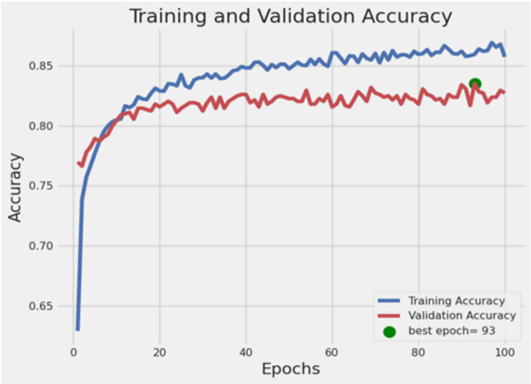

Figure. 7 Perofmance accuracy of the MLTNet model on the south india mango disease dataset for 100 epochs

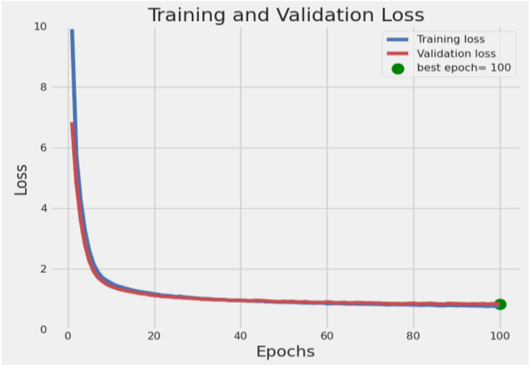

Figure. 8 Performance on MLTNet model South India Mango Dataset

Throughout multiple epochs, the model acquires the ability to identify important characteristics and categorize groupings of illnesses. The accuracy and loss statistics for each epoch indicate the model's performance. Fig. 7, 8, 9, and 10 demonstrate MLTNet architectures' Mango leaf disease classification accuracy for 100 and 20.

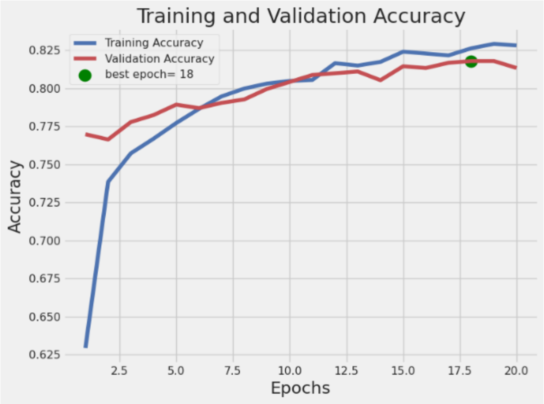

Figure. 9 Performance of accuracy of the MLTNet model on the south india mango disease dataset for 20 epochs

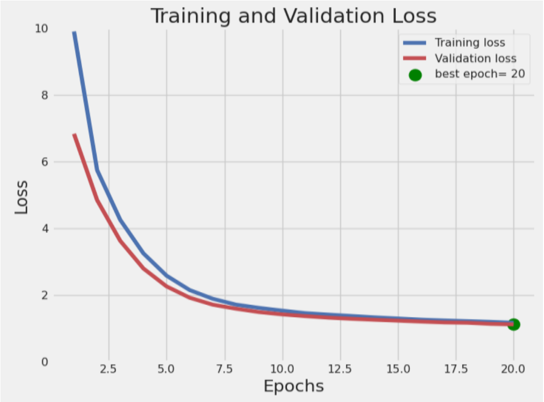

Figure. 10 Performance loss of the MLTNet model on the south india mango disease dataset for 20 epochs

4.3.1. Performance evalutaion of the MLTNet model on the south india mango leaf dataset

The proposed MLTNet model demonstrates promising performance on the South India Dataset for mango leaf disease classification, shown in Fig. 7 and 8. The MLTNet model showcased outstanding performance on the South India mango Disease Dataset, achieving a remarkable training accuracy of 92.8% and a shallow loss of 0.4294% after 100 epochs. The model also exhibited an average class accuracy of 94.33%. These remarkable results highlight the MLTNet model's ability to classify leaf diseases across multiple classes accurately. The model's exceptional accuracy and minimal loss values signify its proficiency in capturing intricate details and patterns in leaf disease images, enabling exact predictions. The MLTNet model's performance showcases its potential for improved plant health through automated leaf disease classification.

MLTNet model performance on the new plant disease dataset is promising after 20 training epochs, as depicted in Fig. 9 and 10. With a low training loss of 0.5355 and a high training accuracy of 85.12%, the model successfully learns mango leaf disease patterns from the training data. Additionally, impressive validation results include a loss of 0.6374 and an accuracy of 85.17%, showcasing the model's effective generalization to unseen data. MLTNet demonstrates strong potential for accurate and interpretable leaf disease detection and classification, positioning it as a valuable tool for precision agriculture and plant disease management. However, to enhance the model's credibility

and applicability in realworld scenarios, further testing on independent datasets and interpretability analyses are warranted.

4.3.2. Standard ResNet50 model performance on south india mango dataset

The standard ResNet50 model exhibits promising performance on the New Plant Disease Dataset for leaf disease classification, illustrated in Fig. 11 and 12. Trained for 70 epochs, the ResNet50 model achieves commendable accuracy on the South India Mango Leaf Disease Dataset at 81.81% with a loss of 0.4549.

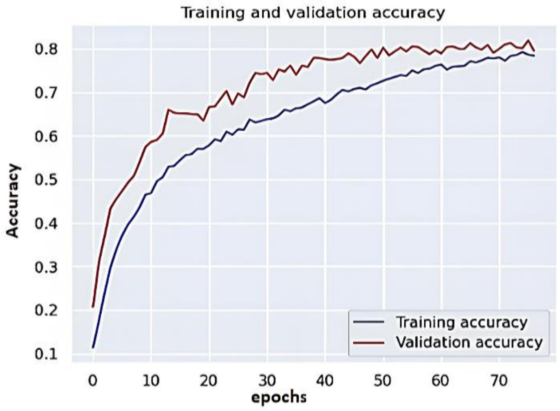

Figure. 11 Performanc Accuracy of the standard ResNet 50 model on plant disease dataset

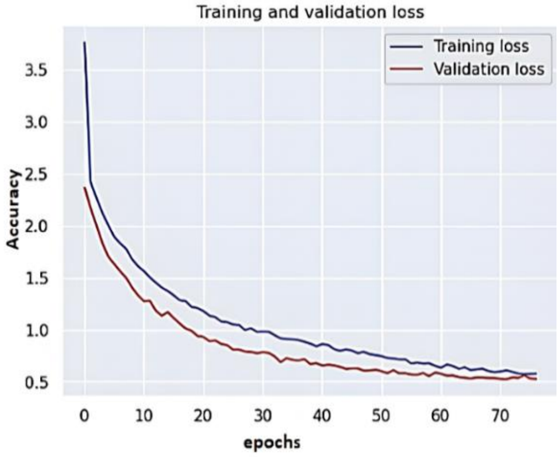

Figure. 12 Performance loss of the standard ResNet50 model on plant disease dataset

Received: April 24, 2024. Revised: June 16, 2024.

The average class accuracy of 77.99% further shows the model's ability to classify leaf diseases across classes. These observations show the model's effectiveness in capturing intricate patterns and features in mango leaf disease images, resulting in accurate predictions. The observed fluctuation in validation accuracy after 15 epochs is attributed to overfitting on the South India mango disease dataset, impacting the reliability of test predictions.

4.4 Performance comparison of proposed method and existing studies

Table 3 presents the results of several models applied to the South India mango leaf Disease dataset, including ResNet50, VGG-16, InceptionV3, DenseNet121, and the MLTNet model, following a predetermined number of training epochs.

In the evaluation of accuracy for mango leaf analysis, various deep learning models partake assessed, each exhibiting distinct levels of performance. Notably, the ResNet50 model achieved a commendable accuracy of 81.81%, showcasing its capability to distinguish between different disease categories. The VGG-16 model demonstrated a notable improvement, reaching an accuracy of 87.99%, underscoring its effectiveness in capturing intricate patterns within the leaf images. InceptionV3 further elevated the performance, achieving an impressive accuracy of 91.84%, which indicates its proficiency in distinguishing among the various disease classes. DenseNet121 continued this trend, achieving a commendable accuracy rate of 92%, demonstrating its robustness in disease classification tasks. The Deep CNN variant cited as [42] marginally exceeds these with a 92.42% accuracy, while the standard CNN [42] variants show a lower accuracy between 85.80% and 89%. The integration of transfer learning with attention mechanism yields an 86.1% accuracy, underscoring the nuanced impact of generative models on performance.

Table 3. Analysis of transfer models on new plant disease dataset performance.

Model/Technique	**Accuracy (%)**
ResNet50 [18]	81.81
VGG-16 [17]	87.99
InceptionV3 [20]	91.84
DenseNet121 [22]	92
Deep CNN [42]	92.42
CNN [22]	85.80
AlexNet[13]	89
Transfer Learning with Attention Mechanism [21]	86.1
MLTNet (Proposed)	92.8

This suggests that MLTNet's advanced features, possibly including specialized data preprocessing and enhanced learning mechanisms, are highly effective for this specific application. The CNN variants and Deep CNN, cited from [42], demonstrate varied results (85.80% to 92.42%), highlighting that adaptations to the base CNN architecture, such as the incorporation of attention mechanism, do not uniformly enhance performance, possibly due to the complexities in training CNNs.

Topping the chart is the MLTNet model, achieving the highest accuracy at 92.8%, hinting at the efficacy of its possibly innovative techniques in improving model performance. This achievement highlights the effectiveness of the proposed model, which likely incorporates advanced architectural features and optimization techniques, thereby enhancing its precision in classifying mango leaf diseases. These comparative outcomes underscore the implication of model selection in achieving higher accuracy in complex multi-class classification tasks, with MLTNet emerging as the top-performing solution.

4.5 MLTNet learning model accuracy on predicted images

The MLTNet model showcases its accuracy in classifying mango leaf diseases by performing well on correctly predicted images. A subset of test dataset images is randomly selected and displayed in a grid format, with actual labels in green and predicted labels in red for easy identification of misclassifications. This graphical depiction enables users to effectively examine model predictions, pinpointing areas that require enhancement or include inaccuracies.

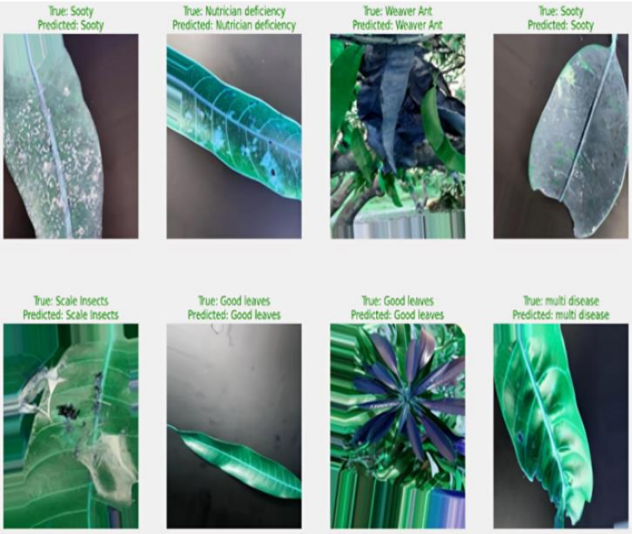

Figure. 13 MLTNet model performance on correctly predicted mango leaf images

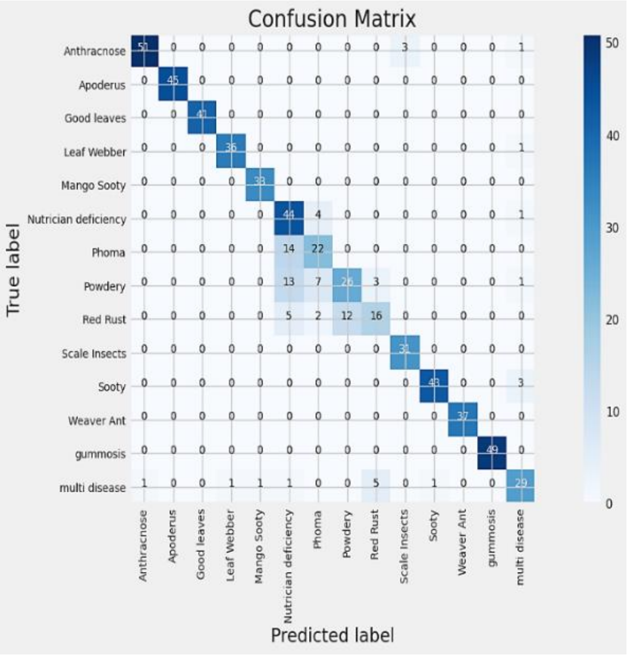

Figure. 14 Confusion matrix for south indian mango leaf disease on MLTNet model

The visualization functions as an intuitive instrument for identifying and resolving errors, refining, and acquiring a deeper understanding of the model's capabilities and limitations. Fig. 13 illustrates MLTNet's predictions on correctly classified images. The transfer learning model demonstrates impressive performance with an overall accuracy of 94.3% and a multi-class accuracy of 94%, showcasing its proficiency in capturing intricate patterns from leaf disease images. These results highlight MLTNet's efficiency in automated leaf disease detection and classification.

4.6 Evaluation of performance measures for detecting and classifying individual plant leaves

The performance of MLTNet on plant leaf diseases is assessed with precision, recall, and F1-score for each disease class. The classification metrics present detailed performance metrics for a multi-class classification model. It covers various classes, including "Anthracnose," "Apoderus," and "Mango Sooty," and assesses. The metrics of accuracy, recall, and F1-Score are provided for each class in the model. For example, the model successfully attained high precision (0.97) for "Anthracnose," indicating a low rate of false positives, while the recall (0.93) demonstrates its ability to capture the truest positives, resulting in an F1-Score of 0.95, signifying a balanced performance. In contrast, "Nutrician deficiency" shows a lower precision (0.89), indicating some false positives, but still maintains a good recall (0.95), resulting in an F1-Score of 0.83. The overall accuracy of the model is 0.94, showcasing its general effectiveness. When considering macro and weighted averages, which aggregate metrics across classes, both show strong performance, with F1-Scores of 0.94, highlighting the model's competence in handling this multi-class classification problem. Figure 14 displays the confusion matrix for the MLTNet model using the mango leaf illness dataset.

4.7 Grad-CAM analysis

MLTNet model had Grad-CAM added to them so that we could see where the neural networks were putting the most effort into feature localization and visual explanations into classifying leaf diseases. The image regions most important to Grad-CAM's prediction are highlighted in the resulting heatmaps. In the case of MLTNet, the Grad-CAM heatmaps provided clear indications of the crucial areas that influenced the model's decision-making process and the classification task. The visualization of these regions aided in the interpretability and explainability of the models, allowing agriculture professionals to gain insights into the models' reasoning and enhancing their trust in the model's predictions.

4.7.1. Visualization of the Grad-CAM technique of MLTNet

The MLTNet model was utilized to apply the Grad-CAM approach to an image of a leaf disease, revealing significant regions crucial for accurate prediction. Grad-CAM highlights discriminative features vital for precise predictions, especially in early leaf disease detection crucial for effective agriculture. MLTNet's accuracy after Grad-CAM reaches 94.3%, a 2% increase in leaf disease prediction. Fig. 15 illustrates the fine-tuned MLTNet's effectiveness with Grad-CAM in predicting disease regions. With an overall accuracy of 94% and high precision, recall, and F1 scores for each class, the model distinguishes various plant diseases effectively. Grad-CAM enhances interpretability, aiding researchers and farmers in detecting and localizing plant health issues for targeted treatments.

4.8 Discussion

The MLTNet model, proposed for mango leaf disease classification on the South India Dataset, has exhibited outstanding performance. Achieving a training accuracy of 92.8% and minimal loss of 0.4294% after 100 epochs, it effectively classifies

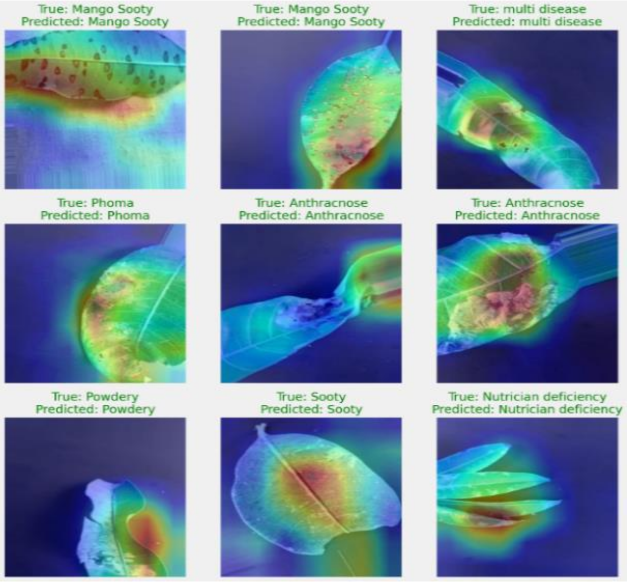

Figure. 15 The effectiveness of a fine-tuned MLTNet model with Grad-CAM for feature localization and visual explanations in accurately predicting significant disease regions on mango leaves

leaf diseases transversely multiple categories. Through an average class accuracy of 94.33%, the model demonstrates its proficiency in distinguishing between different disease types. These results highlight MLTNet's potential to advance plant health management and agricultural productivity through automated leaf disease detection and classification. In comparison, the standard ResNet50 model also demonstrates commendable accuracy in classifying mango leaf diseases, showcasing its ability to capture intricate patterns and features. However, its performance is impacted by overfitting on the South India mango disease dataset. Furthermore, the research compares with other existing studies on the dataset, showcasing the dominance of the projected MLTNet model. It outperforms ResNet50, VGG-16, and InceptionV3, achieving higher accuracy than these popular architectures. While DenseNet121 attains a similar accuracy, the proposed MLTNet model performs slightly better. The variant of Deep CNN referenced as [42] slightly surpasses these results with a 92.42% accuracy, while the conventional CNN [42] variants exhibit lower accuracies ranging from 85.80% to 89%. The incorporation of GANs in Deep CNN+GAN [42] results in an 86.1% accuracy, highlighting the nuanced influence of generative models on overall performance. Leading the rankings is the MLTNet (Proposed) model, attaining the highest accuracy of 92.8%, suggesting the potential effectiveness of its innovative techniques in enhancing model performance. These results affirm the effectiveness of the MLTNet model in plant disease classification and its potential as a promising approach for the accurate and reliable uncovering of plant infections in agricultural settings.

Comparatively, the MLTNet model outperformed its counterparts, achieving the highest accuracy of 94.3%. This highlights the significance of model selection in achieving higher accuracy in complex multi-class classification tasks, with MLTNet emerging as the top-performing solution. Additionally, the MLTNet model's predictions on correctly classified images and the use of the Grad-CAM visualization technique have provided valuable insights into its performance. These visual explanations of predictions enhance the model's interpretability and utility in early disease detection and precise localization on plant leaves.

5. Conclusion

In conclusion, our research presents the MLTNet model as a substantial advancement in the field of agricultural AI, specifically for the classification of mango leaf viruses. Through rigorous testing, MLTNet has demonstrated a notable classification accuracy of 94.3% on the training set and 86.3% on the test set, positioning it as a highly effective tool for plant disease management. These figures underscore the model's superior capability to not only learn from a comprehensive dataset of mango leaf images, which was enhanced through advanced preprocessing techniques like Error Level Analysis, but also to generalize well to new, unseen data.The integration of transfer learning with the ResNet50 architecture, combined with our novel application of Explainable Artificial Intelligence (XAI) techniques like Grad-CAM, has resulted in significant improvements in both interpretability and classification performance. This approach not only aids in the precise localization of disease symptoms on mango leaves but also enhances the transparency of the diagnostic process, making MLTNet a trustworthy tool for agronomists and farmers.Furthermore, comparative analyses with established models such as ResNet50, VGG-16, and InceptionV3, which achieved lower accuracies ranging from 81.81% to 91.84%, highlight the effectiveness of MLTNet. Our model's ability to outperform these benchmarks illustrates its potential to set a new standard in agricultural diagnostics.

As a future work, the MLTNet framework will be pivotal in shaping future research directions. We plan to expand its application to other crops and integrate real-time disease detection systems, which could revolutionize plant health management practices globally. The potential for MLTNet to contribute to sustainable agriculture is immense, underscoring the importance of continuous advancements in AI and machine learning in addressing critical global challenges.

Nomenclature / Notation list

A_{ij}^k: Activation of the k-th feature map at location (i, j) in a layer.
y^c: Output of the network for class c.
α_k^c: Weight of the k-th activation map for class c, computed by global average pooling of the gradients.
Z: Normalization factor in global average pooling, summing over all spatial locations (i, j).
w: Model weights or parameters.
η (eta): Learning rate used in the optimization algorithm.
L: Loss function used to evaluate the model's predictions.
Q(W): Quality or cost function calculated as an average loss over the dataset.
v(w, t): Denominator in the RMSProp update rule, which is the moving average of the squared gradients.
$\boldsymbol{\alpha_k^c}$: Weight of the k-th activation map for class c, computed as the global average of the gradients.
$\boldsymbol{y}$ **:**Output of the network for class c before softmax.
$\boldsymbol{A_{ij}^k}$: Activation of the k-th feature map at spatial location (i, j).
$\frac{\partial y^c}{\partial A_{ij}^k}$: Gradient of the class score ycyc with respect to the activation A_{ij}^k, calculated via backpropagation.
(w,t): Weights ww at iteration tt in an optimizer.
γ: Decay rate in RMSProp or similar adaptive learning rate methods.
∇Qᵢ(w): Gradient of the loss function Qi with respect to weights ww.
η: Learning rate.
∇Q(w): Gradient of the loss function QQ with respect to weights ww.
$\| \boldsymbol{w} \|_2^2$: L2 norm squared of the weight vector ww, used in L2 regularization.
FL(pt): Focal loss function for binary classification.
p_t: Adjusted prediction probability, dependent on the true label yy.
$||\boldsymbol{W}||_1$: L1 norm of the weight vector ww, used in L1 regularization.
$\boldsymbol{\lambda}$: Regularization parameter that scales the contribution of the L1 or L2 term in the overall loss function.
γ : Focusing parameter in the Focal Loss function
L_{Grad-CAM}^c: Localization map for class c, used in Grad-CAM
ReLU: Rectified Linear Unit, a function used to introduce non-linearity in the model.
L_1 and L_2: L1 and L2 regularization terms.

Conflicts of Interest

The authors declare they have no conflicts of interest to report regarding the present study

Author Contributions

Study conception and design: Shaik Thaseentaj, S. Sudhakar Ilango; data collection: Shaik Thaseentaj; analysis and interpretation of results: Shaik Thaseentaj, S. Sudhakar Ilango; draft manuscript preparation: Shaik Thaseentaj, S. Sudhakar Ilango. All authors reviewed the results and approved the final version of the manuscript.

References

[1] K. Srunitha and D. Bharathi, "Mango leaf unhealthy region detection and classification", In: *Computational Vision and Bio Inspired Computing*, New York, NY, USA, pp. 422–436, 2018.

[2] U.P. Singh, S.S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease", *IEEE Access*, Vol. 7, pp. 43721-43729, 2019.

[3] N.F. Rosman, N.A. Asli, S. Abdullah, and M. Rus-op, "Some common disease in mango", In: *AIP Conference Proceedings*, Melville, NY, USA, p. 020019, 2019.

[4] Y. Nagaraju, T.S. Sahana, S. Swetha, and S.U. Hegde, "Transfer Learning based Convolutional Neural Network Model for Classification of Mango Leaves Infected by Anthracnose", In: *Proc. of the IEEE International Conference for Innovation in Technology (INOCON)*, Bengaluru, India, pp. 1–7, 2020.

[5] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models", *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of*

Artificial Intelligence (AI) on Communication Networks and Services, Vol. 1, pp. 1-10, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.

[7] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization", In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 618–626, 2017.

[8] S.P. Mohanty, D.P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection", *Frontiers in Plant Science*, Vol. 7, pp. 1–10, 2016, doi: 10.3389/fpls.2016.01419.

[9] M.R. Mia et al., "Mango leaf disease recognition using neural network and support vector machine", *Iran Journal of Computer Science*, Vol. 3, No. 3, pp. 185-193, 2020.

[10] S. Gupta, R. Sharma, and P. Singh, "Deep Learning-Based Mango Leaf Disease Detection and Classification Using Transfer Learning", *International Journal of Computer Applications*, Vol. 179, No. 11, pp. 15-20, 2020.

[11] A.B. Patil and S.D. Khirade, "Plant disease detection using image processing", In: *Proc. of the International Conference on Computing Communication Control and Automation*, 2015, doi: 10.1109/iccubea.2015.153.

[12] M. Merchant, V. Paradkar, M. Khanna, and S. Gokhale, "Mango Leaf Deficiency Detection Using Digital Image Processing and Machine Learning", In: *Proc. of 2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, doi: 10.1109/i2ct.2018.8529755.

[13] S. Arya and R. Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango Leaf ", In: *Proc. of 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2019, doi: 10.1109/icict46931.2019.8977648.

[14] J. Chen, J. Zhang, D. Zhang, Y. Sun, and Y.A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification", *Computers and Electronics in Agriculture*, Vol. 173, 2020, doi: 10.1016/j.compag.2020.105393.

[15] U.S. Rao et al., "Deep Learning Precision Farming: Grapes and Mango Leaf Disease Detection by Transfer Learning", *Global Transitions Proceedings*, Vol. 2, No. 2, pp. 535-544, 2021, doi: 10.1016/j.gltp.2021.08.002.

[16] J. Chen, D. Zhang, M. Suzauddola, Y.A. Nanehkaran, and Y. Sun, "Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning", *IET Image Processing*, Vol. 15, No. 5, pp. 1115-1127, 2021, doi: 10.1049/ipr2.12090.

[17] X. Li et al., "The Detection Method of Potato Foliage Diseases in Complex Background Based on Instance Segmentation and Semantic Segmentation", *Frontiers in Plant Science*, Vol. 13, 2022, doi: 10.3389/fpls.2022.899754.

[18] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning", *Computers and Electronics in Agriculture*, Vol. 186, 2021, doi: 10.1016/j.compag.2021.106184.

[19] M.H. Saleem, J. Potgieter, and K.M. Arif, "Plant Disease Detection and Classification by Deep Learning", *Plants*, Vol. 8, p. 468, 2019, doi: 10.3390/plants8110468.

[20] V. Sravan et al., "A deep learning-based crop disease classification using transfer learning", *Materials Today: Proceedings*, 2021.

[21] S.J. Pan and Q. Yang, "A survey on transfer learning", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345-1359, 2010.

[22] I.Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using ResNet50", In: *Proc. of 2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 2019.

[23] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases", *Computers and Electronics in Agriculture*, Vol. 72, pp. 1-13, 2010.

[24] S.M. Hassan et al., "Identification of plant-leaf diseases using CNN and transfer-learning approach", *Electronics*, Vol. 10, No. 12, p. 1388, 2021.

[25] Anagnostis et al., "A deep learning approach for anthracnose infected trees classification in walnut orchards", *Computers and Electronics in Agricultur*e, Vol. 182, p. 105998, 2021.

[26] J. Roy, "Error Level Analysis," 2021. Available: https://github.com/jayant1211/Image-Tampering-Detection-using-ELA-and-Metadata-Analysis.

[27] E. Hernández-Valencia, F.J. Gallegos-Funes, J. Olivares-Mercado, and V. Medina-Bañuelos, "Lossless compression for multispectral images of agricultural products based on 3D DCT", *Journal of Applied Research and Technology*, Vol. 18, No. 4, pp. 301-309, 2020.

[28] X. Yao, X. Zhou, S. Lu, and R. Zhang, "Hybrid compression algorithm for remote sensing images based on JPEG and fractal compression", In: *Proc. of the 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 128-132, 2020.

[29] G. Sun and X.J. Geng, "Plant Diseases Recognition Based on Image Processing Technology", *Hindawi Journal of Electrical and Computer Engineering*, Vol. 2018, 2018.

[30] M.K. Gurucharan, "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network", *UpGrad Blog*, 2020. Available: https://www.upgrad.com/blog/basic-cnn-architecture.

[31] D.M.W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", *International Journal of Machine Learning Technology*, pp. 37-63, 2011.

[32] E. Daglarli, "Explainable Artificial Intelligence (xAI) Approaches and Deep Meta-Learning Models for Cyber-Physical Systems", In: *Proc. of Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, A.K. Luhach and A. Elci, Eds., Hershey*, PA, USA, 2021, pp. 42–67.

[33] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, Vol. 13, No. 10, pp. 281–305, 2012.

[34] T.-Y. L. et al., "Focal Loss for Dense Object Detection", *13C-NMR Nat. Prod.*, pp. 30–33, 1992, doi: 10.1007/978-1-4615-3288-0_5.

[35] Gulli and S. Pal, Deep Learning with Keras: Implement Neural Networks with Keras on Theano and TensorFlow, Birmingham, UK, 2017.

[36] Modi et al., "Role of Artificial Intelligence in Detecting Colonic Polyps during Intestinal Endoscopy", *Engineered Science*, Vol. 20, pp. 25–33, 2022, doi: 10.30919/es8d697.

[37] M. Sudhi et al., "Advancements in Bladder Cancer Management: A Comprehensive Review of Artificial Intelligence and Machine Learning Applications", *Engineered Science*, Vol. 26, p. 1003, 2023, doi: 10.30919/es1003.

[38] Bhatt and A. Ganatra, "Explosive Weapons and Arms Detection with Singular Classification (WARDIC) on Novel Weapon Dataset Using Deep Learning: Enhanced OODA Loop", *Engineered Science*, Vol. 20, pp. 252–66, 2022, doi: 10.30919/es8e718.

[39] Y. Qi and C. Liu, "Deep Learning for Medical Materials: Review and Perspective", *ES Materials & Manufacturing*, Vol. 12, pp. 17–28, 2021, doi: 10.30919/esmm5f426.

[40] P. S. et al., "Face Detection Using Deep Learning to Ensure a Coercion Resistant Blockchain-Based Electronic Voting", *Engineered Science*, Vol. 16, pp. 341–53, 2021, doi: 10.30919/es8d585.

[41] D. Sharma et al., "A Convolutional Neural Network Based Deep Learning Algorithm for Identification of Oral Precancerous and Cancerous Lesion and Differentiation from Normal Mucosa: A Retrospective Study", *Engineered Science*, Vol. 18, pp. 278–87, 2022, doi: 10.30919/es8d663.

[42] S. Thaseentaj and S.S. Ilango, "Deep convolutional neural networks for south Indian mango leaf disease detection and classification", *Computers, Materials & Continua*, Vol. 77, No. 3, pp. 3593–3618, 2023.