AVE Trends in Intelligent Health Letters

An Innovative ETLBO-Based HTConvNet Classification Approach for Efficient Apple Leaf Disease Detection

Swathi Baswaraju[1,*], K. A. Jayasheel Kumar[2], Thirumalraj Karthikeyan[3], S. Venkatasubramanian[4], Sheila Agnes Vidot[5]

[1]Department of Computer Science and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[2]Department of Automobile Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[3]Department of Artificial Intelligence, Trichy Research Labs, Quest Technologies, Tiruchirappalli, Tamil Nadu, India.
[4]Department of Computer Science and Business Systems, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India.
[5]Department of Medical Consultant, IPS Health, Mahé, Victoria, Seychelles.
baswarajuswathi@gmail.com[1], jayasheel.81088@gmail.com[2], thirumalraj.k@gmail.com[3], veeyes@saranathan.ac.in[4] vidotsheila@gmail.com[5]

*Corresponding author

Abstract: Automatic plant disease detection is essential because it minimizes the laborious task of keeping an eye on large farms and identifies diseases early on, when they can still be prevented from causing further damage to plants. This scenario has a significant impact on a nation's economy and harms plant health by reducing production. The suggested model used Deep Learning (DL) to achieve the best possible classification accuracy for leaf diseases. For additional processing, the suggested model used the Plant-Microbe Biology and Plant Pathology sections of the Cornell University Dataset. As a crucial component of image preprocessing, denoising can significantly improve image quality, benefiting subsequent operations such as feature extraction and image segmentation. This study used the Non-Local Mean (NLMM) filtering technique for image denoising. Following preprocessing, Convolutional neural networks, encoder-decoders, and Swin Transformers were the three branches into which the extraction process was integrated, creating the SwinTransConv-ED model. Following the extraction procedure, a novel leaf image classification framework, Hybrid-Transformer and CNN (HTConvNet), built on the Transformer architecture, is proposed. In this study, the hyperparameter tuning procedure is used to attain the highest level of accuracy. An improved teaching-learning-based optimisation (ETLBO) to tune the HTConvNet classifier's hyperparameters.

Keywords: Deep Learning; Image Denoising; Swin Transformer; Leaf Image Classification; Teaching and Learning; Non-Local Moment Mean; Hybrid-Transformer and CNN; Swintransconv-ED.

Cite as: S. Baswaraju, K. A. J. Kumar, T. Karthikeyan, S. Venkatasubramanian, and S. A. Vidot, "An Innovative ETLBO-Based HTConvNet Classification Approach for Efficient Apple Leaf Disease Detection," *AVE Trends in Intelligent Health Letters*, vol. 2, no. 3, pp. 163–177, 2025.

Journal Homepage: https://avepubs.com/user/journals/details/ATIHL

Received on: 30/10/2024, **Revised on:** 16/02/2025, **Accepted on:** 04/05/2025, **Published on:** 07/09/2025

DOI: https://doi.org/10.64091/ATIHL.2025.000174

1. Introduction

Plant diseases are disorders that affect a plant's ability to grow and thrive, leading to abnormalities in its structure, vitality, or function [1]. A range of infections, including viruses, bacteria, nematodes, and other microorganisms, can cause these illnesses. Furthermore, environmental stresses weaken plants and make them more susceptible to disease. These stressors include things like poor soil, high moisture content, high temperatures, and insufficient nutrition [2]; [3]. Plant diseases can significantly impact the environment and the economy. They could disrupt ecosystems, reduce crop yields, and affect food availability [4]. Farmers and horticulturists regularly employ a range of strategies to avoid, control, or manage these illnesses. These methods include using fungicides or pesticides, applying resistant plant varieties, implementing cultural practices such as crop rotation and good sanitation, and using biological control methods [5]. Object recognition, image and video classification, and other tasks are among the many areas where CNNs excel [6]. They are modelled after the human brain's cerebral cortex and use a specialised architecture to find patterns in images. Their ability to autonomously extract features from raw data makes them a vital tool in modern deep learning [7]; [8].

Plant disease detection could be revolutionised using Convolutional Neural Networks (CNNs), a technique that has shown enormous promise. Deep learning algorithms of the CNN type are especially skilled at deciphering visual input, such as images. They are therefore a suitable fit for identifying plant diseases from their outward manifestations [9]. To train the CNN model to distinguish between disease-related patterns and symptoms, it is necessary to use a collection of images of both sick and healthy plants [10]. First, a collection of high-quality photos of plants in various disease states, along with photos of healthy plants, is compiled [11]. Once the images have been labelled, the CNN learns from them, enabling it to recognise patterns and traits indicative of specific diseases [12]; [13]. After being trained, the CNN can accurately recognise new plant photos, enabling farmers and agricultural experts to promptly detect and diagnose diseases at an early stage, thereby minimising crop losses. Because CNNs are so effective at detecting plant diseases, there is significant potential to ensure food security and streamline crop management [14]. This paper summarises several significant contributions that include the following essential elements:

- During the preprocessing phase, the image linked to leaf disease is preprocessed using the NLMM filtering technique.
- The extraction process creates a comprehensive picture of leaf images by blending and fusing these different multiscale features. To optimise the integration process, a weight-fusion method is devised that effectively merges multiscale characteristics.
- To attain 2D CNN classification for pixel images, a shallow CNN architecture is updated. To apply 1D pixel leaf images to most 2D CNNs, they are converted into three-dimensional image feature matrices, thereby improving 2D CNNs' generalizability for 1D pixel leaf image classification.
- A proposed framework combines CNN and Transformer to classify leaf images at the pixel level. The transformer receives sequence data as input and converts it into three feature stages, extracted using a CNN. Effective features can be propagated through higher layers via skip connections, which facilitate feature fusion between nearby encoders.
- The ETLBO is proposed to improve optimal ability. The TLBO is improved by using adaptive weights and the Kent chaotic map. These two techniques can help teachers and students in TLBO become more adept at searching.

Performance metrics such as precision (PR), F-measure (F1), accuracy (ACC), and recall (RC) are used to evaluate results [27]; [28].

2. Related Works

An artificial intelligence-based bibliometric evaluation of apple leaf disease detection was conducted by Bonkra et al. [15]. The scientific study's objective was to determine trends in the diagnosis of apple diseases by analysing factors such as new developments, patterns of ownership and cooperation, bibliographic coupling, productivity, and publication and citation patterns. There have been many exploratory, conceptual, and empirical investigations into identifying apple illnesses, but there hasn't been much effort to compile a comprehensive science map of transdisciplinary research. Considering the growing body of work on this subject, the study synthesised knowledge structures to determine the trend in the study issue. Using the Scopus database and a systematic search, 214 publications on diagnosing apple leaf disease were selected for scientometric analysis during 2011–2022. For the study, the web-based Biblioshiny software and the VOSviewer tool from the Bibliometrix suite were used. The software's automated workflow was used to select key journals, authors, countries, articles, and topics. In the publication by Alqahtani et al. [16], an Effective Sailfish Optimiser with Efficient Net-based Apple Leaf disease detection (ESFO-EALD) model has been developed. This technique, called ESFO-EALD, was designed to automatically detect plant leaf diseases. The quality of the photos of the apple plant leaves was improved in this instance by applying the Median Filtering (MF) technique. Additionally, the affected plant section in the test image was identified using SFO in conjunction with Kapur's entropy-based segmentation technique. Additionally, the apple plant leaf photos were detected and classified using the Adam optimiser using Spiking Neural Network (SNN)-based classification and EfficientNet-based feature extraction. To ensure that the ESFO-EALD technique produced effective results on the benchmark dataset, a series of simulations was run. Gong and Zhang [17] proposed an enhanced Faster region-based convolutional neural network (Faster R-CNN) technique. To extract

reliable, multi-dimensional features, a feature extraction network was introduced using the sophisticated Res2Net and feature pyramid network architectures [29]; [30].

In addition, RoIAlign was used in place of RoIPool to generate accurate candidate regions that address object location. Additionally, when concluding the photos, gentle non-maximum suppression was used to achieve accurate detection of apple leaf disease. Compared with previous object detection techniques, which achieved a standard precision of 63.1%, the improved Faster R-CNN architecture performed well on the analysed apple leaf disease dataset. To accurately detect early apple leaf disease spots, Liu et al. [18] developed a novel detection model called Representation-Enhanced RCNN (RE-RCNN). In an object-enhanced branch, the small disease spot feature extractor (SDSFEE) was first introduced to improve feature extraction for small disease spots. Secondly, to equalise the variances across classes of varying sizes of disease spots within the same category, an SCMLoss was proposed. Thirdly, during the training phase, a one-to-one sampling technique was used to obtain a representative sample. To detect apple leaf disease at the tiny target level, Gao et al. [19] created the TTALDD-4 dataset, which included four disease types: Leaf spots caused by Alternaria, Frogeye, Grey, and Rust. They also suggested the YOLOv7-tiny benchmark serves as the foundation for the HSSNet detector. First, it was suggested to use the image's cluttered foreground to highlight the lesions using the H-SimAM attention mechanism. Second, the SP-BiFormer Block was suggested to improve the model's comprehension of the minuscule targets of leaf diseases. Lastly, the prediction box bias example was strengthened by applying the SIOU loss. HSSNet achieved 85.04% mAP (mean average precision), 67.53% AR (average recall), and 83 FPS (frames per second) in the experimental results. HSSNet achieved a higher real-time detection rate and greater accuracy than other standard detectors. This served as a resource for the automated management of illnesses affecting apple leaves [31].

A unique design for the purpose of detecting plant foliar diseases, the hybrid random forest Multiclass SVM (HRF-MCSVM) design in the study by Sahu and Pandey [20]. Before classification, the image features were preprocessed and segmented using Spatial Fuzzy C-Means to improve computational accuracy, using the Plant Village dataset, which included 54,303 images of both healthy and sick leaves. Ultimately, performance metrics such as recall, accuracy, F-measure, specificity, and sensitivity were used to gauge the system's performance. A new, deeper, lightweight convolutional neural network architecture (DLMC-Net) has been presented in the paper by Sharma et al. [21] for plant leaf disease detection across multiple crops, enabling real-time agricultural applications. To extract deep features, the passage layer and a series of collective blocks were added to the suggested model. The vanishing gradient problem was resolved through these advantages in feature dissemination and reuse. To reduce the total number of trainable parameters, pointwise and separate convolutional blocks were also employed. Four publicly available datasets were used to verify the efficacy of the proposed DLMC-Net model: grapes, tomatoes, cucumbers, and citrus. Eight parameters were used to compare the experimental results of the proposed model with seven state-of-the-art models: F1-score, accuracy, precision, recall, sensitivity, specificity, and Matthews correlation coefficient. Experiments showed that the proposed model outperformed all other models considered, even under complex background conditions, with accuracies of 93.56%, 92.34%, 99.50%, and 96.56% for citrus, cucumber, grapes, and tomatoes, respectively.

3. Proposed Methodology

The diagram in Figure 1 illustrates the stages involved in implementing the recommended technique.

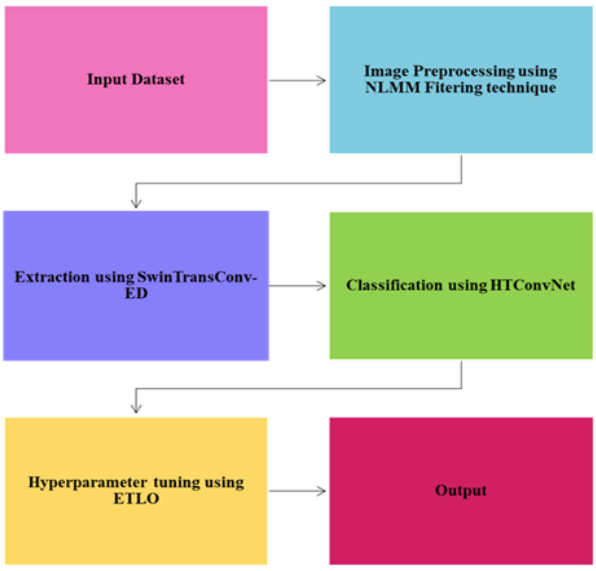

Figure 1: Workflow of the proposed model

This section covers image preprocessing, SwinTransConv-ED feature extraction, HTConvNet classification, and ETLO hyperparameter tuning (workflow).

3.1. Dataset Description

An expanded dataset was produced by Cornell University's Department of Plant Pathology and plant-microbiome biology, which is part of the publicly available dataset used in this study [22]. There are 3651 high-quality photos of apple leaves from the original dataset with various foliar diseases. To represent a more comprehensive dataset that covers the majority of scenarios, the images are manually captured in a variety of settings, including changing lighting, angles, and surfaces. The 3642 photos of apple leaves in the dataset used in this study are categorised into four groups: apple scab, healthy leaves, multiple diseases, and cedar apple rust. Just 5% of the 3642 photos that the study has access to show that a plant has more than one disease, such as both rust and scab. The other three classes are proportionately equal and include apple scab, cedar apple rust, and healthy. The primary objective of this research is to classify apple leaves into the four categories listed below correctly.

3.1.1. Healthy

Figure 2 illustrates how healthy leaves are completely pristine, green, and free of any disease-related symptoms.

Figure 2: Healthy leaf

3.1.2. Apple Scab

A leaf affected by apple scab disease on an apple tree. There are brown marks/spots on the leaves. Fruits that have fungal infections on their leaves or inside of them become unfit for consumption, which frequently results in scabs (Figure 3).

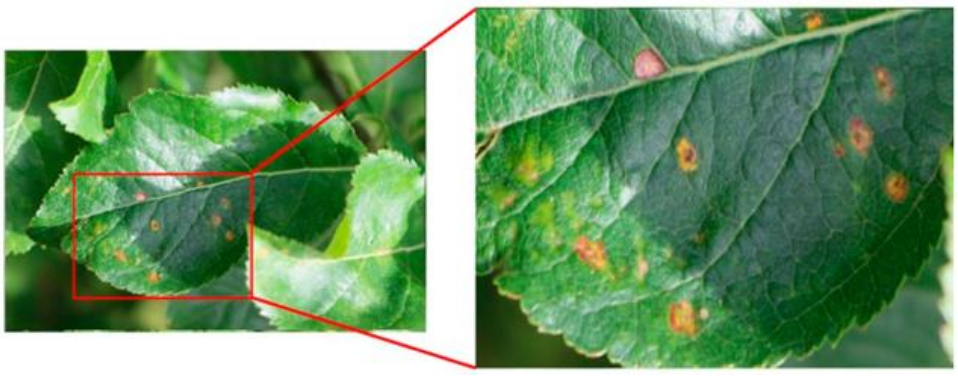

Figure 3: Apple scab-covered leaf

3.1.3. Cedar Apple Rust

Cedar apple rust on a leaf of an apple tree is depicted in Figure 4. The leaves are heavily speckled with yellow. Rust fungus is a type of fungus that often causes rust on plants.

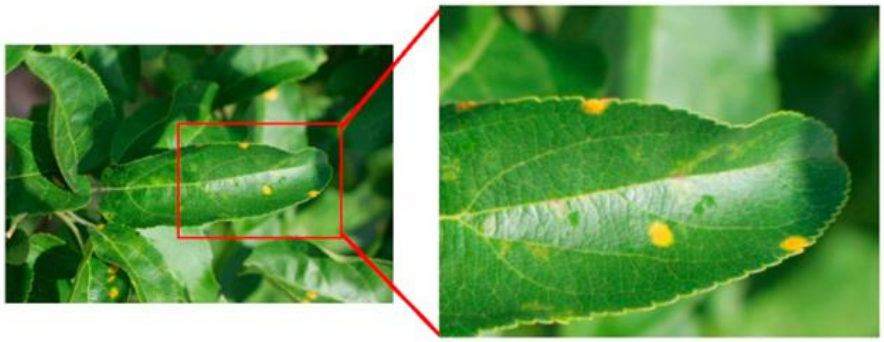

Figure 4: Russet cedar apple leaf

3.1.4. Multiple Diseases

As seen in Figure 5, leaves with multiple illnesses exhibit symptoms of both cedar apple rust, which is represented by yellow marks, and apple scab, which is represented by brown spots. In this instance, the leaves are seriously damaged and very challenging to treat.

Figure 5: Leaf with multiple diseases

3.2. Pre-Processing Using the NLMM Filter

The NLMM filtering technique employs a pixel-by-pixel weighting system to utilise every pixel in the leaf image. The image's structural information is better preserved after filtering, with high clarity and detail retention [23]. The picture of the noise is displayed as $v(i)$ as the total of the picture $u(i)$ and the cacophony $n(i)$, whose average value, in the absence of noise pollution, is 0. $v(i)$ can be written as follows in equation (1):

$$v(i) = u(i) + n(i) \quad (1)$$

Regarding a specific pixel i in a picture v, the picture block N(i), sized n×n, is a block of images centred on i, and N(j) is a picture block located close to N(i). The Gaussian-weighted Euclidean distance among the image blocks determines how similar images I and J are to one another, N(i) and N(j). The closer the space is between N(j) and N(i), the more alike the individual pixels j are to the pixel i, and the higher the weight that the pixel assigns to j in the cumulative restoration process. Considering the image that has been denoised to be I(i), regarding a pixel i, Equation (2) defines the NLMM calculation as follows:

$$I(i) = \frac{\sum_{j \in v} W(i,j)v(j)}{\sum_{j \in v} w(i,j)} \quad (2)$$

$v(N_i)$ is characterised as a rectangular community that is centred on i, as well as the correlation of similarity $w(i,j)$ among the pixels i and j in the picture v is shown as follows in equation (3):

$$w(i,j) = \exp\left(-\frac{\|v(N_i) - v(N_j)\|^2_{2,\alpha}}{h^2}\right) \quad (3)$$

where α is the Gaussian kernel function's standard deviation, $\|v(N_i) - v(N_j)\|_{2,\alpha}^2$ indicates the weighted distance calculated by Euclid between two blocks of images in equation (4); h is a parameter for filtering that controls the smoothness:

$$\|V(N_i) - V(N_j)\|^2 = \frac{1}{d^2}\sum_{i+z\in N_i, j+z\in N_j} \| v(i+z) - v(j+z) \|^2 \quad (4)$$

The texture and edges of the image can be effectively preserved by the non-local mean filtering algorithm, which fully utilises the image's block information. There is an improved filtering effect. But similarity measurement is not robust. The study proposes a novel denoising technique, NLMM, by substituting the weighted Euclidean distance between the two picture blocks for the grey difference. The structure of leaf image representations, as indicated by equation (5), has been thoroughly studied through the use of moments and associated invariants:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p f(x,y)dxdy,\ p,q = 0,1,2,\cdots \quad (5)$$

Seven-moment invariants were introduced from equation (6) to equation (13):

$$M = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7\} \quad (6)$$

$$\phi_1 = \eta_{20} + \eta_{02} \quad (7)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (8)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (9)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (10)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (11)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12}) + (\eta_{21} + \eta_{03}) \quad (12)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (13)$$

The useful property of remaining unchanged under translation, rotation, and scaling of leaf images is known as moment invariants. When a weighted Euclidean distance separates two image blocks, the result is:

$$\|V(N_i) - V(N_j)\|^2 = \frac{1}{d^2}\sum \|v(N_{i,M}) - v(N_{j,M})\|^2 \quad (14)$$

where $N_{i,M}$ is the picture block's moment of value N_i and $N_{j,M}$ is the picture block's moment value N_j as shown in equation (14).

3.3. Feature Extraction Using SwinTransConv-ED

This section aims to elucidate the model's structure for the extraction process. This study examines various feature-extraction techniques at multiple levels to obtain multiscale features from an apple leaf image. After that, a CNN is used for low- and mid-range features, a Swin Transformer for high-level features, and, additionally, a three-branch network is constructed using reconstruction features and an encoder-decoder.

3.3.1. Swin Transformer

The primary differences between the vision transformer (ViT) and the Swin Transformer lie in the underlying feature-mapping methods. Because ViT uses global self-attention, it produces unique low-resolution feature maps. It poses a quadratic computational challenge due to the dimensionality of the input apple leaf image. Conversely, Swin Transformers use an innovative method of combining image patches to create hierarchical feature maps, providing a clever workaround that linearly reduces the relationship between computational complexity and the input image's dimension. With this method, processing images of different sizes can be done more efficiently and is scalable, since self-attention computation is performed within local

windows. The model is well-suited for vision tasks such as image detection, segmentation, and classification, thanks to the Swin Transformer's architecture. Multiscale features can be extracted from the spatial dimension using swing transformers.

The swin transformer model divides the input image into non-overlapping patches. Initially, the dimensionality reduction technique is applied to the model's input to minimise computational complexity and data redundancy. Furthermore, dimensionality reduction is crucial because without it, overfitting happens to address the issue as mentioned above $x = \mathbb{R}^{m \times n \times b}$ is the model's input, where $m \times n$ represents the height and width of the apple leaf image, and b symbolises the bands. y shows the layer for dimensionality reduction and $y \in \mathbb{R}^{z \times 3} \cdot z$ signifies the quantity of bands. The patch token then processes the Swin Transformer's input. The model is composed of layer normalisation layers, an MLP layer, and shifted window self-attention. The number of tokens is reduced using a patch-merging layer, and spatial features are generated by deepening the model. Shifted window self-attention is the central component of the Swin Transformer. Shifted window techniques are used in tasks involving image classification. The following formulas are used to calculate the window partition process: (15) – (18):

$$\hat{z}^l = \mathrm{WMSA}\left(\mathrm{LN}\left(z^{l-1}\right)\right) + z^{l-1} \quad (15)$$

$$z^l = \mathrm{MLP}\left(\mathrm{LN}\left(\hat{z}^l\right)\right) + \hat{z}^l \quad (16)$$

$$\hat{z}^{l+1} = \mathrm{SWMSA}\left(\mathrm{LN}\left(z^l\right)\right) + z^l \quad (17)$$

$$z^{l+1} = \mathrm{MLP}\left(\mathrm{LN}\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \quad (18)$$

where $\hat{z}^l$ serves as the window multihead output of self-attention. z^l the MLP results at the lth block are shown in Sun et al. [24].

3.3.2. CNN

CNN is accurate at extracting leaf features. Suppose that the picture input is $X \in \mathbb{R}^{H \times W \times C}$, where $H \times W$ symbolises the image's height and width, and C symbolises the image's channels. Small patches are created from the input reduced image, and this process requires patch generation $D \in \mathbb{R}^{S \times S \times D}$ centred at the point where in space (a, b), which addresses the size of the spatial window $(s \times s)$. Given the input feature weights and an M convolution kernel w_iEquation (19) allows the output to be calculated as follows:

$$Y = \delta(w_i * D) \quad (19)$$

Wherein δ indicates the function of activation. The max pooling layer is used after the convolution layer, reducing the area and extracting additional distinguishing features. MP stands for the max pooling operation, represented in equation (20) and illustrated below:

$$p = \mathrm{MP}(Y) \quad (20)$$

The incoming batches are normalised by the geographic batch normalisation layer after reduction, which speeds up model training. After the normalisation layer, the output neurons are given non-linearity via a rectified linear unit (ReLU) activation function. The CNN layer, batch normalisation layer, MXPOOL layer, and ReLU layer form a single convolutional block. Block three in this paper has a filter size of (8, 16, 32), and all blocks use the same stride kernel size (3×3).

3.3.3. Encoder-Decoder (ED)

The unsupervised feature extraction technique known as the encoder-decoder makes up the third component of the model [25]. There are two common encoder-decoder architectures: fully convolutional and fully connected. This paper uses a band attention module (BAM) in conjunction with a fully connected method. Rebuilding the band information is the core idea underlying this kind of encoder-decoder. This entails extracting all leaf information from a small set of useful bands. Three essential parts make up the overall architecture: the reconstruction network (RecNet), the band reconstruction weights (BRW), and the band attention module (BAM). The function of the band attention module (BAM) is g. Input X yields weights that are not negative, and the tensor outline is $w \in \mathbb{R}^{1 \times 1 \times b}$:

$$w = g(X; \theta_b) \quad (21)$$

where θ_b the trainable parameter of the BAM is indicated in equation (21). In the output layer of the BAM module, the sigmoid function is integrated using the following formulation to ensure that the weights obtained are required to be non-negative:

$$\phi(w) = \frac{1}{1+e^{-w}} \quad (22)$$

An operation known as band-wise multiplication, or BRW, is performed to create a link between the original inputs as well as their corresponding weights. Equation (23) provides a concise description of this operation:

$$h = X \otimes w \quad (23)$$

The investigation then continues by using RecNet to recreate the original multiband from the equivalent reweighted multiband. RecNet is similarly described as a function indicated by h that takes a reweighted tensor as input. That takes a reweighted tensor y as input, as shown in equation (24), and generates the corresponding predictions:

$$\hat{X} = h(y; \theta_r) \quad (24)$$

All that's needed for feature reconstruction is an MLP with the ReLU activation function and the same number of hidden neurons.

3.3.4. Weight Fusion

Three branch-extracted features are found in this CNN paper: a fully connected encoder-decoder and the Swin Transformer. These branches may display different data features or attributes. Giving each branch a suitable amount of weight when classifying the outcome is the aim of the weighted fusion technique. The study begins by evaluating each branch's significance. This might depend on how pertinent the data it gathers is. Multiplying the data from each branch by the appropriate weights after the importance scores have been determined. Equation (25) computes the weight fusion formulation by combining these features through a sum operation:

$$F_1 = \lambda \times F^{CNN} + (1 - \lambda) \times F^{ED} \quad (25)$$

where F_1 indicates two CNN branches, Features of ED Fusion, and λ represents the parameter range, with a value between [0,1]:

$$F_2 = \lambda \times F_1 + (1 - \lambda) \times F^{Transfromer} \quad (26)$$

where F_2 The result of combining the three branches is as indicated by equation (26). Once more, F_1 it can grow when using transformer branch characteristics. After extraction, classification is performed using HTConvNet, achieving exceptional accuracy in leaf disease image classification.

3.4. Classification Using HTConvNet

This work suggests the HTConvNet architecture for classifying 1D pixel images. HTConvNet consists of a Transformer, an FC layer, a CNN, and a skip connection. A leaf-pixel image is sent to the FC in the first step, which then uses the fully connected layer's linear transformation to change the 1D leaf image's dimensionality and transform it into a 3D matrix of leaf features. The characteristics of three distinct 2D CNN levels are extracted in the second step and fed into the transformer. In the third step, the position encoding is added to the CNN-extracted features along with the pixel information. To accomplish feature fusion, the connections between neighbouring transformer encoders are skipped in the fourth step. At last, the MLP head achieves pixel-level classification by producing classification results. The HTConvNet is composed of a transformer, a feature fusion module, a CNN, and an FC layer.

3.4.1. Feature Fusion Module

The use of previously acquired features in deeper layers is enabled by skip connections (SC), which also help minimise information loss by fusing multiple features. It has been demonstrated that skip connections are a successful technique in classical networks, such as ResNet and U-Net. However, the noteworthy characteristic of Disparities in long skip connections could undermine effective performance, thereby being detrimental to classification. For this reason, the Transformer primarily uses short- and medium-distance skip connections. LeafFormer uses a medium-distance skip-link technique called cross-layer

adaptive fusion (CAF) [26]. CAF omits one encoder, combining two feature layers that are separated by one encoder. In contrast to CAF, the study achieves feature fusion via short-hopping links that maximise feature reuse across neighbouring layers and minimise information loss. The feature fusion module splits up the prior encoder's features. (E^{i-1}) with the encoder's features as they are now $(E^{i})\left(\text{Concat}(E^{i-1}, E^{i})\right)$, and after that uses a convolution layer to alter the dimensionality $\left(\text{Conv}\left(\text{Concat}(E^{i-1}, E^{i})\right)\right)$ to acquire the post-fusion features. Equation (27) allows for its expression to be as follows:

$$\hat{E} = \text{Conv}\left(\text{Concat}(E^{i-1}, E^{i})\right) \quad (27)$$

where $E^{i}, E^{i-1} \in \mathbb{R}^{n\times dim}$ is the extension's dimension $E^{i}, E^{i-1} \in \mathbb{R}^{n\times dim\times 1}$, $\text{Concat}(E^{i-1}, E^{i}) \in \mathbb{R}^{n\times dim\times 2}$, Convolution's kernel of convolution is $[1,2], \hat{E} \in \mathbb{R}^{n\times dim}$.

3.4.2. 2D CNN-equipped Fully Connected Layer

The picture of the leaf can be expressed as $X = .\{x_i\}_{i=1}^{H\times W\times 1} \in \mathbb{R}^{H\times W\times 1}$, where C represents the total number of bands in the leaf image and H and W stand for the image's height and width, and $x_i \in \mathbb{R}^{1\times 1\times C}$ represents the sequence of pixels i. Only a 1D CNN can classify 1D pixel images, since they are not processed. If a 2DCNN is utilised for classification, either the 1D image must be transformed into a 2D/3D leaf characteristic matrix, or a sample consisting of a patch of crucial pixels must be used. In this work, 1D pixel pictures are made more dimensional by adding a fully connected layer. It is then utilised as the 2D CNN input after conversion to a 3D pixel-characteristic matrix:

$$y_i = wx_i + b \quad (28)$$

where w and b represent the fully connected layer's weight matrix as well as bias vector, respectively, x_i is the sequence of 1D pixels input, $y_i \in \mathbb{R}^{1\times 1\times m}$, and m is the fully connected layer's output dimension, and reshapes y_i as $y_i', y_i' \in \mathbb{R}^{n\times n\times \frac{m}{n^2}}; m = 256$ and $n = 4$ in this paper. One uses the network structure to redesign a shallow 2D CNN because transformer training takes longer. It can also classify leaf images, despite its straightforward network structure and low processing power requirements. The first layer of convolution contains 64 convolutional kernels, each of which is 1×1. A larger input sample dimension is achieved by using the first convolutional layer; 64 convolutional kernels in 3×3 sizes are found in the second convolutional layer. A SE module is attached to the subsequent convolutional layer following the application of the second convolution. The goal is to create more effective feature information. Through residual connection, the first and second convolutional layers combine to form a residual unit. To reduce redundant data and expand the receptive field, the residual unit is connected to a 2x2 average pooling layer. The features are compressed into a vector after the two remaining units undergo averaging pooling. Following the extraction of each residual unit's features, the features that are extracted $(n \times n \times m$ and $n/2 \times n/2 \times m)$ are transformed into serial information $(1 \times 1 \times 64)$, utilising a pooling mechanism to calculate the global average $y_i' \in \mathbb{R}^{n\times n\times \frac{m}{n^2}}$ is the CNN's input, $n = 4, m = 64$, and the CNN's output is $1 \times 1 \times 64$.

3.4.3. Transformer

Transformer's image transformation differs from that of an RNN because it uses positional encoding and a self-attention mechanism. As a result, it does not require maintaining consistency in the combined length of the output and input sequences, which offers major advantages when managing sequence data. A transformer can acquire adaptable spatial information thanks to its structure. In domains of image processing, such as semantic segmentation and image classification, Transformers have demonstrated strong performance [27]-[30]. ViT's particularly strong performance has aided the development of Transformers in image classification. A transformer encoder consists of three components: an input, a feedforward neural network, and an attention mechanism. In this paper, the Vision Transformer uses a transformer encoder. Encoding positions and embedding input features are part of the input phase. Transformer creates long-term dependencies via location encoding, and the self-attention mechanism associates relevant information. Multi-headed attention is another name for the self-attention mechanism. Algorithm 1 outlines four steps for implementing a sequence of pixels in the transformer (ViT):

- **Step 1:** Sequence image input x, and m is the number of bands. For each of the m vectors that result from dividing this, pull up linearly to create a new x.
- **Step 2:** After adding the location-encoding data to x, add the location code to obtain the updated x.
- **Step 3:** Raise the dimension to 3 using a linear transformation. Next, Q, K, and V are reshaped and normalised by the number of heads to apply the multi-head technique; these are then used as input to self-attention after normalisation.

- **Step 4:** For instance, take a head, calculate Q and K as an inner product, and divide the resulting inner component by $\sqrt{\dim}$Apply the Softmax activation function, then multiply the result by V to get the resultant value Z_i:

$$\text{Attenion}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (29)$$

3.4.4. Hyperparameter Tuning Using ETLBO

This section provides an extensive overview of the study's hyperparameter tuning procedure. The TLBO and the tuning strategies are first introduced in the study. Next, the ETLBO is presented [31].

3.4.4.1. Teacher Phase

The learner who scored the highest marks in this section of the algorithm is the teacher, and their job is to raise the class mean score. Equation (30) formulates the i-th learner's update process during the teacher phase as follows:

$$X_{i,\,new} = X_i + \text{rand} \times (X_{teacher} - T_F \times X_{ave}) \quad (30)$$

where, X_i is the i-th learner's solution, $X_{teacher}$ symbolises the instructor's resolution, X_{ave} indicates the mean of all students, rand is a random integer in the range of 0 to 1, and T_F is the instructional component that determines whether to modify the mean's value. It is once more a heuristic step with equal probability that determines the value, which can be either 1 or 2.

$$T_F = \text{round}[1 + \text{rand}(0,1)\{2 - 1\}]$$

Furthermore, the new remedy $X_{i,\,new}$ is only approved if it performs better than the earlier fix, and it is expressed as follows in equation (31):

$$X_i = \begin{cases} X_{i,\,new}\, f(X_{i,\,new}) > f(X_i) \\ X_i \text{ otherwise} \end{cases} \quad (31)$$

Where the fitness function is denoted by f.

3.4.4.2. Learner Phase

In the latter portion of the algorithm, the learner interacts with other learners to update its knowledge base. Every time around, two students engage with X_m and X_n, whereby the more creative learner raises other students' grades. When another learner is more knowledgeable than the first, one learner gains new information during the learner phase. The phenomenon is explained as follows in equation (32):

$$X_{m,new} = \begin{cases} X_m + \text{rand} \times (X_m - X_n); f(X_m) > f(X_n) \\ X_m + \text{rand} \times (X_n - X_m); f(X_n) > f(X_m) \end{cases} \quad (32)$$

The interim solution, which can be expressed in equation (33) as follows, is only approved if it outperforms the initial solution:

$$X_m = \begin{cases} X_{m,\,neww}\,; f(X_{m,\,new}) > f(X_m) \\ X_m;\ \text{otherwise} \end{cases} \quad (33)$$

3.4.4.3. Adaptive Weight Strategy

The adaptable weight approach makes global optimisation easier by overcoming local minima. Although the TLBO resolves the complex optimised function issue, the algorithm is prone to quickly reaching the local optimal state. Furthermore, for accurate local search within the current search domain, a lower inertia factor is advantageous. The research developed a novel weight strategy that can be expressed as follows:

$$t = \left(1 - \frac{\text{iter\_ater}}{\text{Max\_iter}}\right)^{1-\sin\left(\pi\frac{\text{iter}}{\text{Max\_itar}}\right)} \quad (34)$$

Wherein iter denotes the iteration's current number, and Max_iter denotes its maximum number.

3.4.4.4. Kent Chaotic Map (KCM)

Nonlinear mappings, such as chaotic mappings, can produce unpredictable number sequences. Because of its sensitivity to beginning values, the encoder can produce a random encoding sequence. Chaotic maps come in a variety of forms, including the Logistic and the Henon maps. The improved strategy in this paper is the Kent map. Equation (34) displays the Kent map formula as follows:

$$f(x) = \begin{cases} \frac{x}{a} & 0 < x \leq a \\ \frac{1-x}{1-a} & a < x < 1 \end{cases} \tag{35}$$

wherein, a is not constant, x is the starting value of the $x(0)$. In this paper, $a = 0.5$.

3.4.5. Proposed ETLBO

The standard TLBO search procedure to update the person's position consists of two stages. The teacher's initial state is enhanced during the teacher phase by utilising the chaotic map of Kent. To teach the various students, the teacher may possess a variety of skills. This tactic showcases educators' skills. The study enhances learning efficiency during the learner phase, thereby improving the learning conditions for pupils. With more iterations, the adaptive weight approach can improve. At the start of the iteration, the students will pick up new information. By the end of the cycle, the students can learn enough, and the adaptive weight decreases. At various stages, students can acquire different knowledge. Equation (36) can be used to represent the formula as follows:

$$X_{m,new} = \begin{cases} X_m \times t + \text{ rand } \times (X_m - X_n); f(X_m) > f(X_n) \\ X_m \times t + \text{ rand } \times (X_n - X_m); f(X_n) > f(X_m) \end{cases} \tag{36}$$

Where t is the adaptive weight, hence, effective hyperparameter tuning for a classifier is achieved using ETLBO.

4. Results and Discussion

4.1. Experimental Setup

Using an NVIDIA RTX 3060 GPU and 64 GB of RAM, the extraction process was conducted. The TensorFlow library was used with Python 3.8 in this work. AMD Ryzen 9 5900HX, coupled with NVIDIA GeForce RTX 3080 laptop graphics and Windows 11, is used to train the model, and Python 3.7 is used to implement all the implementation code for the classification process in PyTorch 1.12.1. MATLAB 2018 b is used to train the proposed ETLBO. The computer's central processing unit (CPU) used to manage the experiments is an i7-11800H.

4.2. Analysis of Classification with Existing Models

Table 1 provides a thorough analysis of the accuracy of the different models for leaf disease classification. The nine models that make up this system are CNN using DenseNet, CNN using ResNet, XDNet, CNN using MobileNet, CNN using VGG, MSO-ResNet, DenseNet-201, EfficientNet, and the Proposed ETLO-based HTConvNet. Every model has a corresponding accuracy percentage that it attained during assessment. DenseNet is combined with a CNN to achieve 93.71% accuracy, a strong performance. XDNet achieves an impressive accuracy of 98.82%.

Table 1: Classification analysis with existing models

Models	Accuracy (%)
CNN using DenseNet [32]	93.71
CNN using ResNet [33]	83.75
XDNet [34]	98.82
CNN using MobileNet [35]	73.50
CNN using VGG [36]	99.01
MSO-ResNet [37]	95.70
DenseNet-201[38]	98.75
EfficientNet [39]	99.11
Proposed ETLO-based HTConvNet	99.74

In contrast, the CNN employing ResNet attains 83.75% accuracy, the CNN employing MobileNet attains 73.50%, and the CNN employing VGG attains 99.01%. At 95.70%, MSO-ResNet's accuracy falls on the higher end of the accuracy spectrum. DenseNet-201 and EfficientNet achieve accuracies of 98.75% and 99.11%, respectively, indicating high proficiency on the task. The last entry, HTConvNet, based on the proposed ETLO, stands out for its remarkable 99.74% accuracy. This model exhibits superior accuracy compared to the other models evaluated in the table, indicating better performance in apple leaf image classification. The proposed ETLO-based HTConvNet emerged as the top-performing model in this assessment. The table provides a clear comparative overview of the models' performance, highlighting the advantages and disadvantages of each in terms of accuracy.

Table 2: Performance metrics analysis on various classes

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Healthy leaf	97.81	97.50	98.23	98.41
Apple scab	98.21	98.56	98.72	98.65
Cedar apple rust	98.89	99.21	99.46	99.52
Multiple diseases	99.74	99.72	99.70	99.71

Table 2 provides a comprehensive summary of classification metrics for a model designed to distinguish among different leaf conditions. A particular class of leaf condition includes healthy leaf, apple scab, cedar apple rust, and multiple diseases, represented by each row. Several performance metrics, expressed as percentages, are displayed in the columns: accuracy, precision, recall, and F1-score. Upon examining the Healthy leaf category, the model achieves an impressive 97.81% accuracy, indicating the percentage of healthy leaves correctly classified out of all leaves. With a precision of 97.50%, which gauges the accuracy of positive predictions, the model is reliable at classifying healthy leaves. The model's recall for "healthy leaf" is 98.23%, indicating it can accurately distinguish healthy from unhealthy leaves. Furthermore, the model's strong performance in identifying healthy leaves is demonstrated by its F1-score of 98.41%, which strikes a notable balance between precision and recall. For Apple scab, the proposed model shows even better overall performance metrics. With 98.21% accuracy, 98.56% precision, 98.72% recall, and 98.65% F1-Score, the proposed model performs exceptionally well at identifying this leaf condition. The performance of cedar apple rust is even more remarkable: it achieves an F1-score of 99.52%, a recall of 99.46%, accuracy of 98.89%, and a precision of 99.21%. This demonstrates how well the model can identify and categorise cases of Cedar apple rust. Then the Multiple diseases category exhibits exceptional performance metrics, including an F1-Score of 99.71%, accuracy of 99.74%, precision of 99.72%, and recall of 99.70%. This suggests that the model is remarkably accurate at identifying leaves affected by multiple diseases simultaneously. Overall, the table shows that the model performs well at classifying leaf conditions; its ability to differentiate between various illnesses and healthy leaves is especially impressive. Based on the given features or characteristics, these metrics highlight the classification model's dependability and efficacy in identifying leaf conditions.

Table 3: Performance metrics analysis on different models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Autoencoder	96.51	96.23	95.90	96.30
RNN	96.89	96.72	96.67	96.81
Swin Transformer	97.54	97.55	97.49	97.43
CNN	98.41	98.23	98.40	98.33
Proposed ETLO-based HTConvNet	99.74	99.72	99.70	99.71

From Table 3, all metrics indicate that the "Autoencoder" model performs well: its accuracy is 96.51%, and its precision, recall, and F1-score are 96.23%, 96.30%, and 95.90%, respectively. This model does a great job of comprehending and accurately representing the data's features. Next, with 96.89% accuracy, 96.72% precision, 96.67% recall, and 96.81% F1-score, the "RNN" (Recurrent Neural Network) performs slightly better overall. Because of their reputation for handling sequential data, RNNs are useful in this situation. With an accuracy of 97.54%, the "Swin Transformer" model shows a significant improvement. Its precision, recall, and F1-score values are balanced at 97.55%, 97.49%, and 97.43%, respectively. Transformer models perform better because, like Swin, they are often very good at capturing long-range dependencies in data. Then, with an accuracy of 98.41%, the "CNN" (Convolutional Neural Network) performs even better. At 98.23%, 98.40%, and 98.33%, respectively, its precision, recall, and F1-score are remarkable. CNNs are widely used in image-related tasks because of their well-known capacity to extract spatial hierarchies and patterns from data. Finally, with an astounding 99.74% accuracy, the "Proposed ETLO-based HTConvNet" model stands out for its outstanding performance. At 99.72%, 99.70%, and 99.71%, respectively, its recall, accuracy, and F1-score are very high. This suggested model, which is probably a new architecture, performs exceptionally well, presumably by utilising sophisticated methods such as convolutional operations for classification.

This is evident in its exceptional accuracy and well-balanced metrics. To summarise, the table presents a comparative analysis of various models, demonstrating their varying performance in the assigned leaf disease classification task. Each model demonstrates its strengths in the classification process of leaf image, including sequential learning, data representation, spatial feature extraction, and specialised architectures.

5. Conclusion

Using an NLMM denoising filter, the suggested model removed noise from the images. For classifying leaf images, this paper proposed a DL extraction procedure that preprocesses using a parallel branch schema. Reconstructive leaf features, weight fusion strategies, and low- and high-level feature mining are all possible with this extraction process. HyTransConvNet is used for classification. To achieve pixel-level image classification, this paper focused on combining the characteristics of the Transformer and 2D CNN. To address the issue that 2D CNNs are unable to classify pixel-level images, this work presents a network framework utilising FC and 2D CNNs. Since 2-dimensional convolutional neural networks cannot achieve pixel-level accuracy, they typically classify pixels by treating neighbouring blocks of the target pixels as input. To improve image information, the classification process involves extracting and merging input waveform data from the transformer with features from different 2D CNN layers. At a deeper level, information is passed from adjacent transformer encoders via skip connections, which minimise information loss. After classification, the ETLBO algorithm is used for tuning. The TLBO is improved by applying the Kent chaotic map and an adaptive weight strategy. An improved teaching-learning-based optimisation is put forth in this paper. The TLBO is improved by applying the Kent chaotic map and an adaptive weight strategy. The suggested model outperforms other current models, achieving 99.74% accuracy. For farmers in remote areas where it is difficult to access plant pathologists' services, this app is very helpful. The proposed study aims to broaden the model's focus to include additional categories of foliar illnesses in subsequent research. To create better models in the future, you also plan to expand the dataset by adding more pictures of apple leaves.

Acknowledgement: The authors express their sincere gratitude to New Horizon College of Engineering, Quest Technologies, Saranathan College of Engineering, and IPS Health for their continuous support and encouragement throughout this work. Their collective guidance and resources significantly contributed to the successful completion of this research.

Data Availability Statement: The datasets generated and analyzed during this study are available from the authors upon reasonable request, ensuring openness and support for future research.

Funding Statement: The authors affirm that this research was carried out without external financial support and that all work was completed using institutional and personal resources.

Conflicts of Interest Statement: The authors collectively declare that they have no competing interests that could have influenced the outcomes or interpretation of this research.

Ethics and Consent Statement: All authors jointly agree to share this publication for academic, educational, and research purposes, ensuring it remains accessible to interested readers.

Reference

1. Wikimedia Foundation, “Apple,” *Wikipedia*, 2021. Available: https://en.wikipedia.org/wiki/Apple [Accessed by 21/08/2024].
2. S. Parez, N. Dilshad, N. S. Alghamdi, T. M. Alanazi, and J. W. Lee, “Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers,” *Sensors*, vol. 23, no. 15, p. 6949, 2023.
3. W. Chen, D. Modi, and A. Picot, “Soil and phytomicrobiome for plant disease suppression and management under climate change: A review,” *Plants*, vol. 12, no. 14, p. 2736, 2023.
4. P. Guzmán-Guzmán, A. Kumar, S. de Los Santos-Villalobos, F. I. Parra-Cota, M. D. C. Orozco-Mosqueda, A. E. Fadiji, S. Hyder, O. O. Babalola, and G. Santoyo, “Trichoderma species: Our best fungal allies in the biocontrol of plant diseases—A review,” *Plants*, vol. 12, no. 3, p. 432, 2023.
5. H. You, Y. Lu, and H. Tang, “Plant disease classification and adversarial attack using SimAM-EfficientNet and GP-MI-FGSM,” *Sustainability*, vol. 15, no. 2, p. 1233, 2023.
6. A. Ahmad, D. Saraswat, and A. El Gamal, “A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools,” *Smart Agricultural Technology*, vol. 3, no. 2, p. 100083, 2023.
7. O. Attallah, “Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection,” *Horticulturae*, vol. 9, no. 2, p. 149, 2023.

8. M. Aggarwal, V. Khullar, N. Goyal, A. Alammari, M. A. Albahar, and A. Singh, “Lightweight federated learning for rice leaf disease classification using non-independent and identically distributed images,” *Sustainability*, vol. 15, no. 16, p. 12149, 2023.
9. S. Parez, N. Dilshad, N. S. Alghamdi, T. M. Alanazi, and J. W. Lee, “Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers,” *Sensors*, vol. 23, no. 15, p. 6949, 2023.
10. H. Wang, S. Qiu, H. Ye, and X. Liao, “A plant disease classification algorithm based on Attention MobileNet V2,” *Algorithms*, vol. 16, no. 9, p. 442, 2023.
11. A. Guerrero-Ibañez and A. Reyes-Muñoz, “Monitoring tomato leaf disease through convolutional neural networks,” *Electronics*, vol. 12, no. 1, p. 229, 2023.
12. D. Zhu, J. Tan, C. Wu, K. Yung, and A. W. H. Ip, “Crop disease identification by fusing multiscale convolution and vision transformer,” *Sensors*, vol. 23, no. 13, p. 6015, 2023.
13. H. Wang, J. Ding, S. He, C. Feng, C. Zhang, G. Fan, Y. Wu, and Y. Zhang, “MFBP-UNet: A network for pear leaf disease segmentation in natural agricultural environments,” *Plants*, vol. 12, no. 18, p. 3209, 2023.
14. H. Yin, Y. H. Gu, C. J. Park, J. H. Park, and S. J. Yoo, “Transfer learning-based search model for hot pepper diseases and pests,” *Agriculture*, vol. 10, no. 10, p. 439, 2020.
15. A. Bonkra, P. K. Bhatt, J. Rosak-Szyrocka, K. Muduli, L. Pilar, A. Kaur, N. Chahal, and A. K. Rana, “Apple leave disease detection using collaborative ML/DL and artificial intelligence methods: Scientometric analysis,” *Int. J. Environ. Res. Public Health*, vol. 20, no. 4, p. 3222, 2023.
16. M. M. Alqahtani, A. K. Dutta, S. Almotairi, M. Ilayaraja, A. A. Albraikan, F. N. Al-Wesabi, and M. Al Duhayyim, “Sailfish optimizer with EfficientNet model for apple leaf disease detection,” *Computers, Materials and Continua*, vol. 74, no. 1, pp. 217-233, 2023.
17. X. Gong and S. Zhang, “A high-precision detection method of apple leaf diseases using improved Faster R-CNN,” *Agriculture*, vol. 13, no. 2, p. 240, 2023.
18. B. Liu, H. Ren, J. Li, N. Duan, A. Yuan, and H. Zhang, “RE-RCNN: A novel representation-enhanced RCNN model for early apple leaf disease detection,” *ACM Transactions on Sensor Networks*, 2023. Available: https://dl.acm.org/doi/pdf/10.1145/3587466 [Accessed by 12/08/2024].
19. X. Gao, Z. Tang, Y. Deng, S. Hu, H. Zhao, and G. Zhou, “HSSNet: An end-to-end network for detecting tiny targets of apple leaf diseases in complex backgrounds,” *Plants*, vol. 12, no. 15, p. 2806, 2023.
20. S. K. Sahu and M. Pandey, “An optimal hybrid multiclass SVM for plant leaf disease detection using spatial fuzzy C-Means model,” *Expert Syst. Appl.*, vol. 214, no. 3, p. 118989, 2023.
21. V. Sharma, A. K. Tripathi, and H. Mittal, “DLMC-Net: Deeper lightweight multi-class classification model for plant leaf disease detection,” *Ecological Informatics*, vol. 75, no. 7, p. 102025, 2023.
22. R. Thapa, N. Snavely, S. Belongie, and A. Khan, “Plant Pathology 2020-FGVC7: Identify the category of foliar diseases in apple trees”, *Kaggle,* 2021. Available: https://www.kaggle.com/c/plant-pathology-2020-fgvc7/data/ [Accessed by 22/08/2024].
23. C. Liu and L. Zhang, “A novel denoising algorithm based on wavelet and non-local moment mean filtering,” *Electronics*, vol. 12, no. 6, p. 1461, 2023.
24. L. Sun, G. Zhao, Y. Zheng, and Z. Wu, “Spectral–spatial feature tokenization transformer for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, no. 1, pp. 1-14, 2022.
25. J. Zhang, Z. Meng, F. Zhao, H. Liu, and Z. Chang, “Convolution transformer mixer for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 19, no. 9, pp. 1-5, 2022.
26. R. Hang, Q. Liu, D. Hong, and P. Ghamisi, “Cascaded recurrent neural networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, 2019.
27. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16×16 words: Transformers for image recognition at scale,” *arXiv:2010.11929*, 2020, Available: https://arxiv.org/abs/2010.11929 [Accessed by 22/08/2024].
28. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, Quebec, Canada, 2021.
29. S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, and Y. Wang, “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in *Proc. CVPR*, Nashville, Tennessee, United States of America, 2021.
30. J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, “TransUNet: Transformers make strong encoders for medical image segmentation,” *arXiv:2102.04306*, 2021. Available: https://arxiv.org/abs/2102.04306 [Accessed by 08/08/2024].
31. D. Wu, H. Jia, L. Abualigah, Z. Xing, R. Zheng, H. Wang, and M. Altalhi, “Enhance teaching-learning-based optimization for Tsallis-entropy-based feature selection classification approach,” *Processes*, vol. 10, no. 2, p. 360, 2022.
32. Y. Zhong and M. Zhao, “Research on deep learning in apple leaf disease recognition,” *Comput. Electron. Agric.*, vol. 168, no. 1, p. 105146, 2020.

33. H. Jiang, Z. P. Xue, and Y. Guo, “Research on plant leaf disease identification based on transfer learning algorithm,” in *Proc. 4th Int. Conf. Artif. Intell., Autom. Control Technol. (AIACT)*, Hangzhou, China, 2020.
34. X. Chao, G. Sun, H. Zhao, M. Li, and D. He, “Identification of apple tree leaf diseases based on deep learning models,” *Symmetry*, vol. 12, no. 7, p. 1065, 2020.
35. C. Bi, J. Wang, Y. Duan, B. Fu, K. Jia-Rong, and Y. Shi, “MobileNet based apple leaf diseases identification,” *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 172–180, 2020.
36. Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, “Apple leaf diseases recognition based on an improved convolutional neural network,” *Sensors*, vol. 20, no. 12, p. 3535, 2020.
37. H. Yu, X. Cheng, C. Chen, A. A. Heidari, J. Liu, Z. Cai, and H. Chen, “Apple leaf disease recognition method with improved residual network,” *Multimed. Tools Appl.*, vol. 81, no. 6, pp. 7759–7782, 2022.
38. P. Pradhan, B. Kumar, and S. Mohan, “Comparison of various deep convolutional neural network models to discriminate apple leaf diseases using transfer learning,” *J. Plant Dis. Prot.*, vol. 129, no. 6, pp. 1461–1473, 2022.
39. Q. Yang, S. Duan, and L. Wang, “Efficient identification of apple leaf diseases in the wild using convolutional neural networks,” *Agronomy*, vol. 12, no. 11, p. 2784, 2022.