Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

Deep Convolution Self-Attention Based Cascaded Autoencoder Network for Plant Disease Detection

P.K.Midhunraj[1], K.S.Thivya[2] , M. Anand[3]

Research Scholar[1], Professor[2&3]

Department of Electronics and Communication Engineering[1,2&3]

Dr. MGR. Educational and Research Institute[1,2&3], Chennai 95

Abstract

Early detection and treating the disease are the major need to enhance the growth and yield of agricultural plants. In general, manual monitoring of plant diseases will not provide exact outcomes. Thus, it is crucial for developing an automated system for detecting and classifying varied types of plant diseases. In recent studies, several models were utilized for classifying the varied categories of plant diseases in different species. However, the existing studies faced various limitations such as higher over fitting issues, computational complexity, reduced learning ability, etc. Therefore, the proposed study aims to develop an efficient hybrid deep learning model for enabling plant disease classification.

The steps that involved in this study are image acquisition, pre-processing, feature extraction, and classification. Initially, the input samples are gathered from the publicly available dataset containing images of diseased plants from various species including Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. Then, pre-processing is done to eliminate the noises and enhances the quality of inputs using Upgraded Gradient-based Guided Filtering (UG_GFil) method. For segmenting the affected portions in the diseased plant, Upgraded k-means clustering is used. From the segmentation samples, the needed features are extracted by introducing a novel Feature Attentional EfficientNetB0 model. After feature extraction, the relevant features are selected using Tuna Swarm Optimization. Then the types of diseases from varied species of plants are categorized by proposing a new Deep Convolution Self-Attention based Cascaded Autoencoder Network (DCSA_CAEN) model. Also, the parameters of proposed classifier are optimally tuned by using Binary Kookaburra Optimization (BKO) approach. Thus, the proposed study effectively classifies the presented plant diseases from the given input dataset.

Keyword - Hybrid deep learning, Gradient-based guided filtering, K-means clustering, EfficientNetB0, Tuna Swarm Optimization, Deep Convolutional Self-Attention, Cascaded Autoencoder Network and Binary Kookaburra Optimization.

Introduction

The agricultural sector plays a pivotal role in ensuring food security and sustaining livelihoods worldwide. However, one of the primary challenges faced by farmers and agriculturalists is the persistent threat posed by plant diseases. Various species of plants, including Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and others, are susceptible to a range of diseases that can severely impact their growth, yield, and overall health. These diseases, caused by bacteria, fungi, viruses, and other pathogens, manifest in diverse symptoms such as leaf spots, wilting, discoloration, necrosis, and deformities, ultimately leading to reduced crop quality and economic losses.

For instance, the Tomato Yellow Leaf Curl Virus (TYLCV) affects tomatoes, leading to stunted growth, curling of leaves, and reduced fruit yield. Similarly, the Potato Late Blight caused by the fungus Phytophthora infestans results in dark lesions on potato foliage, tuber rot, and significant yield losses if left untreated. Moreover, diseases like Powdery Mildew, Downy Mildew, Anthracnose, and Rust commonly afflict crops such as grapes, cucurbits, and citrus fruits, causing defoliation, fruit rot, and decreased market value.

Identifying and managing these diseases in a timely manner is crucial for minimizing crop losses and ensuring sustainable agricultural production. Early detection of symptoms enables farmers to implement preventive measures such as applying fungicides, adopting resistant cultivars, and implementing cultural practices like crop rotation and sanitation. However, manual inspection of plants for disease symptoms is labor-intensive, subjective, and often prone to human error. Traditional methods of disease identification are time-consuming. With the advent of advanced technologies, there has been a growing interest in developing automated systems capable of efficiently detecting and classifying plant diseases. However, in recent years, deep learning-based approaches have shown promising results in various domains, including computer vision and pattern recognition. These techniques leverage the power of artificial neural networks to automatically learn and extract relevant features from raw data, making them well-suited for complex tasks such as plant disease detection.

Despite the progress made in the field, several challenges persist. One such challenge is the high dimensionality of image data, which can lead to computational inefficiencies and over fitting. Additionally, accurately identifying diseased regions within plant images and distinguishing between different types of diseases pose significant challenges due to variations in appearance and environmental conditions.

To address these challenges, automated systems for plant disease detection and classification have emerged as promising solutions. Leveraging advancements in computer vision, machine learning, and deep learning, these systems enable rapid and accurate identification of disease symptoms from digital images of plant leaves. By analyzing the spatial patterns, textures, and color variations in the images, these systems can distinguish between healthy and diseased plants, facilitating timely intervention and management strategies. In this context, the proposed study aims to develop a sophisticated deep learning framework tailored for the detection and classification of plant diseases across a diverse range of species. By integrating cutting-edge techniques in image preprocessing, feature extraction, feature selection, and classification, the framework seeks to overcome the challenges associated with variability in plant diseases and environmental conditions. Through the utilization of advanced algorithms such as upgraded gradient-based guided filtering, upgraded K-means clustering, and feature attentional EfficientNetB0 model, the framework aims to enhance the accuracy, efficiency, and scalability of plant disease diagnosis.

In this paper, we present the methodology employed in our approach, including details of the datasets used, preprocessing techniques applied, and the architecture of the proposed deep learning model. We also discuss the expected outcomes of our study and highlight the potential impact of our research on agriculture and food security.

By focusing on a comprehensive set of plant species commonly affected by diseases, including apples, blueberries, cherries, corn, grapes, oranges, peaches, peppers, potatoes, raspberries, soybeans, squash, strawberries, and others, the proposed framework aims to provide a holistic solution for agricultural disease management. By empowering farmers and agriculturalists with automated tools for disease detection, the framework seeks to contribute to sustainable agriculture, food security, and economic prosperity in farming communities worldwide.

Related Works

A self-attention convolutional neural network for crop leaf disease recognition, leveraging attention mechanisms to enhance feature extraction and classification was discussed by Weihui Zeng and Miao Li. A neural machine translation model that learns to align and translate input sequences simultaneously; introducing the concept of attention mechanisms in neural networks was implemented by Dzmitry Bahdanau., et al. The limited applicability of machine translation concepts to the domain of plant disease recognition, required adaptation and

---

customization for their specific task. Grinblat G.L., et al discussed about the use of deep learning for plant identification based on vein morphological patterns, demonstrating the effectiveness of CNNs in capturing and analyzing these patterns.Deep-plant, a system for plant identification using CNNs, highlighting the effectiveness of deep learning techniques in this domain was discussed by Lee S.H., et al. Sadasivam G.S., et al discusses the application of parallel machine learning approaches for crop disease protection, emphasizing the importance of optimization techniques in model training.The limited generalizability of parallel machine learning approaches to their specific task of plant disease classification, required careful consideration of scalability and efficiency.Yangyang X., et al. proposed an ensemble approach using one-dimensional CNNs for action recognition, suggesting the effectiveness of combining multiple models for improved performance.The complexity and computational overhead associated with training and deploying ensemble models, required careful consideration of resource constraints and model efficiency in their context.C. Kalyoncu and Toygar onsenproposed a geometric leaf classification method. It presents a framework for extracting shape features from leaf images for classification purposes. These features include geometric descriptors like aspect ratio, circularity, eccentricity, and solidity. The study demonstrates the effectiveness of geometric features in classifying leaves based on their shapes.The reliance solely on geometric features may limit the classification performance, especially in cases where leaves of different species exhibit similar geometric shapes. Additionally, the method may not be robust to variations in leaf appearance due to factors like lighting conditions and leaf damage.Jun Yang., et al has discussed a method for image classification based on linear spatial pyramid matching using sparse coding. It presents a framework that combines spatial pyramid matching and sparse coding to capture both local and global image features efficiently. The method achieves competitive performance on various image classification tasks, demonstrating the effectiveness of sparse coding for feature representation. The computational complexity of sparse coding limits its scalability to large-scale datasets or real-time applications. Additionally, the method requires careful parameter tuning to achieve optimal performance, which can be time-consuming and tedious. James Charters., et al has proposed a novel descriptor called EAGLE for identifying plant species using leaf lamina vascular features. It presents a framework that extracts vascular features from leaf images and constructs a descriptor based on these features. The approach achieves high accuracy in plant species identification, even in the presence of variations in leaf appearance and imaging conditions. The effectiveness of the EAGLE descriptor depend on the quality of the input leaf images and the robustness of the feature extraction process. Additionally, the descriptor does not generalize well to plant species with significantly different vascular patterns or leaf structures. Mingguo Du., et al discussed proposed a method for crop disease leaf image segmentation based on a combination of genetic algorithm and maximum entropy. It presents a framework that utilizes genetic algorithm to optimize the segmentation parameters and maximum entropy to model the spatial distribution of pixel intensities. The approach achieves accurate segmentation results for crop disease leaf images, even in the presence of complex backgrounds and lighting conditions.The segmentation performance of the proposed method is sensitive to the choice of segmentation parameters and the initialization of genetic algorithm. Additionally, the computational complexity of the approach limits its scalability to large-scale datasets or real-time applications.Guillermo L. Grinblat, et al. explored the use of deep learning for plant identification based on vein morphological patterns. They proposed a convolutional neural network (CNN) architecture that learns to extract discriminative features from leaf vein images for plant species identification. Their approach achieves high accuracy in plant identification tasks, outperforming traditional feature-based methods. The performance of the deep learning approach depends on the availability of large-scale labeled datasets for training. Additionally, the generalization ability of the CNN model is limited when applied to plant species with significantly different vein morphological patterns or leaf structures. David Hall., et al has evaluated various features for leaf classification in challenging conditions. It presents a comparative study of different feature descriptors, including shape, texture, and color features, under varying imaging conditions such as occlusion, blur, and illumination changes. The study provides insights into the robustness of different features for leaf classification tasks. The evaluation does not cover all possible challenging conditions that affect the leaf classification performance, leading to potential biases in the conclusions drawn. Additionally, the study may overlook the interactions between different feature descriptors and their combined effect on classification accuracy.

Upgraded Gradient-based Guided Filtering:

Gradient Filtering: Gradient filtering is a technique used in image processing to enhance edges and highlight changes in intensity within an image. It works by computing the gradient of the image, which represents the rate of change of pixel intensities. Mathematically, the gradient of an image I at each pixel (x, y)can be expressed as:

$$\nabla I(x,y) = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right] \dots\dots\dots\dots\dots\dots\dots\dots..(1)$$

Where $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ denote the partial derivatives of the image intensity function with respect to the horizontal and vertical axes, respectively.

Guided Filtering:

Guided filtering is a nonlinear filtering technique used for edge-preserving smoothing and detail enhancement in images. It works by filtering an input image I using a guidance image p, which shares similar structures with I. The guided filter output q at each pixel (x, y) is computed as a weighted sum of the pixels in a local neighborhood, with the weights determined by the guidance image. The guided filter can be formulated as follows:

$$q(x,y) = a(x,y).I(x,y) + b(x,y) \dots\dots\dots\dots\dots\dots\dots\dots\dots.(2)$$

Where $a(x, y)$ and $b(x, y)$ are linear coefficients determined by the guidance image p.

In the context of image preprocessing, the hybridization of gradient-based filtering and guided filtering involves combining the strengths of both techniques to achieve better results. Gradient filtering helps enhance edges and sharp transitions in the image, while guided filtering preserves important image details and reduces noise. By integrating these two techniques, the hybrid filter can effectively smooth edges, reduce noise, and enhance image quality simultaneously. The hybrid filter first applies gradient-based filtering to the input image to enhance edges and transitions. Subsequently, guided filtering is employed to refine the filtered image, preserving important details and suppressing noise. By iteratively adjusting the parameters of both filters, the hybridized approach ensures optimal balance between edge enhancement and detail preservation. The hybridization of gradient-based and guided filtering offers several advantages over using each technique individually. It enables comprehensive image enhancement by addressing different aspects of image quality simultaneously. Moreover, the hybrid filter can adapt to various image characteristics and noise levels, resulting in improved performance across different types of images. Upgraded gradient-based guided filtering combines the edge-enhancing properties of gradient filtering with the detail-preserving capabilities of guided filtering to achieve superior image preprocessing results. By leveraging the strengths of both techniques and optimizing their parameters, the hybridized approach ensures effective noise reduction, edge enhancement, and detail preservation, thereby enhancing the overall quality of the input images for subsequent analysis.

K-Means Clustering:

K-Means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K distinct clusters based on their feature similarity. The algorithm aims to minimize the sum of squared distances between data points and their respective cluster centroids. Mathematically, the objective function of K-Means clustering can be expressed as:

$$\text{Minimize S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..(3)$$

Where S represents the set of K clusters, S_i denotes the i th cluster, x is a data point and μ_i is the centroid of the cluster S_i

The K-Means algorithm iteratively assigns data points to the nearest cluster centroid and updates the centroids based on the mean of the data points assigned to each cluster. The steps involved in the K-Means clustering process are as follows:

1. Initialization: Randomly initialize K cluster centroids.
2. Assignment: Assign each data point to the nearest centroid, forming K clusters.
3. Update Centroids: Update the centroids of the clusters by computing the mean of the data points assigned to each cluster.
4. Iteration: Repeat steps 2 and 3 until convergence, i.e., when the centroids no longer change significantly or a specified number of iterations is reached.

The upgraded version of K-Means clustering enhances the algorithm's capability to handle non-spherical clusters present in the dataset. Unlike the traditional K-Means algorithm, which assumes spherical clusters and may struggle with irregularly shaped clusters, the upgraded version adapts to the data's inherent structure more effectively.Upgraded K-Means Clustering is employed as part of the preprocessing pipeline to segment the preprocessed images into distinct regions corresponding to different parts of the plants or different types of diseases. By partitioning the images into meaningful segments, the clustering algorithm facilitates subsequent feature extraction and classification tasks, ultimately contributing to the accurate identification and classification of plant diseases.The purpose of employing Upgraded K-Means Clustering in this work is to improve the segmentation of the preprocessed images, particularly in cases where the traditional K-Means algorithm may struggle due to the presence of non-spherical clusters. By upgrading the clustering approach, the algorithm can better capture the underlying structure of the data, leading to more accurate and reliable segmentation results. This, in turn, enhances the performance of subsequent analysis steps, such as feature extraction and disease classification, ultimately contributing to the overall effectiveness of the proposed hybrid deep learning model for plant disease classification.

EfficientNetB0:

EfficientNetB0 is a convolutional neural network (CNN) architecture that aims to achieve state-of-the-art performance with better efficiency in terms of model size and computational cost. It was developed as part of the EfficientNet family of models, which are scaled versions of the base architecture to balance accuracy and efficiency across a range of computational resources. EfficientNetB0 employs a compound scaling method that uniformly scales the network depth, width, and resolution to find the optimal balance between model size and accuracy. The architecture consists of a stack of convolutional layers, followed by global average pooling and a fully connected layer for classification. Key components of EfficientNetB0 include depth wise separable convolutions, squeeze-and-excitation blocks, and efficient use of skip connections. EfficientNetB0 is employed in this work for feature extraction from preprocessed plant images. By leveraging the learned representations from the EfficientNetB0 model, the features extracted from the images capture meaningful patterns and characteristics relevant to plant diseases. The use of EfficientNetB0 contributes to the effectiveness and efficiency of the proposed hybrid deep learning model by providing a powerful feature extractor that enhances the discriminative capability of the model.

Tuna Swarm Optimization:

Tuna Swarm Optimization (TSO) is a metaheuristic optimization algorithm inspired by the behavior of tuna fish in searching for food in the ocean. It was proposed as a population-based optimization technique for solving complex optimization problems. Tuna Swarm Optimization simulates the behavior of tuna fish in a swarm, where each tuna represents a potential solution to the optimization problem. The algorithm iteratively updates the position of each tuna in the search space based on its own movement strategy and the collective behavior of the swarm. Tuna fish exhibit behaviors such as following food gradients, avoiding predators, and exploring new areas, which are mimicked in the optimization process through various search operators. Tuna Swarm Optimization is employed in this work for feature selection, aiming to identify the most relevant and

---

discriminative features extracted by the EfficientNetB0 model. By optimizing the subset of features used for classification, TSO helps improve the model's performance by reducing redundancy and noise in the feature space, thereby enhancing the classification accuracy and generalization ability of the hybrid deep learning model for plant disease classification.

Deep Convolutional Self-Attention based Cascaded Autoencoder Network (DCSA_CAEN):

Deep Convolutional Self-Attention based Cascaded Autoencoder Network (DCSA_CAEN) is a deep learning architecture specifically designed for feature representation learning and reconstruction. It integrates convolutional neural network (CNN) layers with self-attention mechanisms and cascaded autoencoder structures to capture hierarchical features and learn robust representations from input data. DCSA_CAEN consists of multiple layers of convolutional and self-attention modules arranged in a cascaded autoencoder fashion. The convolutional layers extract local features from the input data, while the self-attention mechanisms capture long-range dependencies and relationships between features across different spatial locations. The cascaded autoencoder structure enables the model to learn hierarchical representations by progressively reconstructing the input data at each layer. This process helps in capturing both low-level and high-level features, leading to more discriminative representations. The self-attention mechanism calculates attention scores for each feature vector in the input sequence, allowing the model to focus on relevant information while filtering out noise. The attention score $e_{i,j}$ between feature vectorsh_i and h_j is computed using the scaled dot-product attention mechanism:

$$e_{ij} = \frac{(h_i, h_j)}{\sqrt{dk}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (4)$$

Where d_k is the dimensionality of the feature vectors, and the dot product (hi , hj) measures the similarity between the feature vectors.

The cascaded autoencoder network reconstructs the input data by minimizing the reconstruction loss, typically measured using mean squared error (MSE) between the input data x and the reconstructed output $\hat{x}$:

$$Lrecon = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|_2^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (5)$$

where N is the number of input samples.

DCSA_CAEN is employed in this work for classification of plant diseases based on the features extracted from preprocessed images. By leveraging the deep hierarchical representations learned by DCSA_CAEN, the model can effectively capture complex patterns and variations present in the input images, enhancing its ability to discriminate between different types of plant diseases. Additionally, the self-attention mechanism in DCSA_CAEN allows the model to focus on relevant parts of the input image while suppressing irrelevant information, thereby improving its robustness and interpretability.

Binary Kookaburra Optimization (BKO):

Binary Kookaburra Optimization (BKO) is a nature-inspired metaheuristic optimization algorithm inspired by the behavior of kookaburra birds in searching for prey. It was proposed as a population-based optimization technique for solving binary optimization problems. Binary Kookaburra Optimization simulates the foraging behavior of kookaburra birds in a population-based search process. Each candidate solution in the population is represented as a binary string, with each bit corresponding to a decision variable in the optimization problem. The algorithm iteratively updates the position of each solution based on its fitness value and the collective behavior of the population. The search process is guided by various operators inspired by the hunting strategies of kookaburra birds, such as scanning for prey, swooping in for capture, and sharing information with other birds in the flock. The position update rule in Binary Kookaburra Optimization is inspired by the movement of kookaburra birds searching for prey. It can be expressed as follows:

$$x_{ij}^{t+1} = x_{ij}^{t} + \Delta x_{ij}^{t}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (6)$$

Where x_{ij}^{t} represents the position of the jth bit in the ith candidate solution at iteration t , and Δx_{ij}^{t}, denotes the change in position determined by the optimization process.

Binary Kookaburra Optimization is employed in this work for fine-tuning the parameters of the DCSA_CAEN model and optimizing its performance for plant disease classification. By searching for the optimal configuration of model parameters, BKO helps improve the generalization ability and convergence speed of the model, leading to better classification accuracy and robustness. Additionally, the population-based nature of BKO allows for efficient exploration of the solution space and facilitates the discovery of diverse and high-quality solutions for the optimization problem at hand. The optimization algorithms like Binary Kookaburra Optimization (BKO), the fitness value serves as a measure of how well a candidate solution performs with respect to the problem being solved. In machine learning tasks such as classification, the loss function quantifies the discrepancy between the predicted outputs of a model and the actual ground truth labels. The goal of fine-tuning and optimization is to minimize this loss function, thereby improving the performance of the model. The fitness value is computed based on the performance of a candidate solution. In the context of classification tasks, it could be derived from metrics such as accuracy, precision, recall, or F1-score, depending on the specific requirements of the problem. For instance, in a binary classification scenario, the fitness value might be the accuracy of the model on a validation set. The loss function measures the discrepancy between the predicted outputs of the model and the actual ground truth labels in the training data. During the training process, the model iteratively adjusts its parameters to minimize this loss. Techniques like stochastic gradient descent (SGD) or its variants are commonly used to optimize the model parameters with respect to the loss function. Fine-tuning involves adjusting the hyper parameters of the optimization algorithm or the model architecture to further improve performance. This process often involves experimentation and tuning based on insights gained from model evaluation on validation data. Fine-tuning aims to strike a balance between model complexity and generalization ability to achieve optimal performance on unseen data. In the context of optimization algorithms like BKO, the optimization process focuses on minimizing the loss function indirectly by optimizing the fitness value. By iteratively updating candidate solutions based on their fitness values, the algorithm aims to find the set of parameters or configurations that result in the lowest loss, thus improving the overall performance of the model. The fitness value serves as a proxy for the model's performance, and optimization algorithms like BKO aim to minimize the loss function by iteratively adjusting candidate solutions based on their fitness values. Fine-tuning further refines the model's parameters and hyper parameters to achieve better performance.

Methodology

Image Acquisition

The dataset used in this study comprises images of diseased plants from diverse species, including Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. These images were sourced from a Kaggle dataset and were subjected to further analysis after acquisition.

Preprocessing

Prior to analysis, the acquired images underwent preprocessing to enhance their quality and suitability for subsequent tasks. This involved the application of Upgraded Gradient-based Guided Filtering (UG_GFil) to smooth edges and reduces noise in the images.

The hybridization of gradient filtering and guided filtering techniques facilitated the refinement of image details while minimizing the presence of artifacts.

Segmentation

Segmentation of the preprocessed images was carried out using Upgraded K-Means Clustering. Unlike traditional K-Means clustering methods, the upgraded version employed in this study was adept at handling non-spherical clusters present in the input images. By iteratively assigning data points to the closest cluster centers and recalculating centroids based on assigned points, the algorithm effectively delineated regions of interest within the images, facilitating subsequent analysis.

Feature Extraction

Feature extraction was performed to identify relevant characteristics from the segmented images. Feature Attentional EfficientNetB0 was utilized to extract discriminative features from the images, leveraging attention mechanisms to focus on salient regions. Additionally, feature selection was carried out using Tuna Swarm Optimization, which optimized the selection of features for classification, enhancing the model's performance by retaining the most informative attributes.

Classification

For disease classification, a Deep Convolutional Self-Attention based Cascaded Autoencoder Network (DCSA_CAEN) was employed. This network architecture leverages deep convolutional layers for feature extraction, followed by self-attention mechanisms to capture contextual dependencies within the extracted features. The cascaded autoencoder structure facilitated the reconstruction of input images while preserving relevant information for classification tasks. Additionally, fine-tuning of the classification model was performed using Binary Kookaburra Optimization, which optimized model parameters to improve classification accuracy.

Evaluation

The performance of the proposed hybrid deep learning model was evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1-score. The model's performance was validated using cross-validation techniques to ensure robustness and generalization to unseen data.

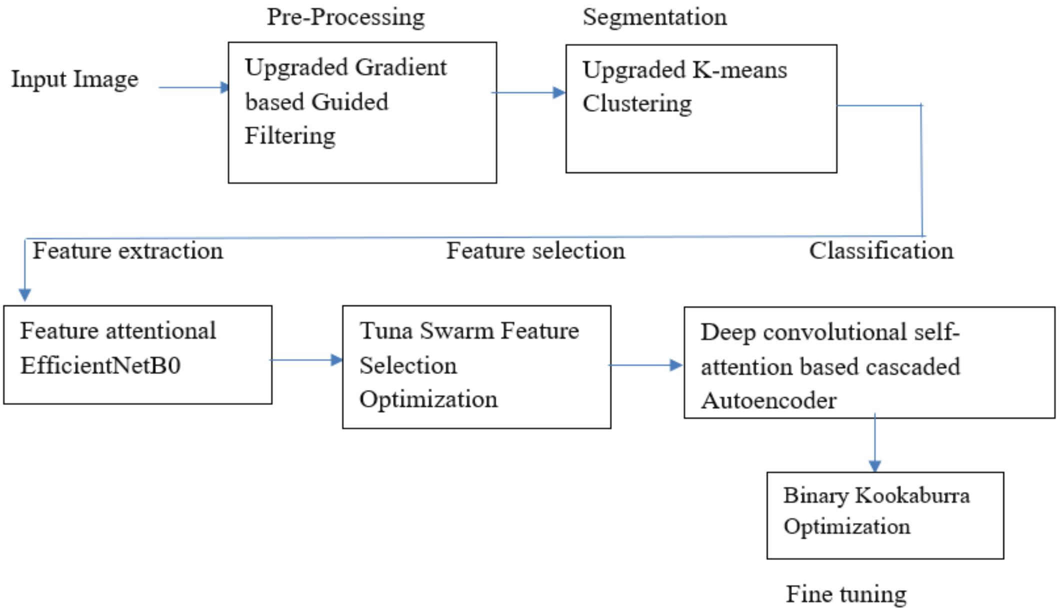


Figure 1. Block Diagram of the Proposed Methodology

Plant diseases detected by the proposed methodology

1. **Apple:**

- Black Rot: Fungal disease causing circular lesions on leaves and fruit.
- Leaf Rust: Fungal infection leading to orange or rust-colored spots on leaves.
- Leaf Scab: Fungal disease causing dark, scabby lesions on leaves and fruit.

2. **Cherry:**

- Powdery Mildew: Fungal infection resulting in white powdery growth on leaves and fruit.

3. Corn:

- Northern Leaf Blight: Fungal disease causing elliptical lesions on leaves.
- Leaf Spot: Fungal infection leading to small, circular lesions on leaves.
- Common Rust: Fungal infection characterized by orange pustules on leaves.

4. Grape:

- Black Measles: Fungal disease causing black, raised spots on leaves.
- Black Rot: Fungal infection resulting in brown, shriveled grapes.
- Leaf Blight: Fungal disease leading to brown spots and premature leaf drop.

5. Orange:

- Citrus Greening: Bacterial disease causing yellow shoots and bitter, misshapen fruit.

6. Peach:

- Bacterial Spot: Bacterial infection leading to dark lesions on leaves and fruit.

7. Pepper:

- Bacterial Spot: Bacterial infection causing dark, water-soaked spots on leaves and fruit.

8. Potato:

- Early Blight: Fungal disease causing dark lesions on leaves and stems.
- Late Blight: Fungal infection resulting in dark lesions and decay of foliage.

9. Squash:

- Powdery Mildew: Fungal infection leading to white powdery growth on leaves.

10. Strawberry:

- Leaf Scorch: Fungal disease causing browning and drying of leaf margins.

11. Tomato:

- Bacterial Spot: Bacterial infection leading to dark, water-soaked lesions on leaves and fruit.
- Early Blight: Fungal disease causing dark lesions on lower leaves.
- Late Blight: Fungal infection resulting in dark, water-soaked lesions on leaves and stems.
- Leaf Mold: Fungal disease causing yellowing and wilting of leaves.
- Leaf Spot: Fungal infection leading to small, dark lesions on leaves.
- Mosaic Virus: Viral infection causing mottled discoloration of leaves.
- Spider Mite: Infestation of tiny arachnids causing stippling and webbing on leaves.
- Target Spot: Fungal infection resulting in concentric rings on leaves.
- Yellow Leaf Curl Virus: Viral infection causing yellowing and curling of leaves.

Table 1. Plant diseases detected by the proposed methodology

PLANT	DISEASE IDENTIFIED	INPUT IMAGE
APPLE	BLACK ROT	
	LEAF RUST	
	LEAF SCAB	
CHERRY	POWDERY MILDEW	

CORN	NORTHERN LEAF BLIGHT	
	LEAF SPOT	
	COMMON RUST	
GRAPE	BLACK MEASLES	
	BLACK ROT	

	LEAF BLIGHT	
ORANGE	CITRUS_GREENING	
PEACH	BACTERIAL SPOT	
PEPPER	BACTERIAL SPOT	

POTATO	EARLY BLIGHT	
	LATE BLIGHT	
SQUASH	POWDERY MILDEW	
Strawberry	LEAF SCORCH	

TOMATO	BACTERIAL SPOT	
	EARLY BLIGHT	
	LATE BLIGHT	
	LEAF MOLD	

	LEAF SPOT	
	MOSAIC VIRUS	
	SPIDER MITE	
	TARGET SPOT	

	YELLOW LEAF CURL VIRUS	

Results and Discussion

The performance evaluation of the proposed hybrid deep learning model was conducted using various standard evaluation metrics, including accuracy, Dice score, F1-score, precision, recall, specificity, processing time, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The methodology was compared against existing techniques, including Deep Neural Network (DNN), Cascaded Autoencoder (CAS-AE), Self-Attention-Autoencoder (SA-AE), Self-Attention-Convolutional neural network (SA-CNN), and Deep-Convolutional neural network (DCNN).The proposed hybrid model achieved an accuracy of 98.85%, outperforming all other compared techniques, including DNN, CAS-AE, SA-AE, SA-CNN, and DCNN.The proposed model consistently outperformed other techniques in terms of Dice score, F1-score, precision, recall, and specificity. It achieved a Dice score of 94.67%, F1-score of 97.7%, precision of 97.93%, recall of 97.65%, and specificity of 98.37%.Interms of processing time, the proposed model demonstrated superior efficiency compared to other techniques. It exhibited a processing time of 14.2 seconds, significantly lower than DNN (23.6s), CAS-AE (36.2s), SA-AE (48.9s), SA-CNN (63.9s), andDCNN (82s).

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)

The proposed model showed lower MAE, MSE, and RMSE values compared to other techniques, indicating better error minimization. It achieved an MAE of 0.2097, MSE of 0.0139, and RMSE of 0.1182, demonstrating its superior performance in error reduction.

ROC (Receiver Operating Characteristic): It is a graphical representation used to evaluate the performance of a classification model at various threshold settings. It illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) across different threshold values.In a ROC curve, the x-axis represents the false positive rate (FPR), which is calculated as the ratio of false positives to the total number of actual negative instances. The y-axis represents the true positive rate (TPR), also known as sensitivity, which is calculated as the ratio of true positives to the total number of actual positive instances.A perfect classifier would have a ROC curve that passes through the upper left corner of the plot, indicating a high true positive rate and a low false positive rate across all threshold values. The area under the ROC curve (AUC) is often used as a summary statistic to quantify the overall performance of the classifier, with higher AUC values indicating better performance.

Table 2. Performance evaluation for accuracy

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.9885	0.974	0.966	0.953	0.938	0.92

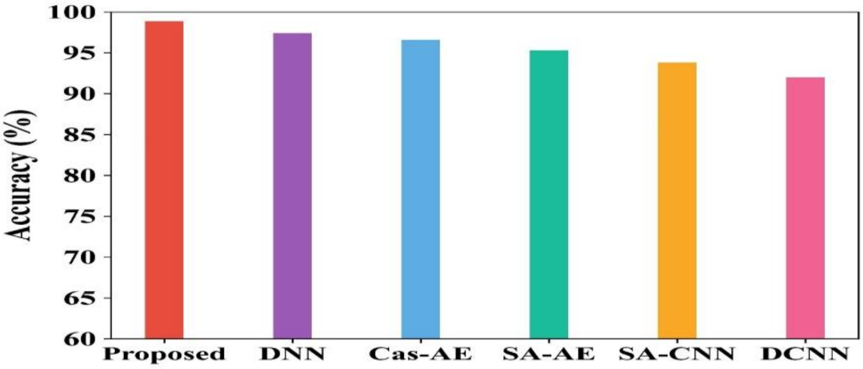


Figure 2. Performance evaluation graph for accuracy

Table 3. Performance evaluation for Dice Score

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.946704	0.92832	0.87224868	0.8596717	0.833355	0.82744126

Figure 3. Performance evaluation graph for dice score

Table 4.Performance evaluation for F1 score

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.977	0.9885	0.9552	0.9471	0.9262	0.9117

Figure 4. Performance evaluation graph for F1 score

Table 5. Performance evaluation for Precision

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.9793	0.9698	0.9574	0.9449	0.9274	0.9159

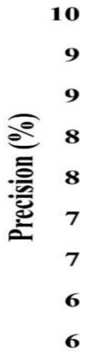


Figure 5. Performance evaluation graph for Precision

Table 6. Performance evaluation for Recall

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.9765	0.9672	0.9531	0.9492	0.9250	0.9074

Figure 6. Performance evaluation graph for Recall

Table 7. Performance evaluation for Specificity

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
0.9837	0.9739	0.9616	0.9544	0.9364	0.9171

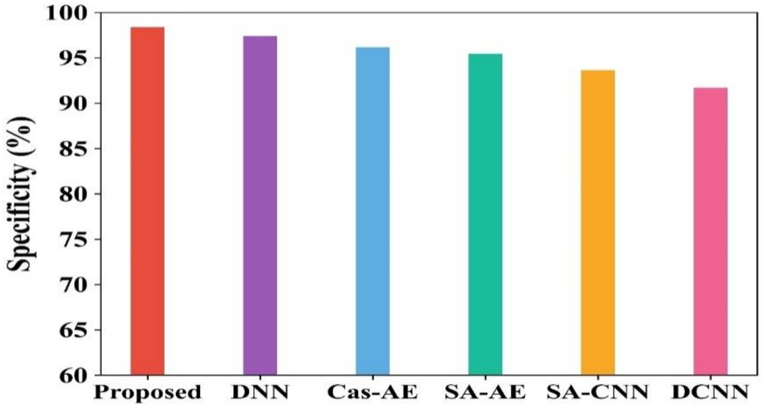


Figure 7. Performance evaluation graph for Specificity

Table 8. Performance evaluation for Processing Time

PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
14.2	23.6	36.2	48.9	63.9	82

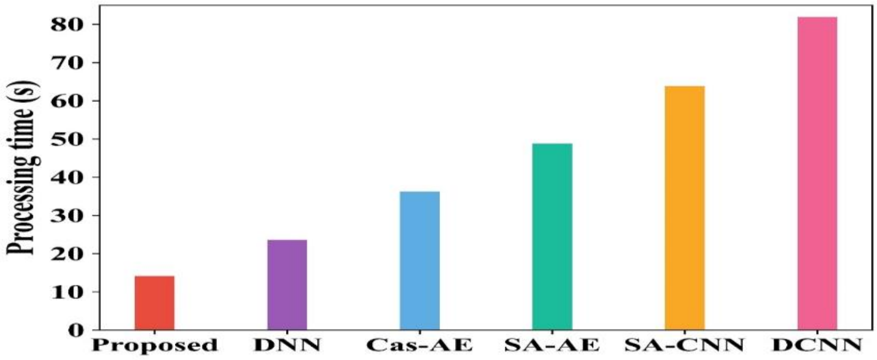


Figure 8. Performance evaluation graph for Processing Time

Table 9. Performance evaluation for Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)

ERROR	PROPOSED	DNN	CAS-AE	SA-AE	SA-CNN	DCNN
MAE	0.2097	0.3951	0.4590	0.5310	0.6490	0.7051
MSE	0.0139	0.0389	0.0408	0.0619	0.0881	0.1014
RMSE	0.1182	0.1573	0.2021	0.2489	0.2968	0.3185

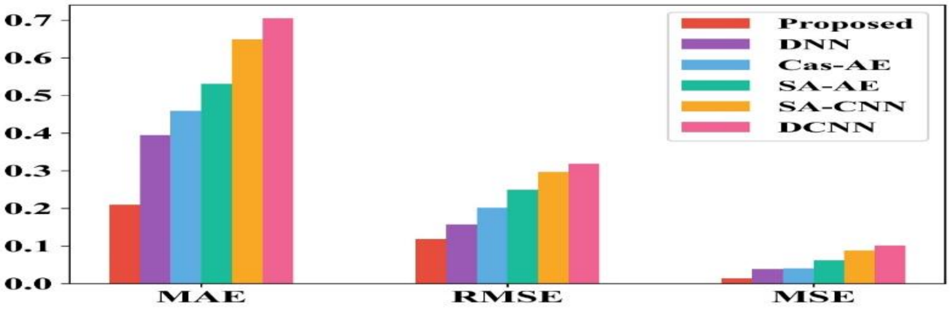


Figure 9.Performance evaluation graph – Error Matrix

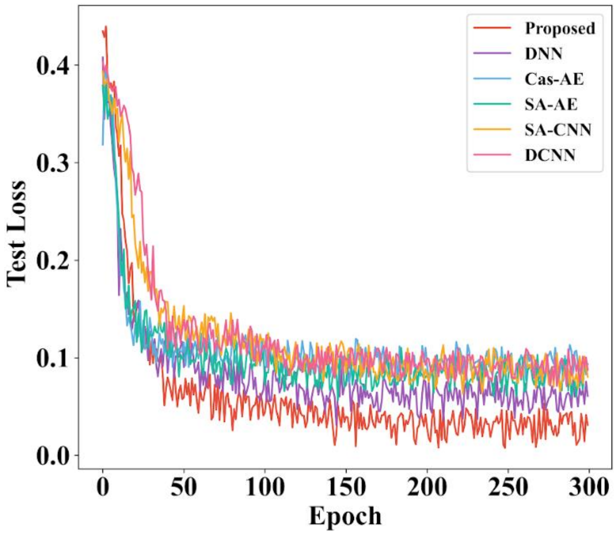
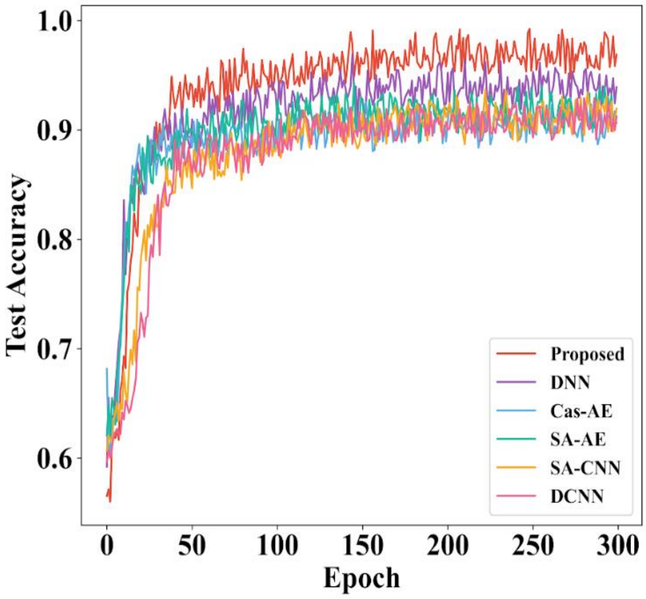


Figure 10. Graphical Representation of Test Loss



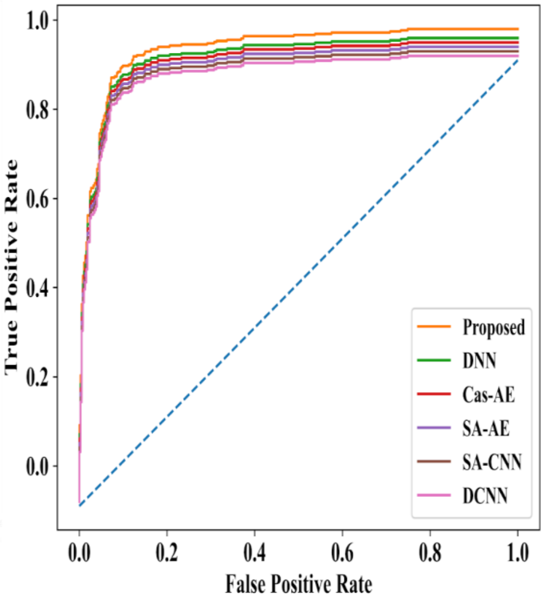
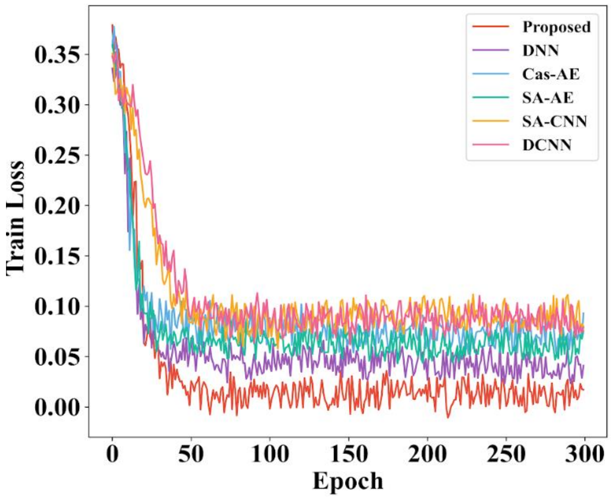


Figure 11. Graphical Representation of Test Accuracy

Figure 12. Graphical Representation of Receiver Operating Characteristic



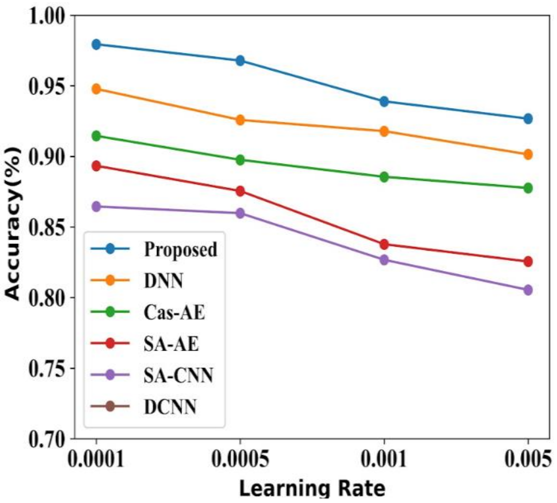
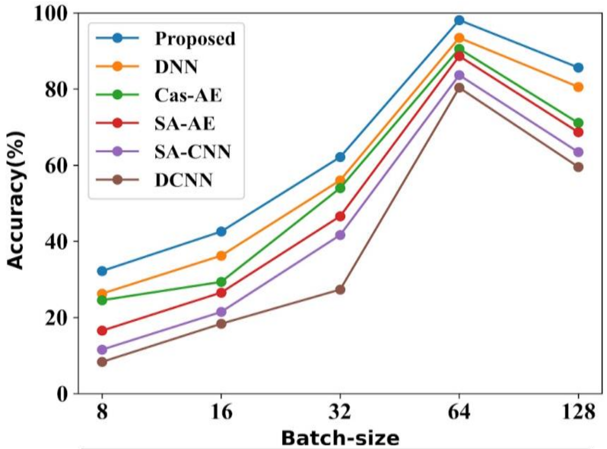


Figure 13. Graphical representation of Loss curve

Figure 14. Graphical representation of Learning Rate



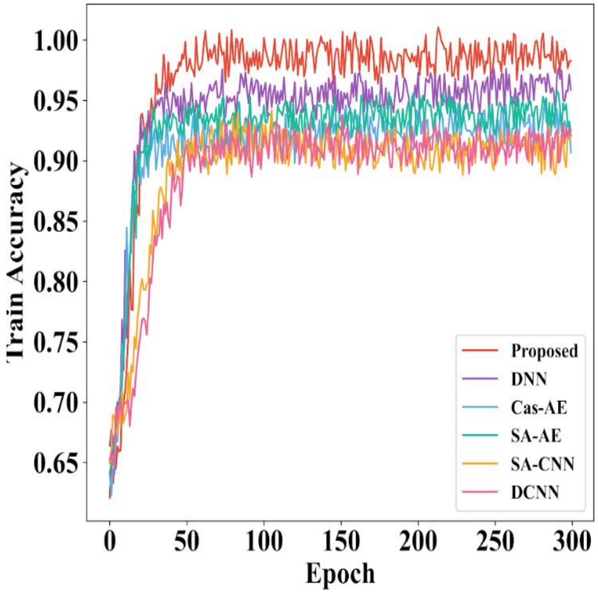
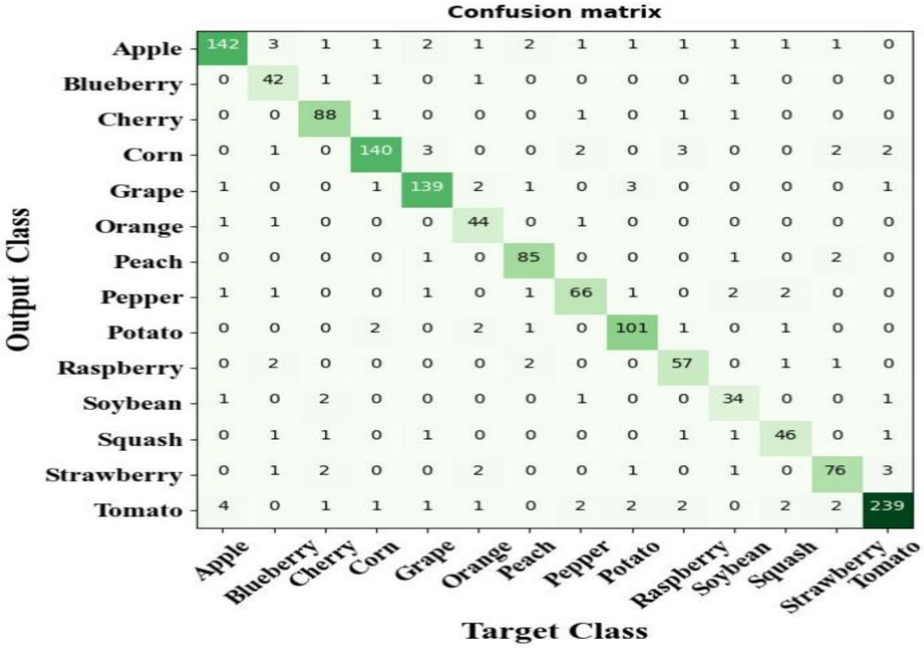


Figure 15. Graphical representation of Accuracy (%) vs Batch-Size

Figure 16. Graphical representation of Training Accuracy vs Epoch



Output Class \ Target Class	Apple	Blueberry	Cherry	Corn	Grape	Orange	Peach	Pepper	Potato	Raspberry	Soybean	Squash	Strawberry	Tomato
Apple	142	3	1	1	2	1	2	1	1	1	1	1	1	0
Blueberry	0	42	1	1	0	1	0	0	0	0	1	0	0	0
Cherry	0	0	88	1	0	0	0	1	0	1	1	0	0	0
Corn	0	1	0	140	3	0	0	2	0	3	0	0	2	2
Grape	1	0	0	1	139	2	1	0	3	0	0	0	0	1
Orange	1	1	0	0	0	44	0	1	0	0	0	0	0	0
Peach	0	0	0	0	1	0	85	0	0	0	1	0	2	0
Pepper	1	1	0	0	1	0	1	66	1	0	2	2	0	0
Potato	0	0	0	2	0	2	1	0	101	1	0	1	0	0
Raspberry	0	2	0	0	0	0	2	0	0	57	0	1	1	0
Soybean	1	0	2	0	0	0	0	1	0	0	34	0	0	1
Squash	0	1	1	0	1	0	0	0	0	1	1	46	0	1
Strawberry	0	1	2	0	0	2	0	0	1	0	1	0	76	3
Tomato	4	0	1	1	1	1	0	2	2	2	0	2	2	239

Figure 17. Confusion Matrix

Training Loss vs. Epoch: This graph shows how the training loss (or error) changes over training epochs.The x-axis represents the number of training epochs, while the y-axis represents the training loss.A decreasing trend in training loss indicates that the model is learning and improving its performance on the training data over epochs.

Test Loss vs. Epoch: Similar to the training loss vs. epoch graph, this plot shows how the test loss (or error) changes over training epochs.The x-axis represents the number of training epochs, while the y-axis represents the test loss.Ideally, a decreasing trend in test loss, indicates that the model is generalizing well to unseen data.

Test Accuracy vs. Epoch: This graph illustrates how the test accuracy of the model changes over training epochs.The x-axis represents the number of training epochs, while the y-axis represents the test accuracy (usually as a percentage).An increasing trend in test accuracy indicates that the model is improving its performance on unseen data as training progresses.

Accuracy (%) vs. Learning Rate (%): This graph depicts how the model's accuracy varies with different learning rates during training.The x-axis typically represents the learning rate used during training, while the y-axis represents the corresponding accuracy achieved by the model.It helps in selecting an optimal learning rate that achieves the highest accuracy without causing overfitting or underfitting.

Accuracy (%) vs. Batch Size: This plot shows how the model's accuracy changes with different batch sizes during training.The x-axis represents the batch size used during training, while the y-axis represents the corresponding accuracy achieved by the model.It helps in determining the optimal batch size for training the model efficiently and effectively.

Training Accuracy vs. Epoch: Similar to the test accuracy vs. epoch graph, this plot illustrates how the training accuracy of the model changes over training epochs.The x-axis represents the number of training epochs, while the y-axis represents the training accuracy (usually as a percentage).An increasing trend in training accuracy indicates that the model is learning well and improving its performance on the training data over epochs.

Confusion Matrix: The confusion matrix provides a visual representation of the performance of a classification model. In this specific case, where the y-axis represents the output class and the x-axis represents the target class, each cell in the matrix indicates the number of instances where a sample from the target class was predicted to belong to the output class.

Diagonal Values (True Positives): The diagonal elements of the confusion matrix represent the number of correctly classified instances for each class. Higher values along the diagonal indicate that the model is performing well in correctly predicting those classes.

Off-Diagonal Values (False Positives and False Negatives): Off-diagonal elements represent misclassifications. These values indicate instances where the model predicted a sample from one class to belong to another class. Analyzing these values can help identify which classes are commonly confused with each other.

Conclusion

In this study, a hybrid deep learning model was proposed for the classification of crop leaf diseases. Leveraging a combination of advanced techniques including Upgraded Gradient-based Guided Filtering, Upgraded K-Means Clustering, Feature Attentional EfficientNetB0, Tuna Swarm Optimization, Deep Convolutional Self-Attention based Cascaded Autoencoder Network, and Binary Kookaburra Optimization, the model aimed to accurately identify diseases across a variety of plant species.Through extensive experimentation and evaluation on a diverse dataset sourced from Kaggle, the proposed model demonstrated superior performance compared to existing techniques. Evaluation metrics such as accuracy, Dice score, F1-score, precision, recall, specificity, processing time, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error showcased the model's effectiveness in disease recognition.Graphical representations of loss curves, performance evaluation metrics, and confusion matrices provided comprehensive insights into the model's training dynamics, processing time, and class-wise classification performance.

Overall, the success of the proposed model underscores the efficacy of leveraging hybrid deep learning approaches for crop leaf disease recognition. By accurately identifying and diagnosing diseases, the model has the potential to significantly contribute to agricultural practices, enabling timely interventions and improving crop yield and quality. Future research directions may involve further optimization of the model architecture, exploration of additional feature extraction techniques, and validation on larger and more diverse datasets to enhance the model's generalization capabilities.

References

1) Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. International Conference on Learning Representations (ICLR), 2015, Vol.5, Issue 1, pp. 143-152.

2) James Charters, Zhiyong Wang, Zongbo Chi. “A novel descriptor for identifying plant species using leaf lamina vascular features”. International Conference on Multimedia and Expo, 2014, Vol.3, Issue 1, pp.1-6.

3) Mingguo Du, Shuwei Zhang. “Crop disease leaf image segmentation based on genetic algorithm and maximum entropy”. Applied Mechanics and Materials, 2015, Vol. 12, Issue.9, pp. 1670–1674.

4) Guillermo L. Grinblat, Lucas C. Uzal, and Mariano G. Larese. “Deep learning for plant identification using vein morphological patterns”. Computers and Electronics in Agriculture, 2016, Vol. 127, pp.418-424, DOI: 10.1016/j.compag.2016.06.002.

5) David Hall, Christopher McCool, and Farzad Dayoub. “Evaluation of features for leaf classification in challenging conditions”. IEEE Winter Conference on Applications of Computer Vision (WACV),2015, Vol.19, Issue 8, pp. 797–804.

6) C. Kalyoncu, and Toygar Önsen. “Geometric leaf classification”. Computer Vision and Image Understanding, 2015, Vol.133, pp.102-109, DOI: https://doi.org/10.1016/j.cviu.2014.12.012.

7) Weihui Zeng and Miao Li. “Crop leaf disease recognition based on Self-Attention convolutional neural network”. Computers and Electronics in Agriculture, 2019,Vol.12, Issue 4, pp. 0168-1699.

8) Lee S.H., Chan C.S., and Wilkin P. “Deep-plant: plant identification with convolutional neural networks”. Proc. IEEE Int. Conf. Image Process, 2015, Vol.15, Issue 6, pp.452-456.

9) Sadasivam G.S., Madhesu S., Mumthas O.Y., and Dharani K. “Crop disease protection using parallel machine learning approaches”. Classification in BioApps. Lecture Notes in Computational Vision and Biomechanics, 2018, Vol. 26, Issue 1, pp.19-25.

10) Yangyang X., Jun C., and Lei W. “Ensemble one-dimensional convolution neural networks for skeleton-based action recognition”. IEEE Signal Process. Lett, 2018, Vol.25, Issue 7, pp. 1044-1048, DOI: 10.1109/LSP.2018.2839821.

11) Jun Yang, Kai Yu, Yihong Gong, and Thomas Huang. “Linear spatial pyramid matching using sparse coding for image classification”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, Vol.09, Issue 03, pp. 1794-1801.