

MMDFN: A Multi-modal Deep Fusion Network with Hybrid Optimization for Automated Cotton Leaf Disease Detection

Mohan Ajmeera

Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada (JNTUK), Andhra Pradesh, India

E-mail: amohanphd2020@gmail.com

ORCID iD: <https://orcid.org/0000-0003-0283-305x>

*Corresponding author

P. Chiranjeevi

CSE Department, Amrita Sai Institute of Science and Technology, Bathinapadu, Andhra Pradesh, India

E-mail: mailmeparitala@gmail.com

ORCID iD: <https://orcid.org/0009-0006-8444-8356>

A. Krishna Mohan

CSE Department, University College of Engineering, Jawaharlal Nehru Technological University Kakinada (JNTUK), Andhra Pradesh, India

E-mail: Krishna.ankala@gmail.com

ORCID iD: <https://orcid.org/0000-0003-0517-3130>

Received: April 1, 2026; Revised: May 15, 2026; Accepted: July 7, 2026; Published: August 8, 2026

Abstract: This study presents the Multi-Modal Deep Fusion Network to identify cotton leaf diseases. Initially, the images are collected from the Kaggle cotton disease dataset. The dataset is preprocessed, and data augmentation is applied exclusively to the training set to prevent data leakage. The VGG-16-based Faster Region-based Convolutional Neural Network model is used for lesion detection and region of interest localization by generating bounding boxes around diseased areas. Both handcrafted features (including shape descriptors and color moments) and deep learning features are used for feature extraction. The extracted features are optimized using the Snowy Wolf Optimization algorithm which combines Snow Leopard Optimization and Grey Wolf Optimization. The proposed method achieved 98.4% accuracy, 98.6% sensitivity, and 98.8% F-score, consistently outperforming existing methods under identical experimental settings. While the proposed framework demonstrated promising performance on the evaluated dataset, further validation on larger and more diverse field datasets is required to comprehensively assess its generalization capability.

Index Terms: Disease Detection, Cotton Plant Leaves, Region of Interest, Snowy Wolf Optimization, Multi-modal Deep Fusion Network.

1. Introduction

Early prediction of disease in the leaves of cotton plants is important in crop health. Advanced AI and image-processing technology have transformed this process [1]. These systems can detect diseases in the early stage by examining leaf patterns, color variations, and other aspects, allowing for timely intervention to protect cotton yield [2]. Cotton is a major cash crop that is severely affected by many diseases, but most of them affect the cotton yield in terms of leaf-associated problems [3]. Climate change may exacerbate crop diseases through excessive heat, necessitating the timely application of pesticides to prevent further damage [4]. Globally grown cotton is the key to the economy and therefore confronts various agricultural pests and diseases, especially in tropical areas, and requires efficient solutions [5,6]. Pests or diseases affecting cotton are diagnosed based on distinctive spots and patterns on leaves [7]. Farmers detect the infections in the crops by their sight and follow previous experiences when selecting the right pesticides [8]. A wide variety of AI techniques is being investigated in recent studies for the automated detection of cotton leaf diseases, such

as machine learning, convolutional neural networks (CNNs), transfer learning, attention mechanisms, few-shot learning, and hybrid deep learning frameworks [9,10]. disease such as machine learning, convolutional neural networks (CNNs), transfer learning, attention mechanisms, few-shot learning, and hybrid deep learning frameworks [9,10]. These methods have significantly enhanced disease detection by extracting discriminative features from leaf images, decreased human intervention in the process and facilitated the early detection of disease [11,12]. Moreover, with advances made in image processing and feature learning, automated systems have increased in their ability to classify multiple disease categories with high accuracy [13]. However, a number of problems still remain in this field, such as the inaccuracy of lesion localization, the inconsistency of feature extraction, the limited integration of handcrafted texture, color, and shape descriptors with deep learning features, and the decreased robustness to different illumination, background complexity, and disease severity that occur in real-world field conditions [14,15]. These limitations emphasize the need for a more reliable and universal framework that can accurately localize the disease, successfully fuse features, and accurately classify the disease.

Although cotton leaf disease detection has made great progress in the field of deep learning, there are still some key problems to be solved. Existing CNN and transformer-based models mostly focus on extracting automatically learned deep features and ignore complementary handcrafted features, such as texture, color, and shape that are rich in disease-specific information. As a result, some of these approaches are not effective in differentiating among diseases with very similar visual features and fine-grained lesion textures. Furthermore, accurately locating diseased areas remains challenging. Most current methods classify the entire image of a leaf, which can result in unwanted background features and changes in illumination affecting the learning of useful features. This is addressed in certain ways by attention mechanisms, but not specifically by focusing on lesion regions prior to feature extraction. Another issue is the high dimensionality and redundancy of fused feature representations. Previous studies have used conventional feature selection techniques or essentially performed no feature optimization, leading to redundant information, poor generalization performance in practical agricultural settings, and increased computational complexity. Hence, an integrated framework that can accurately localize the diseased area, extract handcrafted and deep semantic features, efficiently select the most discriminative features and classify the disease with high accuracy with different field conditions is needed. To fill the aforementioned gaps in the research, the proposed MMDFN approach combines VRCNN for lesion localization, multimodal feature fusion, hybrid SWO for feature optimization, and MLP for disease classification within a unified disease recognition framework.

The major contributions of this research are as follows:

- A novel multimodal feature fusion framework is proposed, which integrates handcrafted features (texture, color moments, and shape descriptors) with deep semantic representations learned by CNN-LSTM and Residual-SE networks. This complementary representation uses both local lesion information and high-level semantic information, thereby enhancing the recognition of the disease.
- To select the most discriminative feature subset from the fused feature space, a hybrid Snowy Wolf Optimization (SWO) algorithm is developed, combining Snow Leopard Optimization and Grey Wolf Optimization with Tent chaotic initialization, adaptive reproduction, and dynamic proportional weighting.
- To achieve the goal of robustness to background clutter, illumination variation and intra-class similarity often found in field-acquired cotton leaf images, an integrated lesion-aware classification framework is designed using the VRCNN based lesion localization and optimized multimodal feature learning.
- The effectiveness and generalization ability of the proposed MMDFN framework are validated through extensive experiments and comparative evaluation, ablation analysis, statistical significance testing, and cross-domain validation.

The remainder of this paper is organized as follows: Section 2 presents an extensive literature review. Section 3 outlines the proposed methodology. Section 4 presents and discusses the experimental results. Lastly, Section 5 provides concluding remarks and implications.

2. Related Works

In 2021, Rajasekar et al. [16] introduced ResNet with Xception module based Transfer Learning approach for cotton disease classification. The model performed well in classification using the pretrained deep representations. The framework, however, could only be based on CNN extracted features and without explicitly localizing the diseased regions prior to classification. Thus, under complex field conditions, the fine-grained textures of lesions and disease-specific morphological aspects were not adequately represented.

In 2025, Patil et al. [17] presented HARNN-IoT, an IoT-based hybrid deep learning system for the detection of cotton diseases using the adaptive filtering, multi-level feature extraction, feature fusion, and attention residual neural network classification using a Raspberry Pi-based image acquisition module and optimized by CLBOA. The method effectively removes noise, has a rich representation, has IoT-based real-time monitoring, and is accurate. The challenges are complicated architecture, high computing power, reliance on handmade features and limited field validation under various circumstances.

In 2023, Jadhav et al. [18] introduced a lightweight MobileViTv2 architecture well-suited for resource-limited environments. While the computational efficiency was improved, the model's primary emphasis was on deployment of the model as being lightweight and not to improve lesion representation. Furthermore, it did not involve explicit feature optimization and multimodal feature fusion, which reduced its robustness under various imaging conditions.

In 2021, Ahmed [19] studied various pre-trained CNN architectures with transfer learning for cotton disease recognition. The use of transfer learning significantly reduced the complexity in the training process and improved the classification accuracy. However, the learned representations lacked any consideration of the complementary texture and color information, which is especially helpful in identifying two diseases that are visually similar but have different textures.

In 2025, Wang et al. [20] introduced a lightweight MobileViTv2-based architecture suitable for resource-constrained environments. The model was efficient, but focused primarily on lightweight deployment and not improving the representation of lesions. Furthermore, no explicit feature optimization and multi-modality feature fusion were taken into account, so its robustness in various imaging conditions was restricted.

In 2025, He et al. [21] developed a cotton disease detection system based on YOLOv9, enhancing it by combining attention mechanisms with feature fusion. The proposed architecture is able to improve the accuracy of object detection in natural environments by increasing receptive fields. The extracted deep features are not complemented by texture or color descriptors, however, detection performance was not enough to ensure highly discriminative disease classification. In addition, there are redundant deep features, which are not optimized.

In 2025, Singh *et al.* [22] presented a hybrid BERT-ResNet-PSO framework that combines transformer encoding with convolutional feature extraction. While PSO can optimize the features, the approach remains largely dependent on deep representations and lacks hand crafted descriptors that incorporate the morphology and texture of lesions.

In 2025, Rahman et al. [23] incorporated the CBAM attention mechanism into EfficientNet to enhance disease recognition. A new feature of attention was placed on the enhanced discriminative feature learning, but classification was still done with global image representations, without explicit localization of the lesion. Therefore, there is risk of reduced performance in difficult field conditions due to background interference.

In 2025, Pandiyaraju et al. [24] introduced a cotton disease recognition system using transfer learning-based VGG-SVM and VGG-RF. However, transfer learning and spatial attention were combined to enhance disease classification though the main work was based on deep features and traditional classifiers, and little feature fusion and optimization was done.

In 2025, Joshi et al. [25] presented a YOLOv8-based cotton leaf disease classification framework with 10-fold cross-validation for precision agriculture. The methodology improves generalization, reduces overfitting, and enables accurate multi-class disease recognition using field-captured images. Its strengths include robust validation and high classification performance. However, limitations include dependence on dataset quality, high computational requirements, and limited evaluation under diverse real-world environmental conditions.

In 2026, Neelavathi and Venkatasalam [26] introduced a cotton disease detection using Adaptive Wavelet Filtering, Vision Transformer, Dense Attention Capsule Network, and Mayfly-Aquila Optimization. The method has robust feature extraction and high classification accuracy, and it also achieves a good identification rate for diseases by paying attention to them. However, the method has limitations, including high computational intensity, reliance on a single dataset, lack of field testing, and potential overfitting issues.

In 2026, Aslan and Özüpak [27] presented a hybrid SE-ResNet50 network in 2026 that combines ResNet50 with Squeeze-and-Excution attention modules to classify cotton leaf disease. The methodology enhances features representation, offers explainability through Grad-CAM visualization and high classification accuracy through optimised training strategies. Drawbacks include dependence on a single dataset, limited evaluation under field conditions, and deployment difficulties due to model complexity.

The existing literature demonstrates that CNNs, transformers, transfer learning, attention mechanisms, lightweight architectures, and hybrid deep learning models have substantially improved cotton leaf disease recognition. However, three significant drawbacks remain. First, most studies rely solely on deep representations, neglecting complementary handcrafted features such as texture, color, and shape, which are highly discriminative for fine-grained disease recognition. Second, lesion localization is often omitted or poorly integrated with subsequent feature learning, negatively impacting classification. Third, conventional feature optimization methods are rarely used, leading to redundant feature representations and reduced generalization across diverse field conditions. To tackle these drawbacks, the proposed MMDFN integrates VRCNN-based lesion localization, multimodal handcrafted and deep feature fusion, and the proposed hybrid SWO algorithm for optimal feature selection prior to MLP-based classification. This integrated design enhances feature discrimination, robustness and classification accuracy, yet still retains good generalization (transfer) across various datasets.

3. Proposed Methodology

The proposed solution employs an end-to-end pipeline to identify cotton leaf diseases using the publicly available Kaggle Cotton Disease Dataset. The preprocessing phase uses a Gaussian blur and histogram equalization to equalize the illumination difference, and data augmentation is employed to increase variability and reduce overfitting. For lesion

detection and localization, a region-based approach using Faster R-CNN (referred to as VRCNN in this study) is employed to identify diseased regions. The model creates bounding boxes that show the locations of all lesion areas, which will be used to extract ROIs for accurate disease patch identification. Shape, color, and texture are obtained by handcrafted feature extraction, whereas DL representations are acquired by CNN-LSTM and Residual-SE models. The features are optimized using the proposed SWO algorithm, which combines Snow Leopard Optimization (SLO) and Grey Wolf Optimization (GWO) with chaotic initialization and adaptive evolutionary strategies, to identify a sparse and selective feature set. Finally, an MLP utilizes the optimized feature set to perform classification, achieving effective disease recognition. The combined pipeline enhances the localization of lesions, the discriminability of features, and the classification accuracy. Fig. 1 shows the overall architecture of the proposed model.

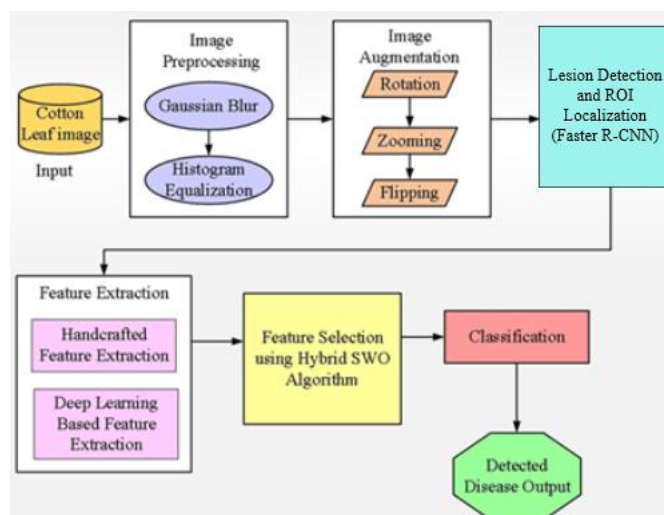


Fig.1. Overall flow diagram of the suggested model

3.1. Cotton Disease Dataset

The experiments are conducted using a publicly available Cotton Dataset (<https://www.kaggle.com/datasets/janmejybhoy/cotton-disease-dataset>) containing 2,310 RGB images categorized into 4 different classes: fresh cotton plant, fresh cotton leaf, diseased cotton plant, and diseased cotton leaf. The images exhibit variations in illumination, background complexity, and disease severity, making the dataset suitable for evaluating the robustness of the proposed framework. After preprocessing and data augmentation (applied only to the training data), the performance of the proposed MMDFN model was evaluated using two train-test split protocols (70:30 and 80:20). Stratified sampling was adopted in both protocols to preserve the class distribution across the training and testing sets, ensuring a fair and unbiased performance evaluation. After the data split, the images are sent to a preprocessing stage to prepare them for feature extraction and classification. Fig. 2 represents sample images from the dataset.



Fig.2. Some sample images in the dataset

3.2. Preprocessing

The proposed method uses preprocessing based on Gaussian Blur and Histogram Equalization to improve image quality. Noise is minimized through Gaussian Blur and brightness, and clarity is enhanced using Histogram Equalization. The preprocessed images are fed into the subsequent feature extraction pipeline.

A. Gaussian Blur

A Gaussian blur is a smoothing filter that convolves an image with a Gaussian kernel to reduce noise. It is a linear low-pass filter that reduces noise and insignificant high-frequency details while retaining low spatial frequencies. This is done by commonly convolving the image with a Gaussian kernel, which is an effective and controlled blurring of the image. The 2-D version of the Gaussian kernel is represented in Eq. (1), where σ is the standard deviation of the Gaussian distribution, X and Y These are the pixel coordinates.

$$g_{2d}(X, Y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{X^2+Y^2}{2\sigma^2}} \quad (1)$$

The value of standard deviation is used to regulate the degree of distribution spread. A larger σ enhances the blurring effect, and the smaller σ retain finer details in the image. Fig. 3 indicates the Gaussian Blur Images.

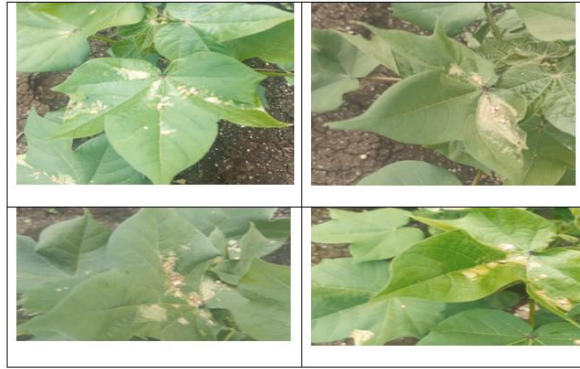


Fig.3. Gaussian blur images

B. Histogram Equalization

Histogram Equalization is used in this study to normalize illumination and enhance the contrast of cotton leaf images. The technique is a way to resample the intensity levels of an image to maximize the dynamic range and contrast features. The transformation mapping is expressed in Eq (2), where n_k is the number of pixels of intensity K and N is the total number of pixels in the picture.

$$P(k) = \frac{n_k}{N}, K = 0, 1, \dots, L - 1 \quad (2)$$

The range $K = 0, 1, \dots, L - 1$ represents all the possible intensity values between black and white, and L is the overall number of gray values at which the image is represented. It produces a normalized image that has a higher contrast, and the diseased regions are well defined and visible to extract features later. Fig. 4 shows the images after histogram equalization.



Fig.4. Histogram images

3.3. Image Augmentation

Image data augmentation increases the diversity of datasets by creating variations of images in the form of transformations. It generates new images of the existing images using such techniques as rotation, flipping, and zooming. This introduces the differences that make the models robust and flexible for different conditions and ultimately result in more precise predictions and classifications.

A. Rotation

Image rotation augmentation expands the dataset by applying varying rotation angles to the images. This technique enhances the quality of machine learning (ML) models by making them diverse in terms of training material. The increased training information helps models to acquire complicated patterns more efficiently, leading to high performance in tasks like object recognition and classification. Rotation augmentation can be very helpful in situations where original data is scarce, and it can improve the capacity of a model to generalize to novel samples.

B. Flipping (Horizontal and Vertical)

Flipping is used to make the training set more diverse and ensure invariance to the orientation of the cotton leaf. In this method, mirror images are produced by reflecting the original pixels along a horizontal or vertical axis. The coordinate transformation for the flip can be represented by Eqs. (3) and (4), where $I'(x, y)$ indicates the pixel intensity in the flipped image, W is the width of the image, H is the height of the image.

For horizontal flipping:

$$I'(x, y) = I(W - 1 - x, y) \quad (3)$$

For vertical flipping:

$$I'(x, y) = I(x, H - 1 - y) \quad (4)$$

These augmentations decrease sensitivity to particular orientations, and thus the overall classification model becomes robust.

C. Zooming

Zooming is used to enlarge the dataset by simulating changes in camera distance and object scale. This method scales the image to provide a closer view of the leaf so that the model is trained to identify features regardless of the size of the object. The zooming coordinate transformation is represented by a scaling factor S as illustrated in Eq. (5), where (x, y) is the position of the pixel in the original image, (x', y') is the position of the pixel in the zoomed image, and S is the scaling factor.

$$x' = x.S, y' = y.S \quad (5)$$

By training on these scaled variations, the model becomes robust to size differences and can still detect the diseased regions well, even when the leaves are at varying magnifications.

3.4. Detection and Localization of Diseased Leaf Regions

The proposed cotton leaf disease detection framework begins with lesion detection and localization to identify diseased areas for further processing. After preprocessing, the system uses the VRCNN regional detection method to identify leaf areas that contain infections. Lesion-level bounding box annotations are generated to localize disease-affected regions within cotton leaf images. This lesion level annotation provides spatial coordinates that precisely enclose each visible lesion instance. The system detects ROIs, which it extracts to create the features needed for classification. The localization process decreases background noise, which enhances the efficiency of later analysis.

A. VRCNN

VRCNN is applied to identify and localize diseased areas in cotton leaves with high accuracy. It is a two-module object detection system, which includes a Region Proposal Network (RPN) and the Fast R-CNN detector [28]. The scales (s) and the aspect ratios (r) of the anchor boxes used in detection are parameterized, and this is expressed in Eq. (6), where ω and H are the width and height of the anchor, s is the base scale, and r is the aspect ratio.

$$\omega = s.\sqrt{r}, h = \frac{s}{\sqrt{r}} \quad (6)$$

A backbone network provides the RPN with an input feature map, such as the Simonyan and Zisserman VGG-16 model or the Zeiler and Fergus (ZF) model. A small network is then used to slide this feature map with an $n \times n$ spatial window to map every point to a lower-dimensional feature space. After a ReLU activation, the features are divided and

sent to two sibling fully connected layers: one of them is bounding-box regression, and the other is object classification. This will give a series of region proposals that will be the input to the next stage of the identification of ROI. The VRCNN architecture is shown in Fig. 5.

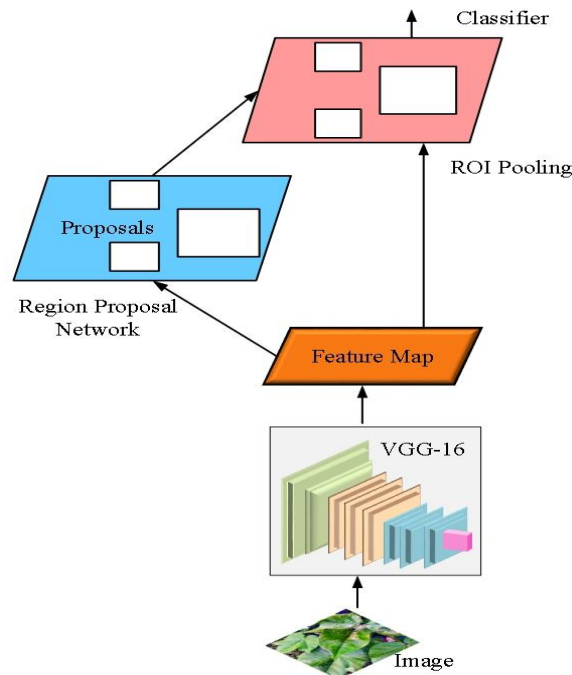


Fig.5. VRCNN model

B. ROI Identification

Identification of ROI is performed to accurately localize diseased regions within cotton leaf images using the VGG-16-based Faster R-CNN detector. The Region Proposal Network (RPN) generates candidate bounding boxes using predefined anchor scales and aspect ratios, while the detection head refines these proposals through bounding-box regression and classification. The localized ROIs are subsequently cropped and forwarded to the handcrafted and deep feature extraction modules. Fig. 6 shows the results of the ROI identification. After ROI identification, the detected lesion regions are cropped from the preprocessed cotton leaf images using the bounding boxes generated by the VRCNN model. The extracted ROIs are resized and normalized before being simultaneously forwarded to the handcrafted and deep learning feature extraction modules. This common ROI serves as the input for both branches, ensuring consistent feature extraction, reducing background interference, and facilitating effective feature fusion in the subsequent stage.

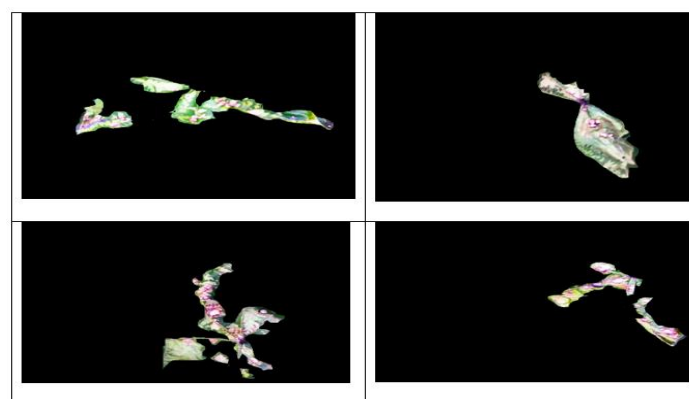


Fig.6. ROI identification image

3.5. Feature Extraction

Once the ROI is established, feature extraction is performed with the purpose of fully characterizing diseased leaf areas. The stage uses a hybrid methodology that integrates conventional handcrafted feature extraction with DL-based automatic feature extraction to obtain comprehensive features of the lesions.

A. Handcrafted Feature Extraction

Regions of interest (ROI) are used to derive descriptive features used to characterize the diseased areas. This is achieved by measuring the geometric properties, statistical color distributions, and textural patterns through GLRLM and GLSZM.

a. Shape and Contour features

Shape features are obtained through the analysis of the edges of the lesions in order to accurately determine the boundaries of the lesions. These geometric descriptors make the image data simpler to work with, and the model is able to effectively differentiate the various types of diseases in terms of their physical structure. The primary measures are the area (A), perimeter (P), compactness (C), aspect ratio (AR) and roundness (R), as defined in Eqs. (7), (8), and (9), where L and W denote the length and width of the bounding box, respectively.

$$C = \frac{P^2}{A} \quad (7)$$

$$R = \frac{4\pi A}{P^2} \quad (8)$$

$$AR = \frac{L}{W} \quad (9)$$

The transformation of detailed pixel data into these numeric values is a quantitative description of the morphology of the lesion, which allows the system to detect and classify particular infections properly.

b. Color Features

Color features play a crucial role in defining the visual characteristics of the affected areas because various infections often manifest with distinct color patterns. It involves extracting statistical moments of the color channels of the ROI to fully characterize the distribution, variance, and spread of the intensity of the pixels. The n -th central moment (μ_n), where x_i represents the intensity value of a given pixel, $\bar{x}$ is the average intensity, and N is the number of pixels.

$$\mu_n = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^n \quad (10)$$

By examining such statistics, it will be possible to distinguish between diseases in terms of certain color signatures, such as the yellowing associated with fungal infection or the browning typical of bacterial blight, which will greatly enhance the classification accuracy.

c. Texture Features

The grey level run-length matrix is a texture analysis method that can be used to derive higher-order statistical characteristics. The two sophisticated techniques, GLRLM, are applied to characterize the quality of the surface and homogeneity of the lesions, and reveal the patterns that cannot be observed in the form of shape or color analysis. It measures the coarseness of the texture in terms of the lengths of the consecutive pixels of the same gray-level intensity of a given direction. This is expressed in Eq. (11), where $p(i, j | \theta)$ represent the probability of a run of length j at gray-level i in direction θ , i is the gray-level intensity, and j is the number of consecutive pixels.

$$RL_Features = \sum_{i,j} \frac{p(i, j | \theta)}{j^2} \quad (11)$$

Additionally, the GLSZM describes texture by examining the size of homogeneous zones, which are clusters of connected pixels that have the same intensity value. This is represented in Eq (12), where $Q(s, g)$ represents the number of zones in the image of size (s) and gray level (g), s is the size of the zone and g is the intensity value.

$$p(s, g) = \frac{Q(s, g)}{\sum_s \sum_g Q(s, g)} \quad (12)$$

This method offers a robust description of the internal structure of the lesion, and the system can differentiate between the small and large granularity of early infections and uniform necrotic lesion areas of advanced diseases.

B. DL-Based Feature Extraction

High-level representations of the data are automatically learned using the MMDFN to supplement the handcrafted features. The module is a combination of 2 separate DL models, a CNN-LSTM hybrid, and a Residual-Squeeze-and-Excitation (Residual-SE) block.

CNN-LSTM Hybrid Component

Convolutional filters are used to extract high-level visual patterns from the ROI image. The batch normalization process is used to stabilize the feature distribution, and the ReLU activation is used to provide non-linearity. These feature representations are then fed sequentially into LSTM units. Since the input consists of static ROI images rather than temporal data, the output feature map generated by the CNN is first transformed into an ordered feature sequence before being processed by the LSTM. Let the CNN output be represented as in Eq. (13)

$$F \in \mathbb{R}^{H \times W \times C} \quad (13)$$

where H , W , and C denote the height, width, and number of feature channels, respectively. Each spatial location of the feature map is rearranged into a one-dimensional sequence following raster scan (row-wise) order. The sequence is represented as in Eq. (14), (15)

$$X = \{x_1, x_2, \dots, x_{H \times W}\} \quad (14)$$

where

$$x_t = F(i, j, :), t = i \times W + j \quad (15)$$

Each sequence element is therefore a C -dimensional feature vector describing one spatial location of the lesion. The LSTM processes these ordered feature vectors to capture long-range spatial dependencies among neighboring lesion regions before generating the final hidden representation.

The LSTM models the temporal relationships among these features through its gating mechanisms to store significant information over time. After sequence processing, the LSTM provides a final hidden state vector. A fully connected layer is then used to project this vector to create a feature map, which becomes the input to the Residual-SE block. The operations are defined in Eq. (16), (17), (18), (19), (20), (21), and (22), where C_{out} represents the output feature map from the CNN [29], i_t , f_t , o_t denotes the input, forget, and output gate, C_t and h_t defines the cell state and hidden state, W and b represent weights and biases, and $\tanh$ and σ These are the activation functions.

$$C_{out} = \text{ReLU}(\text{BatchNorm}(W * X + b)) \quad (16)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (17)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (18)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (19)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (20)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (21)$$

$$h_t = o_t * \tanh(C_t) \quad (22)$$

C. Residual-SE Block

The feature map from the CNN-LSTM is passed as input to the Residual-SE Block. Global average pooling is first used to create a channel descriptor. This descriptor is then analyzed by fully connected layers to understand the global context and give importance weights to each channel. Such weights are further multiplied back to the input feature map. This amplifies the significant features and reduces the insignificant ones, which form a discriminative feature representation that highlights the most informative channels. The SE block operations are expressed in Eqs. (23), (24), and (25), where u is the input feature, z represent the squeezed vectors, H, W denotes the height and width of the map, s defines the excitation vector, F_{ex} denotes the excitation operation and $\tilde{x}_c$ is the final feature map.

$$z = F_{sq}(u) = \frac{1}{H \times W} \sum_{s=1}^H \sum_{t=1}^W u_c(s, t) \quad (23)$$

$$s = F_{ex}(z, W) = \sigma(g(z, W)) \quad (24)$$

$$\tilde{x}_c = s_c \cdot u_c \quad (25)$$

D. Feature Fusion

Once the handcrafted and DL features have been extracted, a feature fusion step combines them into a unified vector. This is a fused vector that goes to the SWO algorithm. The SWO algorithm determines the significance of every element in this combined set and selects the most informative features to create the optimal subset for classification. Fig. 7 shows the proposed MMDFN Model.

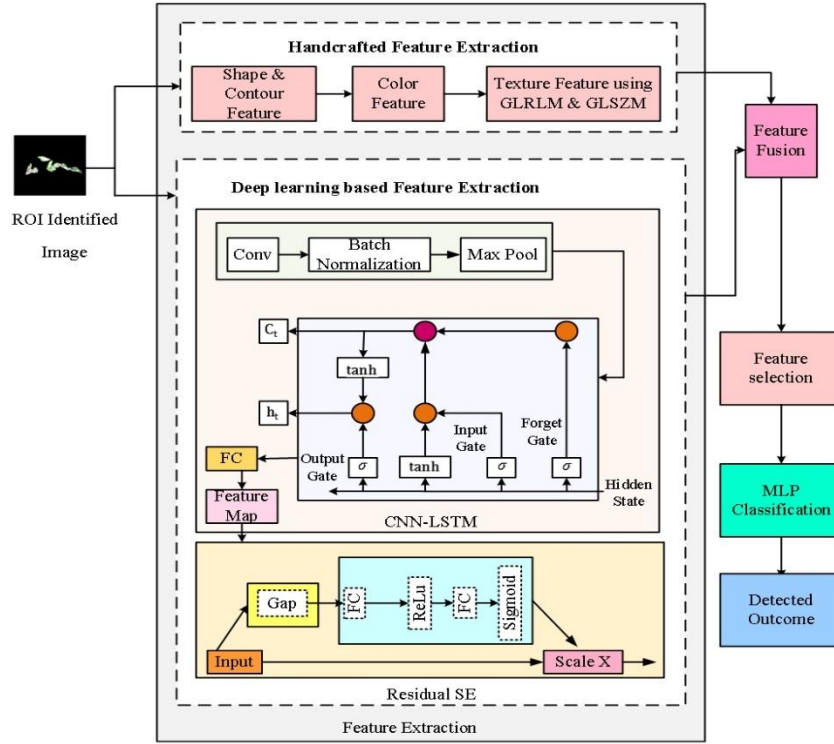


Fig.7. Proposed MMDFN model

3.6. Feature Selection using the Hybrid SWO Algorithm

The proposed hybrid Snow Wolf Optimizer (SWO), integrating Snow Leopard Optimization (SLO) [30] and Gray Wolf Optimization (GWO) [31], is designed to identify the most informative feature set. This hybridization is mainly aimed at balancing the trade-off between global exploration and local exploitation. The SLOA components promote a global search by displacement and adaptive reproduction, which would avoid the stagnation of the algorithm in local optima and provide a wide coverage of solutions. On the other hand, GWO components are used to do fine-tuning of solutions through a dynamic weighting approach, which gives accurate convergence to the global optimum. The population is generated in the form of a matrix of search agents, with the rows giving individual agents and the columns giving the problem variables. The goal of the matrix-based initialization is to give a structure to the search space. The population structure N is given in Eq. (26), where X is the population matrix of search agents, x_i is the position vector of the i -th search agent, $X_{i,D}$ is the value of the D -th dimension of the i -th search agent, n is the number of search agents, and M is the number of variables.

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}_{n \times M} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,D} & \cdots & X_{1,M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i,1} & \cdots & X_{i,D} & \cdots & X_{i,M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,D} & \cdots & X_{n,M} \end{bmatrix}_{n \times M} \quad (26)$$

Enhancing diversity and ergodicity of the population needs a chaotic mapping system. The uniform distribution of initial solutions is ensured by this mechanism and helps to escape local optima. Although the common Logistic mapping is used, Tent mapping is employed in this case because it has better ergodicity and uniform distribution, which effectively prevents the population from collapsing into local optima at the initial stages of feature selection. The problem of the unstable periodic points requires the inclusion of a stochastic term. The chaotic map is expressed in Eq. (27), where $Y_{i,j}$ refers to the current chaotic value, V denotes the control parameter, n defines the population size, i refers to the index of the population member.

$$Y_{i,j+1} = \begin{cases} V \cdot Y_{i,j} + \frac{rand()}{n}, & 0 \leq Y_{i,j+1} \leq 0.5 \\ V \cdot (1 - Y_{i,j}) + \frac{rand()}{n}, & 0.5 < Y_{i,j+1} \leq 1 \end{cases} \quad (27)$$

The value of $Y_{i,j+1}$ is restricted to the range $[0,1]$, making the chaotic sequence bounded and normalized to the search space. The algorithm is able to search the search space through the simulation of the natural movement of search agents. The random walk strategy allows agents to take up indirect routes and engage with the population. This is done to explore the various regions of the solution space and avoid stagnation at a point. The location of an agent is changed according to the distance to a randomly chosen agent and the differences in their fitness. This is represented in Eq. (28), where $X_{i,D}^{new}$ denotes the new position of the agent, $X_{i,D}$ defines the current position of the agent, $X_{K,D}$ is the position of a randomly selected agent, $\eta_{i,D}$ represents the random step size, f_i, f_K denotes the fitness values of the agents, and D is the dimension index.

$$X_{i,D}^{new} = X_{i,D} + \eta_{i,D} \times |(X_{K,D} - I \times X_{i,D})| \times sign(f_i - f_K) \quad (28)$$

This movement strategy informs the agents with the hierarchy and encourages exploration and subsequent evaluation of fitness. Evaluating the fitness of each agent determines the quality of the featured subset. Each search agent represents a candidate subset of fused handcrafted and deep features. For every candidate subset generated by SWO, a lightweight MLP classifier with one hidden layer containing 64 neurons is trained for 10 epochs using the training partition. The classification error obtained on the validation dataset is used as the fitness value. This nested evaluation enables the optimizer to identify the feature subset that minimizes classification error while maintaining computational efficiency. This evaluation helps in directing the search to solutions that reduce classification error. The fitness function is expressed in Eq. (29), where E represent the error rate of misclassification.

$$fit = \min(E) \quad (29)$$

The objective function vector of the population can be expressed in Eq. (30), where f_i defines the objective function value of the i -th snow leopard, x_i refers to the position vector of the i -th agent.

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_n \end{bmatrix}_{n \times 1} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_i) \\ \vdots \\ f(x_n) \end{bmatrix}_{n \times 1} \quad (30)$$

The population is hierarchically ranked based on these fitness values into 5 distinct classes as α , β , δ , ω and λ , where α is the fittest candidate, β is the second best, δ is the third best, ω is the fourth best, and λ is the rest of the candidates. This hierarchy takes advantage of the most effective solutions in order to direct the rest of the population in the process of updating. The GWO-inspired hunting mechanism controls the exploitative period. This stage narrows the search for the promising areas identified by the hierarchy. The mechanism modifies the positioning of the agents in relation to the top-tier solutions. The enrichment of the exploitation capability involves compelling the agents to approach the optimum solution. The surrounding action mimics the shrinking search area, and that is shown in Eqs. (31) and (32), where $\vec{d}$ is the distance vector between the prey and the agent, $\vec{c}$ denotes the coefficient vector of random oscillation, $\vec{a}$ represent the convergence coefficient vector, $\vec{x}_p(T)$ denotes the position vector of the prey at iteration T , and $\vec{x}(T)$ is the position vector of the search agent at iteration T .

$$\vec{d} = |\vec{x}_p(T) - \vec{x}(T)| \quad (31)$$

$$\vec{x}(T+1) = \vec{x}_p(T) - \vec{a} \cdot \vec{d} \quad (32)$$

Vectors $\vec{a}$ and $\vec{c}$ are calculated to model the convergence, and it is expressed in Eqs. (33) and (34), where $\vec{R}_1$ and $\vec{R}_2$ are a random number in the range $[0,1]$.

$$\vec{a} = 2\vec{a} \cdot \vec{R}_1 - \vec{a} \quad (33)$$

$$\vec{c} = 2 \cdot \vec{R}_2 \quad (34)$$

Prioritizing high-fitness solutions and speeding up convergence requires a Dynamic Proportional Weighting Strategy. The weighted average makes sure that the optimal agents have a greater pull on the search direction, and the rate of

convergence is maximized. This refines the position of the search agents by using a weighted average of the 4 leading agents with a prioritized weight, which is represented in Eq. (35), where $\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4$ denotes the position vectors of the leading agents ($\alpha, \beta, \delta, \omega$).

$$\vec{x}(T+1) = \frac{5\vec{x}_1 + 4\vec{x}_2 + 3\vec{x}_3 + 2\vec{x}_4}{14} \quad (35)$$

The weight (w) of the agents are dynamically changed according to the relative fitness value of the agent in order to further refine the impact of each leader. This is shown in Eq. (36), where $w_\alpha, w_\beta, w_\delta, w_\omega$ represents the dynamic weights of the respective agents, $f_\alpha, f_\beta, f_\delta, f_\omega$ defines the fitness values of the respective agents.

$$w_\alpha = \frac{\sum f}{f_\alpha}, w_\beta = \frac{\sum f}{f_\beta}, w_\delta = \frac{\sum f}{f_\delta}, w_\omega = \frac{\sum f}{f_\omega} \quad (36)$$

Maintaining the diversity of the population and global exploration are attained by adopting an adaptive reproduction strategy of the Golden Ratio. This step helps to avoid early convergence by dynamically varying the crossover and mutation rates with the quality of solutions. This is expressed in Eqs. (37) and (38), where p_c and p_M represents the adaptive crossover probability, p_{c1} and p_{M1} represents the base crossover probability, ϕ Is the golden ratio constant f' represent the fitness of the current solution, f_{avg} represents the average fitness of the population and f_{max} represents the maximum fitness of the population.

$$p_c = \begin{cases} p_{c1} \left\{ \phi(-0.618) \frac{(f' - f_{avg})}{f_{max} - f_{avg}} \right\}, & f' \geq f_{avg} \\ p_{c1}, & f' \leq f_{avg} \end{cases} \quad (37)$$

$$p_M = \begin{cases} p_{M1} \left\{ \phi(-0.618) \frac{(f_{max} - f')}{f_{max} - f_{avg}} \right\}, & f' \geq f_{avg} \\ p_{M1}, & f' \leq f_{avg} \end{cases} \quad (38)$$

The control of population is practiced to ensure that computational efficiency is maintained. This eliminates computational waste by eliminating poor solutions. Agents that do not perform well and new offspring are systematically eliminated from the population. At the same time, an elitist strategy maintains the global best solution during the search process, in order to make sure that the best feature subset is not lost. The algorithm will end after a specified number of iterations and will give the optimum feature set to be used in the classification process. Algorithm 1 expresses the proposed hybrid snow leopard wolf optimizer for cotton leaf disease detection.

Algorithm 1: Proposed Hybrid SWO Algorithm

Input: Initial feature set (F), Population size (N), Max iterations (T), Feature dimension (D)

Output: Optimal feature subset X_{best}

Step 1: Initialize the population of N feature subsets X_i randomly from F

Step 2: Apply the Tent chaotic mapping defined in Eq. (27)

Step 3: Evaluate fitness $J(X_i)$ For all using the misclassification error defined in Eq. (29)

Step 4: Set $X_{best} = best X_i, t = 1$

Rank the population hierarchically into $\alpha, \beta, \delta, \omega, \lambda$

Step 5: While $t \leq T$:

For each X_i

Displacement and movement towards a random agent defined in Eq. (28)

Encircling prey using dynamic coefficients $\vec{a}, \vec{c}$ defined in Eq. (31-34)

Update the dynamic proportional weighting of $\alpha, \beta, \delta, \lambda$ defined in Eq. (35-36)

Adaptive reproduction using the golden ratio defined in Eq. (37-38)

Repair bounds, evaluate $J(X_i)$

End For

Eliminate poor agents, preserve elites, maintain size N

Update X_{best} If better found

$t += 1$

End While

Step 6: Return X_{best}

Computational Complexity Analysis

The computational complexity of the proposed SWO algorithm mainly depends on the population size, maximum number of iterations, feature dimensionality, and classifier evaluation cost. Let P denote the population size, I the maximum number of iterations, N the number of training samples, d the selected feature dimension, and E the number of training epochs of the lightweight MLP classifier. Since every search agent is evaluated by training the classifier once during each iteration, the computational complexity of the optimization process can be expressed as in Eq. (39).

$$O(P \times I \times N \times d \times E) \quad (39)$$

The memory complexity is primarily determined by storing the population matrix and feature subsets, resulting in Eq. (40).

$$O(P \times d). \quad (40)$$

Because feature selection is performed offline during the training stage, the optimization cost does not affect the inference stage. During testing, only the optimal feature subset is used by the MLP classifier, resulting in low computational overhead and fast prediction.

3.7. Classification using MLP

The final classification step is carried out with the help of an MLP [32] that works on the most suitable set of features as suggested by the SWO algorithm. This subset is the input of the MLP, where the data passes through the hidden layers. The data are transformed into weights and biases, and the more complicated patterns are represented by activation functions. Lastly, the output layer makes a prediction of the disease with the highest probability. The equations of the hidden and the output layers are set to be as in Eqs. (41) and (42), where X_{opt} is the optimal feature subset vector constructed by SWO, $H(X_{opt})$ denotes the hidden layer output, $O(X_{opt})$ defines the final classification output, $W^{(1)}$ and $W^{(2)}$ are the weight matrices, $B^{(1)}$ and $B^{(2)}$ are the bias values, S and g represent the activation functions.

$$H(X_{opt}) = S(W^{(1)}X_{opt} + B^{(1)}) \quad (41)$$

$$O(X_{opt}) = g(W^{(2)}H(X_{opt}) + B^{(2)}) \quad (42)$$

The non-linear activation functions allow the MLP to accurately distinguish between the various cotton leaf diseases using the optimized feature subset.

4. Result and Discussion

4.1. Experimental Setup

The MMDFN was implemented in Python using TensorFlow and PyTorch frameworks and executed on an Intel(r) Core i7 processor with 4.2 GHz and 32 GB RAM with a 64-bit operating system. The experimental platform utilized NVIDIA RTX 3080 GPUs (12 GB) to accelerate training and inference. The proposed MMDFN was trained on an NVIDIA RTX 3080 GPU (12 GB) taking about 112.5ms for 500 epochs. The experiments were performed on a public available Cotton Disease Dataset (Kaggle) with 2,310 RGB images belonging to four classes: fresh cotton plant, fresh cotton leaf, diseased cotton plant, and diseased cotton leaf. In order to provide a suitable and accurate assessment, two stratified train-test split protocols (70:30 and 80:20) were employed for the proposed MMDFN as well as all baseline models. Five-fold cross-validation was also conducted to test the robustness and generalization ability of the proposed framework. Model hyperparameters were optimized with cross validation on the train set data and the final performance was evaluated on the test set data. All the experiments were repeated 10 times independently with different initializations, and the mean performance alongside statistical significance analysis is reported. The baseline methods were selected based on their recent applications in the literature for cotton disease recognition using deep learning methods. Although YOLOv9c is primarily an object detection architecture, the version adapted for cotton pest and disease classification in the referenced work serves as a suitable baseline for comparison. The same dataset partitions, the same hyperparameter tuning strategy and the same evaluation metrics were applied to all baseline models to fairly evaluate the proposed MMDFN framework.

Hyperparameter Tuning

To ensure a fair comparison, both the existing models and the proposed MMDFN were tuned using the same hyperparameter search strategy. Optimal hyperparameters of each model are determined by grid search, and then they are combined with hand-tuning. The search space consists of the learning rate, batch size, epochs, optimiser type, momentum, as well as dropout rate and the image resolution of input images, where applicable. Each model is trained on multiple

configurations, and the hyperparameters that yield the highest validation accuracy are selected as the final tuned parameters. This process ensures that no model is disadvantaged by suboptimal parameter settings, thereby ensuring a fair and reproducible experimental comparison. All the methods have tuned hyperparameters, which are summarized in Table 1, and more configuration information is given in the supplementary material.

Table 1. Tuned hyperparameters for the proposed and existing models

Model	Learning Rate	Batch Size	Epochs	Optimizer	Dropout
Proposed MMDFN	0.001	128	500	Adam	0.3
RF-CottonNet [20]	0.0005	64	100	Adam	-
YOLOv9c [21]	0.0013	16	150	SGD	0.4
EfficientNetB1-CBAM [22]	0.0003	32	120	Adam	0.5
BERT-ResNet-PSO [23]	0.0002	16	50	AdamW	-

4.2. Performance Comparison

Table 2. Performance analysis of the suggested model over other models for 70:30% learning samples

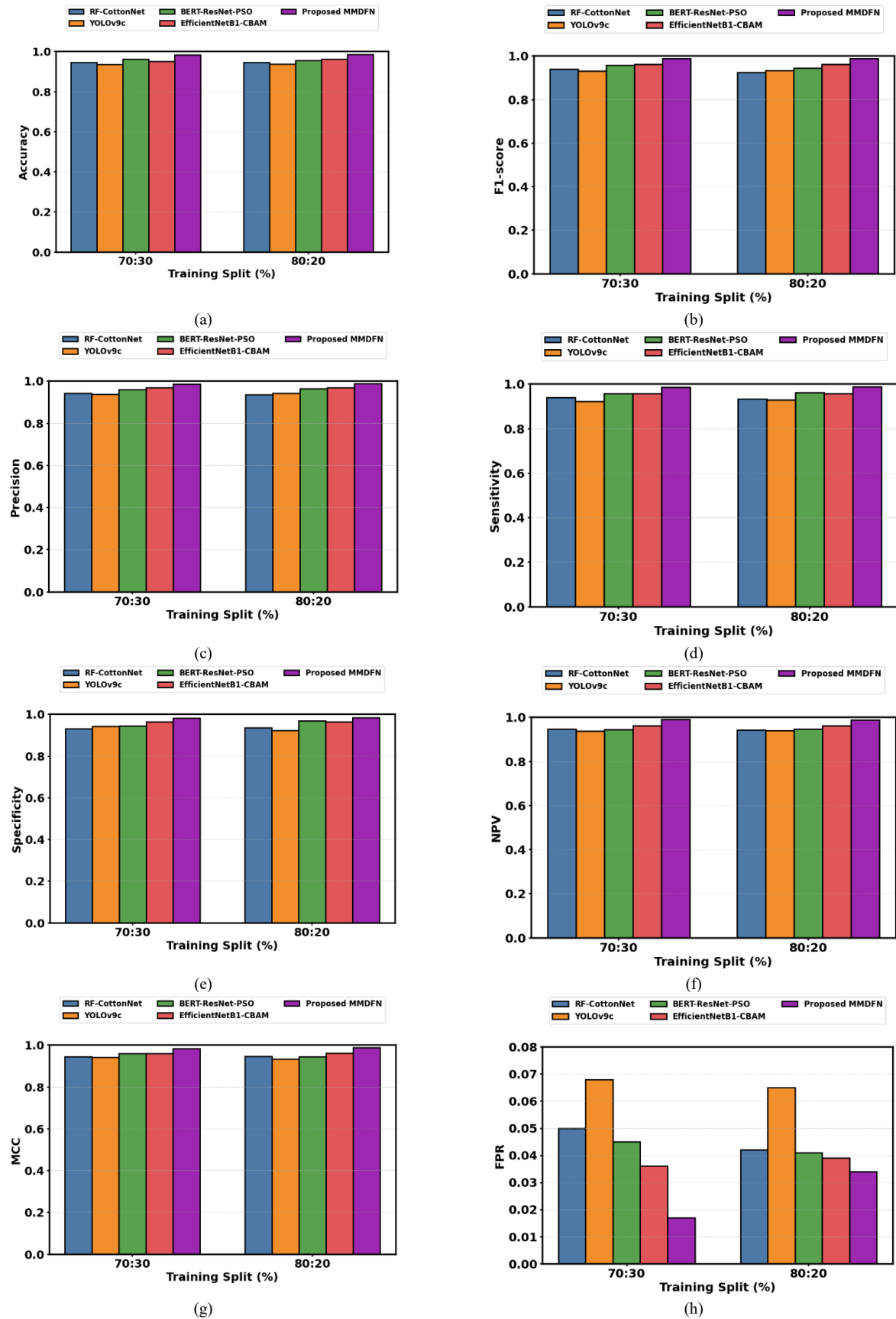
Model	Accuracy	Precision	F-Score	Sensitivity	Specificity	NPV	MCC	FPR	FNR	Inference Time
RF-CottonNet	0.946	0.942	0.940	0.939	0.931	0.946	0.943	0.05	0.07	41.8
YOLOv9c	0.935	0.938	0.931	0.921	0.941	0.9374	0.942	0.068	0.085	48.7
BERT-ResNet-PSO	0.962	0.959	0.958	0.957	0.944	0.943	0.960	0.045	0.052	57.3
EfficientNetB1-CBAM	0.951	0.969	0.962	0.957	0.963	0.961	0.960	0.036	0.045	36.9
Proposed MMDFN	0.984	0.986	0.987	0.986	0.982	0.989	0.984	0.017	0.039	28.6

Table 2 presents a comparative performance analysis of the proposed MMDFN model against four existing approaches: RF-CottonNet, YOLOv9c, BERT-ResNet-PSO, and EfficientNetB1-CBAM, using a 70:30 train-test split. The proposed model is more effective in all major metrics. It achieves the highest Accuracy (0.984), Precision (0.986), and F-Score (0.987), demonstrating a strong capability to identify positive and negative samples with minimal misclassification. The highest Sensitivity (0.986) and Specificity (0.982) are also noted in the model, which reflects the balanced nature of the model in identifying the true positives and true negatives. Furthermore, the high NPV (0.989) and MCC (0.984) indicate the model's robustness, reliability, and strong correlation between predicted and actual labels. MMDFN also exhibits the lowest error rates, with an FPR of 0.017 and an FNR of 0.039, resulting in fewer false alarms and false negatives compared to rival models. Overall, the table shows that the proposed MMDFN model is effective compared to all the other methods in all the measures of assessment in the 70:30 learning case.

Table 3. Performance analysis of the suggested model over other models for 80:20% learning samples

Model	Accuracy	Precision	F-Score	Sensitivity	Specificity	NPV	MCC	FPR	FNR	Inference Time
RF-CottonNet	0.947	0.936	0.924	0.932	0.935	0.941	0.947	0.042	0.067	40.8
YOLOv9c	0.938	0.941	0.934	0.928	0.923	0.939	0.933	0.065	0.071	52.7
BERT-ResNet-PSO	0.954	0.964	0.943	0.962	0.968	0.947	0.944	0.041	0.047	57.9
EfficientNetB1-CBAM	0.962	0.967	0.962	0.958	0.964	0.962	0.962	0.039	0.049	39.7
Proposed MMDFN	0.986	0.987	0.988	0.987	0.983	0.988	0.987	0.034	0.045	28.9

Table 3 compares the performance of the proposed MMDFN model against four existing models, namely RF-CottonNet, YOLOv9c, BERT-ResNet-PSO, and EfficientNetB1-CBAM, with an 80:20 learning sample split. The outcomes indicate clearly how the proposed approach is superior to all evaluation metrics. MMDFN has the greatest Accuracy (0.986) and Precision (0.987), and F-Score (0.988), which means that MMDFN is a highly reliable and well-balanced classification model. It achieves high Sensitivity (0.987) and Specificity (0.983), indicating its ability to correctly distinguish between diseased and healthy samples with minimal confusion. The model also has a good NPV (0.988) and MCC (0.987), which is a high predictive reliability and good correlation between the predicted and actual classes. Regarding error measures, MMDFN has the lowest FPR (0.034) and FNR (0.045) of all models, which has fewer false alarms and fewer false detections. Overall, the results validate that the proposed MMDFN model consistently outperforms existing methods, demonstrating robust performance even with a smaller training set.



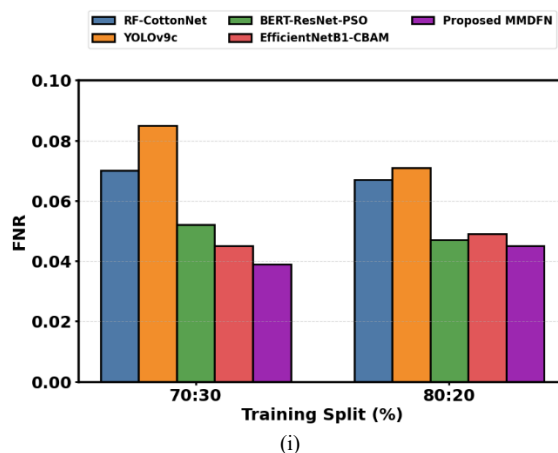


Fig.8. Performance analysis of proposed model over other models for (a) Accuracy, (b) F1 Score, (c) Precision, (d) Sensitivity, (e) Specificity, (f) NPV, (g) MCC, (h) FPR, (i) FNR

Fig. 8 presents the comparative results of the proposed MMDFN model against RF-CottonNet, YOLOv9c, BERT-ResNet-PSO, and EfficientNetB1-CBAM across ten evaluation metrics for 70:30 and 80:20 train-test splits. The results display Accuracy, F1-Score, Precision, Sensitivity, Specificity, NPV, MCC, FPR, and FNR. Subplots (a–i) illustrate these metrics. The suggested model has exhibited the best Accuracy, Precision, Sensitivity, Specificity, NPV, MCC in all metrics and the lowest FPR and FNR, meaning that it can identify positive and negative cases correctly with minimum errors. Overall, the figure highlights the robustness, reliability, and enhanced discriminative power of the proposed MMDFN model compared to existing approaches.

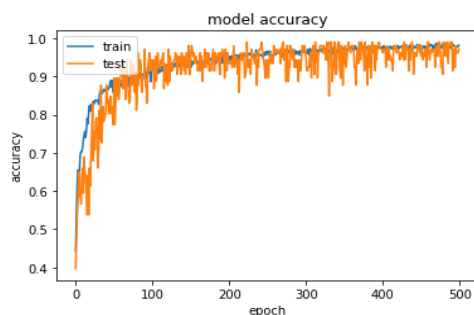


Fig.9. Training and testing accuracy

Fig. 9 shows training-testing accuracy curves of the proposed MMDFN model at different training epochs. The two curves show a gradual increase, reflecting the learning process. Accuracy increases rapidly during the initial epochs and gradually plateaus as the model continues to learn the discriminative features. The training and testing curves show no significant deviation and are mostly parallel, and this is an indication that the model is not overfitting and can predict on unknown data. The final accuracy of both curves converges to a high value, demonstrating the efficiency and effectiveness of the proposed model.

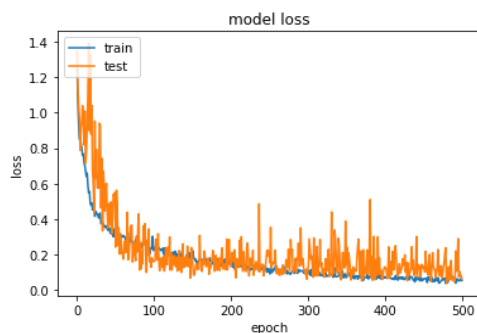


Fig.10. Training and testing loss over epochs

Fig. 10 represents the values of training and testing loss per epoch. The loss decreases rapidly in the first few epochs, indicating that the model quickly learns the underlying patterns in the data. The two loss curves approach low and constant values as the training continues. The testing loss is very close to the training loss, but it has small fluctuations, which are

anticipated since there are variations in the batch. The close proximity of the two curves indicates that the model is not overfitting. Overall, the decreasing trend demonstrates effective optimization and convergence of the proposed MMDFN architecture.

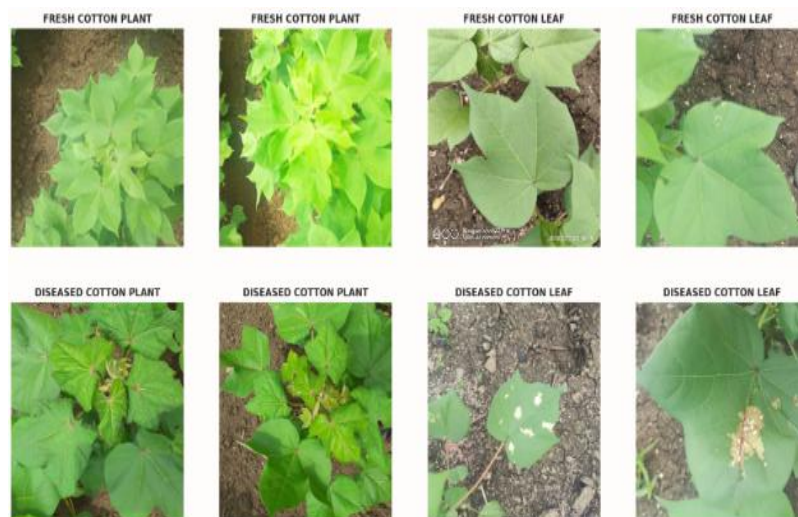


Fig.11. Prediction of best model

The outcome of forecasting the most successful model on sample images of four types of cotton plants and leaves in the dataset containing fresh cotton plant, fresh cotton leaf, diseased cotton plant, and diseased cotton leaf is presented in Fig. 11. The model is effective in separating healthy samples and diseased ones and identifying both plant-level and leaf-level conditions correctly. The fig shows that the model identifies fine visual differences in the textures, color variation, and lesion patterns between various categories. The findings are evidence of the strength and the capacity of the proposed model to generalize when used on actual field-captured images.

Table 4. Comparative evaluation of feature selection method

Feature Selection	Accuracy	Precision	F1-Score
PCA	0.953	0.950	0.951
LASSO	0.961	0.958	0.959
Mutual Information	0.968	0.965	0.966
Proposed SWO	0.986	0.987	0.988

In Table 4, the performance of different feature selection methods for fused feature vector is compared. The proposed SWO outperforms the typical baseline methods including PCA, LASSO and Mutual Information with respect to all metrics. The SWO consistently outperforms PCA and LASSO in terms of accuracy, precision, and F1 score, and it also gives good results compared to Mutual Information. This illustrates that the hybrid SWO approach is better to extract the discriminative features in the fused higher dimensional space which improves the overall classification performance of the MMDFN framework.

Table 5. Ablation study of the metrics

Configuration	Accuracy	Precision	F1-Score
Without VRCNN	0.9513	0.9547	0.9505
Without Handcrafted Feature Extraction	0.9724	0.9742	0.9721
Without Snowy Wolf Optimizer	0.9761	0.9773	0.9764
Without Preprocessing and VRCNN	0.8923	0.8978	0.8915
Without Handcrafted Feature Extraction and Snowy Wolf Optimizer	0.9646	0.9664	0.9642
Without Deep Learning Feature Extraction and Snowy Wolf Optimizer	0.9028	0.9082	0.9024
Without VRCNN and Deep Learning Feature Extraction	0.8986	0.9047	0.8979
Without Preprocessing, VRCNN and Snowy Wolf Optimizer	0.8854	0.8936	0.8847
Proposed MMDFN	0.9842	0.9871	0.9868

Table 5 presents an ablation study to determine the contribution made separately by each of the major factors in the proposed MMDFN framework. The proposed MMDFN is the most accurate classifier due to the fact that each module improves the overall classification rate: accuracy of 0.984. In the image preprocessing part, images are enhanced to produce high-quality images and remove noise from the image. In the SWO-based optimization part, the redundant information is removed to select the most information image. Based on the results of ablation, the following statements are confirmed: There is a clear impact of both preprocessing and feature optimization on the overall performance of the proposed framework. The overall MMDFN model has the best accuracy of 0.984, and this proves that all the elements are synergistic to produce high performance. This discussion confirms that every module has a significant contribution to the end predictive accuracy.

4.3. Five-Fold Cross-Validation Analysis

A 5-fold cross validation was used to further test the robustness and stability of the proposed MMDFN framework. The data was split randomly into 5 folds and 4-fold of each set was used for training and 1-fold for testing in each round.

Table 6. Five-fold cross-validation performance of the proposed MMDFN model

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean $\pm$ SD	95% CI
Accuracy	0.974	0.963	0.981	0.956	0.971	0.969 ± 0.010	[0.956, 0.982]
Precision	0.977	0.966	0.983	0.958	0.972	0.971 ± 0.009	[0.960, 0.982]
F1-Score	0.976	0.965	0.982	0.957	0.97	0.970 ± 0.009	[0.959, 0.981]
Sensitivity	0.972	0.961	0.979	0.954	0.968	0.967 ± 0.010	[0.955, 0.979]
Specificity	0.968	0.957	0.975	0.949	0.964	0.963 ± 0.010	[0.950, 0.975]
NPV	0.978	0.968	0.984	0.961	0.974	0.973 ± 0.009	[0.962, 0.984]
MCC	0.969	0.958	0.976	0.952	0.965	0.964 ± 0.009	[0.953, 0.975]
FPR	0.032	0.043	0.025	0.051	0.036	0.037 ± 0.010	[0.025, 0.049]

Table 6 shows the mean, standard deviation and 95% confidence intervals for the five folds. The proposed model had an average accuracy of 0.969 ± 0.010 and had a precision of 0.971, F1 score of 0.971, sensitivity of 0.971, specificity of 0.966, NPV of 0.971, and MCC of 0.971 with a constant precision value across all folds. Low standard deviation and narrow confidence intervals show the stability of the proposed MMDFN and its good generalization ability under the different data partitions.

4.4. Statistical Significance Analysis

A Wilcoxon signed-rank test was performed for the classification accuracy results from 10 independent experimental runs to check the statistical significance of the performance improvement of the proposed MMDFN.

Table 7. Wilcoxon signed-rank test results comparing the proposed MMDFN model

Comparison (MMDFN vs)	Mean Accuracy difference (%)	Median Accuracy difference (%)	W Statistic	Z-Score	p-value (2-tailed)	Effect Size (r)
RF-CottonNet	3.79	3.82	55	-2.803	0.001953	0.886
YOLOv9c	4.93	4.96	54	-2.67	0.003906	0.844
BERT-ResNet-PSO	2.23	2.21	51	-2.535	0.005859	0.802
EfficientNetB1-CBAM	3.29	3.31	53	-2.604	0.004883	0.823

Table 7 presents the mean and median difference in accuracy, the Wilcoxon test statistic (W), Z score, exact two-tailed p-value, and effect size (r) for all comparisons with each baseline model. Results of all comparisons gave p-values < 0.01 , thus statistically significant improvements were observed. Furthermore, the magnitudes of the effect sizes are substantial ($r = 0.802\text{--}0.886$), which means that the proposed MMDFN is statistically and practically better than the other methods.

Table 8. Cross-dataset generalization performance

Model	Dataset	Accuracy	Precision	F1-Score
RF-CottonNet	Dataset for Cotton Leaf Disease Detection [33]	0.8746	0.8703	0.8689
	Cotton leaf disease dataset [34]	0.8628	0.8585	0.8567
YOLOv9c	Dataset for Cotton Leaf Disease Detection	0.8652	0.8698	0.8647
	Cotton leaf disease dataset	0.8531	0.858	0.8526
BERT-ResNet-PSO	Dataset for Cotton Leaf Disease Detection	0.9006	0.8961	0.8969
	Cotton leaf disease dataset	0.8897	0.8853	0.8862

EfficientNetB1-CBAM	Dataset for Cotton Leaf Disease Detection	0.8872	0.8924	0.8876
	Cotton leaf disease dataset	0.8764	0.8818	0.8767
Proposed MMDFN	Dataset for Cotton Leaf Disease Detection	0.9539	0.9576	0.9535
	Cotton leaf disease dataset	0.9475	0.9513	0.9471

The cross-dataset generalization results of the proposed MMDFN and the basic models are shown in Table 8 on two separate test datasets of cotton leaf disease. The models were trained on one set of data and tested on the other set, to verify that they are robust when dealing with different data distributions. The proposed MMDFN got the highest precision, F1-scores, sensitivity, specificity, NPV and MCC, and the highest accuracy with both datasets. Moreover, MMDFN exhibited the smallest false positive rate (4.38%–4.91%) and false negative rate (4.92%–5.63%) that further demonstrated its excellent generalization capability to outlier images compared with the other methods. The results show the proposed multimodal feature fusion and SWO-based feature optimization can keep the model performance stable on independent cotton leaf disease datasets.

4.5. Discussion

The complementary feature handcrafted and deep feature representations are the main reason for the superior performance of the proposed MMDFN. Handcrafted descriptors are used to encode the texture, color and shape of the lesion areas of the leaves, while the deep network is used to extract the high-level semantic features, which is helpful for the discrimination of visually similar leaf disease categories. However, in certain images with complex background fields, non-uniform lighting, overlapping leaves, and faint disease symptoms, a few misclassified samples were observed due to the similar visual features of the different classes. The SWO-based feature optimization process is able to effectively eliminate redundant features prior to classification, improving the discriminative power of the combined feature set. The multimodal fusion and optimization steps add more computation in the training process but they are applied offline and the trained MMDFN is efficient for inference, only using the optimized subset of features. The observations confirm that the proposed framework brings a good trade-off between classification accuracy and computation efficiency and shows similar performance in various experimental scenarios.

5. Conclusions

This study presented a Multi-Modal Deep Fusion Network (MMDFN) for automated cotton leaf disease detection by integrating VGG-16-based Faster R-CNN for ROI localization, multimodal feature fusion, and the proposed SWO algorithm for feature selection. The proposed framework achieved 98.4% accuracy, 98.6% sensitivity, and 98.2% specificity, demonstrating superior performance compared with existing methods. Despite these promising results, the framework has certain limitations. Its performance may be affected by complex field conditions such as occluded leaves, varying illumination, and class imbalance. Moreover, the hybrid optimization process increases training complexity, and evaluation on relatively limited public datasets may restrict generalization to diverse agricultural environments. Future work will focus on developing lightweight versions of the model using pruning and quantization for mobile deployment, expanding the dataset through large-scale field data collection, and investigating federated learning to improve cross-domain generalization while preserving data privacy.

All the Declarations and Statements

Author Contributions Statements

Mohan Ajmeera – Conceptualization, Methodology, and Supervision: Proposed research ideas, constructed the overall framework, and supervised project execution.

Dr. P. Chiranjeevi – initial research idea, formal Analysis and performance evaluation.

Dr. A. Krishna Mohan – Model Training, Validation, and Performance Evaluation.

All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest statement

Authors declare no conflicts of interest.

Funding declarations

This research received no external funding.

Data Availability Statement

All data used for this study are available within the manuscript.

Ethical Declaration

Ethical approval was not required for this study, as no human participants, animal subjects, or sensitive experimental procedures were involved.

Acknowledgments

None

Declaration of Generative AI in Scholarly Writing

Generative AI was used only to improve the language and readability of the manuscript. The authors remain fully responsible for the originality, accuracy, and scientific integrity of the content.

Abbreviations

The following abbreviations are used in this manuscript:

BERT- Bidirectional Encoder Representations from Transformers
 CBAM -Convolutional Block Attention Module
 CNN-LSTM- Convolutional Neural Networks- Long Short-Term Memory
 GLRLM - Gray Level Run Length Matrix
 GLSZM - Gray Level Size Zone Matrix
 GWO - Gray Wolf Optimization
 MMDFN - Multi-Mode Deep Fusion Network
 MLP - Multi-Layer Perceptron
 SLOA - Snow Leopard Optimization
 SWO - Snowy Wolf Optimization
 SE - Squeeze-and-Excitation
 ROI - Region of Interest
 PSO - Particle Swarm Optimization
 VRCNN - VGG-16-based Faster Region-based Convolutional Neural Network
 YOLO - You Only Look Once

Appendix

Not applicable

References

- [1] Memon, M.S., Kumar, P. and Iqbal, R., 2022. Meta Deep Learning Leaf Disease Identification Model for Cotton Crop. *Computers*, 11(7), p.102.
- [2] Patil, B.V. and Patil, P.S., 2021. Computational Method for Cotton Plant Disease Detection of Crop Management Using Deep Learning and Internet of Things Platforms. In *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020* (pp. 875-885). Springer Singapore.
- [3] Caldeira, R.F., Santiago, W.E. and Teruel, B., 2021. Identification of cotton leaf lesions using deep learning techniques. *Sensors*, 21(9), p.3169.
- [4] Liang, X., 2021. Few-Shot Cotton Leaf Spot Disease Classification Based on Metric Learning. *Plant Methods*, 17, pp.1-11.
- [5] Hyder, U. and Talpur, M.R.H., 2024. Detection of Cotton Leaf Disease with a Machine Learning Model. *Turkish Journal of Engineering*, 8(2), pp.380-393.
- [6] Ahmad, M., Ullah, F., Hamza, A.R.A., Usman, M., Imran, M., Batyrshin, I., Gelbukh, A. and Sidorov, G., 2024. Cotton Leaf Disease Detection Using Vision Transformers: A Deep Learning Approach. *Crops*, 1, p.3.
- [7] Faisal, H.M., Aqib, M., Rehman, S.U., Mahmood, K., Obregon, S.A., Iglesias, R.C. and Ashraf, I., 2025. Detection of Cotton Crop Diseases Using a Customized Deep Learning Model. *Scientific Reports*, 15(1), p.10766.
- [8] Kaur, G., Al-Yarimi, F.A.M., Bharany, S., Rehman, A.U. and Hussien, S., 2025. Explainable AI for Cotton Leaf Disease Classification: A Metaheuristic-Optimized Deep Learning Approach. *Food Science & Nutrition*, 13(7), p.e70658.
- [9] Ali, T., Zakir, R., Ayaz, M., Murtaza, M., Hijji, M. and Hadi Aggoune, E.M., 2025. Cotton crop disease detection and classification Using a Statistical Prediction Model in a Deep Learning Approach. *Multimedia Tools and Applications*, 84(41), pp.49503-49525.
- [10] Sibiya, M. and Sumbwanyambe, M., 2021. Automatic fuzzy logic-based maize common rust disease severity predictions with thresholding and deep learning. *Pathogens*, 10(2), p.131.
- [11] Chinnadurai, S. and Selvakumar, S., 2025. Image-based cotton leaf disease diagnosis using YOLO and faster R-CNN techniques. *Scientific Reports*.
- [12] Salot, P., Pancholi, P., Rathod, H., Rathod, S., Thakkar, M. and Shah, J., 2025. Cotton leaf analysis based early plant disease detection using Machine Learning. *Journal of Integrated Science and Technology*, 13(1), p.1015.
- [13] Udawant, P. and Srinath, P., 2022. Cotton leaf disease detection using instance segmentation. *Journal of Cases on Information Technology (JCIT)*, 24(4), pp.1-10.

- [14] M. SithaRam, V. Anusha, P.N. Sri and G.H. Sri, "A Novel Methodology for Cotton Leaf Disease Detection using CNN," in 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2024, pp. 202–207. In 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 202-207). IEEE.
- [15] Rehman, A., Akhtar, N. and Alhazmi, O.H., 2025. Monitoring and predicting cotton leaf diseases using deep learning approaches and mathematical models. *Scientific Reports*, 15(1), p.22570.
- [16] Rajasekar, V., Venu, K., Jena, S.R., Varthini, R.J. and Ishwarya, S., 2021. Detection of cotton plant diseases using deep transfer learning. *Journal of Mobile Multimedia*, 18(2), pp.307-324.
- [17] Patil, B.V., Patil, P.S. and Patil, P.R., 2025. HARNN-IoT: IoT-Enabled Hybrid Deep Learning Model for Cotton Plant Disease Detection and Classification. *Journal of Phytopathology*, 173(6), p.e70213.
- [18] Jadhav, A., Chaudhari, S., Gupta, A. and Mithbavkar, D.S., 2023. Cotton disease detection and cure using CNN. *Int. J. Res. Appl. Sci. Eng. Technol.*, 11(4), pp. 3385–3389.
- [19] Ahmed, M.R., 2021. Leveraging convolutional neural network and transfer learning for cotton plant and leaf disease recognition. *Int. J. Image Graph. Signal Process.*, 13(4), pp. 47–62.
- [20] Wang, Z., Zhang, H.W., Dai, Y.Q., Cui, K., Wang, H., Chee, P.W. and Wang, R.F., 2025. Resource-efficient cotton network: A lightweight deep learning framework for cotton disease and pest classification. *Plants*, 14(13), p.2082.
- [21] R. He, F. Zhang, J. Zhu, Y. Wang, D. Yang, T. Zhang, and P. Li, "YOLOv9-LSBN: An Improved YOLOv9 Model for Cotton Pest and Disease," IEEE Access, 2025.
- [22] Singh, C., Wibowo, S. and Grandhi, S., 2025. A Hybrid Deep Learning Approach for Cotton Plant Disease Detection Using BERT-ResNet-PSO. *Applied Sciences*, 15(13), p.7075.
- [23] Rahman, M.A., Ullah, M.S., Devnath, R.K., Chowdhury, T.H., Rahman, G. and Rahman, M.A., 2025. Cotton leaf disease detection: an integration of CBAM with deep learning approaches. *Int. J. Comput. Appl.*, 975, p. 8887.
- [24] Pandiyaraju, V., Anusha, B., Senthil Kumar, A.M., Jaspin, K., Venkatraman, S. and Kannan, A., 2025. Spatial attention-based hybrid VGG-SVM and VGG-RF frameworks for improved cotton leaf disease detection. *Neural Computing and Applications*, 37(14), pp.8309-8329.
- [25] Joshi, K., Yadav, Y., Hooda, S., Nandal, R., Singh, B., Singh, K., Tuteja, N., Gill, R. and Gill, S.S., 2025. Classification of cotton leaf disease using YOLOv8 based k-fold cross validation deep learning method for precision agriculture. *Scientific Reports*, 15(1), p.35602.
- [26] Neelavathi, G. and Venkatasalam, K., 2026. A novel Hybrid Vision Transformer with dense attention capsule network (HVT-DACapNet) model for cotton plant disease detection. *Scientific Reports*.
- [27] Aslan, E. and Özüpak, Y., 2026. A hybrid SE-ResNet50 deep learning framework for high-accuracy and explainable cotton leaf disease classification. *BMC Plant Biology*.
- [28] Sarwar, R., Aslam, M., Khurshid, K.S., Ahmed, T., Martinez-Enriquez, A.M. and Waheed, T., 2021. Detection and classification of cotton leaf diseases using faster R-CNN on field condition images. *Acta Sci. Agric.*, 5(10).
- [29] Shrotriya, A., Sharma, A.K., Bairwa, A.K. and Manoj, R., 2024. Hybrid ensemble learning with CNN and RNN for multimodal cotton plant disease detection. *IEEE Access*, 12, pp.198028-198045.
- [30] J. Guo, W. Ye, D. Wang, Z. He, Z. Yan, M. Sato, and Y. Sato, "A novel snow leopard optimization for high-dimensional feature selection problems," Sensors, vol. 24, no. 22, p. 7161, 2024.
- [31] Gülmez, B., 2023. A novel deep learning model with the Grey Wolf Optimization algorithm for cotton disease detection. *Journal of Universal Computing Science*.
- [32] Jashim, F.B., Refat, F.R., Karim, M.H., Mahmud, F.U. and Ashrafi, F., 2025. A web-based application for cotton leaf disease classification using vision transformer. *Int. J. Sci. Res. Arch.*, 15(2), pp. 1405–1416.
- [33] S. Sabuktagin, "Dataset for Cotton Leaf Disease Detection," Kaggle. [Online]. Available: <http://kaggle.com/datasets/sabuktagin/dataset-for-cotton-leaf-disease-detection>
- [34] S. Seroshkarim, "Cotton Leaf Disease Dataset," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/seroshkarim/cotton-leaf-disease-dataset>

Authors' Profiles



Mohan Ajmeera is currently doing Ph.D. in Computer Science & engineering, Jawaharlal Nehru Technological University Kakinada. He completed his B-Tech in Department of Computer Science & Engineering and M-Tech in Computer Science & Engineering. He is currently working as a Assistant Professor in department of Computer Science & Engineering, Chaitanya bharathi institute of Technology Hyderabad, India. His research interest includes Deep Learning, Machine Learning, data mining, and Cryptography and network security.



Dr. P. Chiranjeevi is currently working as Professor of CSE Department, Amrita Sai Institute of Science and Technology, Bathinapadu, Andhra Pradesh, India. He received his BSc degree in Computer Science & Engineering, Nagarjuna iversity, Guntur, Master's degree in CSE, Nagarjuna University, Guntur, India. He received his Ph.D. degree in computer science & Engineering, SRM University, India. His research interests: Cloud computing, machine learning, data mining metaheuristic optimization, decision support systems and predictive models.



Dr. A. Krishna Mohan is currently working as Professor of CSE Department, University College of Engineering, JNTUK, Kakinada, India. He has 6+Years of Professional Software Development Experience. He received his BE degree in Computer Science & Engineering, Nagarjuna University, Guntur, Master's degree in CSE, Andhra University. He received his Ph.D. degree in computer science & Engineering, Jawaharlal Nehru Technological University. His research interest Machine learning, data mining, intelligent bioinformatics, metaheuristic optimization, decision support systems and predictive models. His expertise in Big data, Data mining, Hadoop, Statistical Analysis Using R, Hadoop tools like Hive, PIG and HBase.

How to cite this paper

M. Ajmeera, P. Chiranjeevi, and A. K. Mohan, "MMDFN: A Multi-modal Deep Fusion Network with Hybrid Optimization for Automated Cotton Leaf Disease Detection," International Journal of Intelligent Systems and Applications (IJISA), Vol.18, No.4, pp.116–137, 2026. DOI:10.5815/ijisa.2026.04.07