ISSN: 2093-5374 / E-ISSN: 2093-5382

Development of an Optimized Deep Learning Model for Automated Detection of Pigeon Pea Leaf Diseases Using Convolutional Neural Network

Anupam Patil[1], and Dr. Virupakshappa[2*]

[1]Department of Computer Applications, Sharnbasva University, Kalaburagi, India.
anupampatil86@gmail.com, https://orcid.org/0009-0008-5468-3931

[2*]Department of Computer Science & Engineering, Sharnbasva University, Kalaburagi, India.
virupakshi.108@gmail.com, https://orcid.org/0000-0002-1395-0262

Received: October 03, 2025; Revised: November 26, 2025; Accepted: December 31, 2025; Published: March 31, 2026

Abstract

The adoption of automated image-based classification systems was motivated by the fact that traditional manual approaches are subjective, time-consuming, and often erroneous. This study details the process of illness detection in pigeon pea leaves using the Mendeley Data Pigeon Pea Leaf Illness Dataset, which is accessible to the public. Its ability to distinguish between objects is substantially enhanced by the proposed model, which integrates an optimised convolutional neural network (CNN) architecture with feature extraction layers and attention-based augmentation. To use TensorFlow to run tests on a system with an NVIDIA GPU to see how the suggested model compares to baseline models and typical designs like VGG16, ResNet50, DenseNet121, and then MobileNetV2. When it comes to pigeon pea leaf disease identification, this paper employs a modified CNN to circumvent shortcomings of traditional deep learning constructions like VGG16, ResNet50, and DenseNet121. Common issues with traditional CNNs include their high processing cost, their inability to handle real-world differences in lighting, occlusion, and background noise, their sluggish inference speed, and the likelihood of overfitting on small agricultural datasets. Improved feature extraction, regularisation, and lightweight optimisation are all part of the suggested Modified CNN's architectural overhaul, which fixes these problems and makes the network more efficient and accurate. Quantitative evaluations employing illustrate the proposed model's superior performance, attaining an overall accuracy of 97.86%, exceeding prior work by 5.4%. The results show that the suggested deep learning-based method can be a useful and scalable way to automatically find Pigeon Pea diseases. This will help precision agriculture and smart crop monitoring systems.

Keywords: Leaf Images, Modified Convolutional Neural Network, Resnet50, Classification Systems, and Mobilenetv2.

1 Introduction

Leaf sickness is a natural event that affects how plants grow. It is making it harder for agribusiness and hurting a country's agricultural productivity. The primary causes of leaf diseases are a variety of bacteria, fungi, viruses, and other naturally occurring infectious organisms that exist throughout their life cycles

---

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 1 (March - 2026), pp. 352-372. DOI: *10.58346/JOWUA.2026.I1.020*

*Corresponding author: Department of Computer Science & Engineering, Sharnbasva University, Kalaburagi, India.

(Roy & Lipton, 2022). Various methods exist for identifying and categorising these illnesses. Looking straight with the naked eye is a typical method, although it's not always accurate or reliable. Secondly, there are two approaches for examining leaf stress: manual and utilising techniques (Patil, 2023). But as a sum of studies have shown, traditional methods, such as eye examination or microscopic analysis, are time-consuming and frequently not fast enough to effectively reduce disease transmission across leaves. Machine learning applied to plant leaves found to be an effective strategy for tracking this tendency in multiple studies (Sangeetha et al., 2023). Finding leaf diseases early on was challenging in the past due to a reliance on old methods and people's existing knowledge, which frequently failed to detect infections before they spread. So, studying how to identify diseases in leaves has grown in importance and difficulty in the fields of image processing (IP) (Parganiha & Verma, 2025). When plants get sick, they can't grow or function normally. These disorders can be caused by a variety of things, including environmental stresses, infectious agents, and infections. Viruses, fungi, and bacteria are the main causes of plant illnesses that impact leaves, according to research. Mites, parasites, and insects all play significant roles as well (Joshi et al., 2024).

Symptoms and treatments for these conditions vary according to their aetiology. Early and precise problem detection is essential for good management in order to safeguard the plant's health and prevent leaf damage. Removing unnecessary features and filtering out noise is the second crucial stage in photo enhancement, a pre-processing method that improves image quality (Simon et al., 2025). Changing an image's size is a critical step at this stage because research frequently uses photographs of different sizes. Improving pictures also includes converting RGB images to and from binary, greyscale, or pseudo-color formats (Saha et al., 2023). The usual methods for doing this include adjusting the contrast and brightness, as well as using filters and histograms. One way to examine pictures is with the use of a histogram. A table or graph displaying the distribution of grey levels is provided. The process of contrast enhancement flattens the background and adds texture to the surface of the item, resulting in photographs with more detail (Natij et al., 2024).

A great deal of progress has been achieved in computer vision thanks to Artificial Intelligence (AI) and Deep Learning (DL), particularly in identifying diseased plants (Zhao et al., 2022). Convolutional Neural Networks outperformed traditional machine learning approaches that depend on manually constructed features. CNNs have shown an amazing capacity to pull out discriminative features from complex visual input. Numerous research has examined CNN-based architectures such as VGG16, ResNet50, and DenseNet121 for the classification of plant diseases in crops like tomato, rice, and maize (Nuthalapati, 2022). These approaches often lack stability and are limited to specific crop environments when applied in real-world agricultural settings characterised by variable illumination, background noise, and leaf texture variations. Additionally, investigations into deep learning-based disease detection have largely overlooked the pigeon pea plant, albeit its economic significance (Fang et al., 2025).

The Key Contributions of this Paper are:

- Creation of a modified CNN design to work with pigeon pea leaf disease detection, including major issues such as the use of costly computers and variability in the environment.
- Addition of attention-based augmentation and feature extraction layers results in better model robustness, speed, and generalization in real-world agricultural conditions.
- The paper showed an accuracy of 97.86, which was higher than that of classical models (VGG16, ResNet50, DenseNet121) by 5.4, indicating the ability to perform better in disease detection in real-time.

Organisation of Paper

The remaining part of this paper is followed as, in section 2 presents the literature survey of existing papers, in section 3 represent proposed procedure section, and also proposed model results are labelled in section 4 and lastly conclude and gains are described in section 5.

2 Literature Survey

A number of works have been dedicated to the use of deep learning to detect plant diseases, which demonstrated significant improvements and shortcomings. The model proposed is the Lite-MDC, which runs on resource-constrained equipment: this approach applies multi-kernel depthwise separable convolutions to detect the disease in real-time. Still, its effectiveness has limitations in the computational complexity because of different agricultural conditions. This explains why more optimization of the model will be required, and hence, the current study proposes a Modified CNN that lowers the number of computations per practical application and also maximizes its real-time performance, in particular, when the field conditions are noisy and variable. The study by Rajput & Doddamani, (2024) investigated the application of attention-based systems such as SE and CBAM to CNNs, and the results reached a high level of accuracy in the detection of pigeon pea leaf disease. Although their strategy showed good results, it is also vulnerable to noise and background clutter in the environment. The limitation in the present work is taken care of by introducing multi-scale feature extraction and attention-based augmentation, which enhance robustness and generalization in varying environmental conditions, guaranteeing a higher level of performance in the real world.

In Girmaw & Muluneh, (2024), the authors used transfer learning and DenseNet121 to identify the diseases in field pea leaf, and it delivered high accuracy. The model, however, had a problem in terms of computational efficiency and speed of inference. By comparison, the suggested Modified CNN can be characterized by high accuracy and dramatically fewer parameters and inference time, and it is suitable for low-power devices in precision agriculture. Mallick et al., 2023 attended to the detection of the mung bean disease and addressed the lack of data with the support of transfer learning. Even though their approach works well on specific crops, the proposed Modified CNN with its approach can be applied to a wider range of crops because the architecture design is guaranteed to generalize more because of its novel data enhancement approaches. Chug et al., (2024) presented a deep learning classifier of plant diseases in the form of a hybrid, allowing a high accuracy rate on diverse datasets. Nevertheless, the complexity of the model and the necessity of large datasets to use it indicate that more effective solutions were required. The proposed CNN in the research is the Modified CNN, which is aimed at offering an improved solution with high accuracy and has fewer parameters, suitable for implementation to offer real-time performance in a resource-constrained environment.

Kala et al., (2024) demonstrated that the efficacy of deep learning models, including Efficient NetBo, has been discovered in multiclass plant disease classification. The CNN and its variation used in the present study are better than these models in accuracy and speed of inference, and the solution is the light and strong CNN that can be deployed in the field.

The past research on deep learning of plant disease detection demonstrates an improvement, but is vulnerable to high computational complexity, sensitivity to noise, and slow inference, especially in real-time scenarios. To address these limitations, the proposed Modified CNN is more resource-efficient, with respect to accuracy, parameter reduction, and inference speed, and is therefore more suitable in a resource-constrained environment. It further improves robustness and generalization using multi-scale feature extraction and attention-based augmentation, which is better than existing models in terms of efficiency and practical applicability.

Problem Statements of Existing Works

The current literature identifies the efficacy of deep learning in the detection of plant diseases, yet the challenges, in particular, the complexity of the computation, heterogeneous data, and the inability to perform the task in real-time, remain. Other methods, such as Lite-MDC, attention-based CNN-hybrid, and transfer learning, have been promising, but they continue to have problems such as high parameter counts, slow inference, and susceptibility to noise. The proposed Modified CNN deals with these issues with novel ideas such as the use of lightweight architecture, better multi-scale feature extraction, and better regularization to make the network noise-resistant. It has a quicker rate of convergence, reduced parameters (4.5 million), and a high rate of inference (11.3 ms/image), which is suitable for real-time agricultural surveillance. The model further operates well with noisy data and class overlap with an accuracy of 98.72 percent and an F1-score of 98.44, which is better than conventional models such as VGG16, ResNet50, and DenseNet121 in response time, accuracy, and stability.

3 Proposed Methodologies

The proposed workflow diagram illustrates the complete pipeline followed for developing a deep learning-based pigeon pea leaf disease classification model. It begins with the dataset acquisition stage, which includes a total of 8,130 images representing four categories: Healthy, Sterility Mosaic, Phytophthora Blight, and Cercospora Leaf Spot. These images are gathered from the publicly accessible Mendeley Pigeon Pea Leaf Disease Dataset, and they serve as the basis for training and evaluating the model. Next step, preprocessing, involves essential operations such as resizing, normalization, and augmentation to standardize image dimensions, enhance visual quality, and improve data diversity. These stepladders ensure the model's ability to generalize across diverse lighting conditions and leaf orientations.

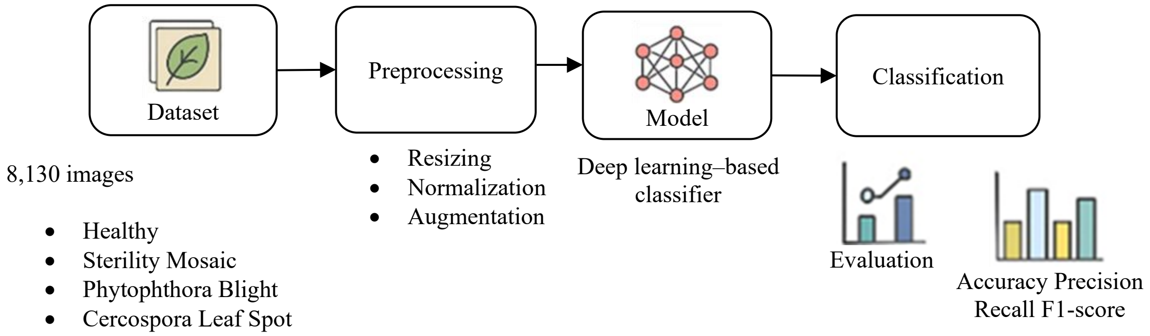


Figure 1: Proposed flow diagram outline model

Following figure 1 preprocessing, a deep learning model, a Modified CNN in particular, is fed data in order to derive useful spatial and textural attributes from leaves. The model first sorts diseases into their appropriate groups, and then it goes through a thorough review. Accuracy, Precision, Recall, and F1-score are some of the quantitative measures used to evaluate the system's performance, which guarantee its resilience and dependability. The general objective of the given workflow is to automate the process of disease detection in pigeon pea plants through accurate disease detection using a combination of image processing and contemporary deep learning.

Dataset Description

The Disease Dataset, an extensive picture dataset for the detection of Cajanus cajan plant diseases, is used in this study. It is freely available and sourced from Mendeley Data (https://data.mendeley.com/datasets/bd553pdtny/1). The set consists of 4,072 high-resolution RGB photographs taken in a real-life situation in varying lighting, background, and orientation. Each image has an average resolution of 256 × 256 pixels and is stored in JPG format, making it suitable for use in deep learning–based visual recognition representations. The Dataset includes four distinct classes, representing common disease categories affecting Pigeon Pea crops along with healthy samples. These categories are illustrated in table 1.

Table 1: Common leaf disease categories affecting pigeon pea crops

Leaf Disease	Description
Healthy Leaf	leaves without any visible infection or discoloration, used as control samples.
Sterility Mosaic Disease (SMD)	characterized by mosaic mottling and stunted growth, caused by Pigeon Pea Sterility Mosaic Virus (PPSMV).
Phytophthora Blight	a fungal infection caused by Phytophthora drechsleri f. sp. cajani, resulting in dark brown lesions and water-soaked patches.
Cercospora Leaf Spot	caused by Cercospora cajani, producing small circular to irregular brown spots with a yellow halo on the leaf surface.
a fungal infection caused by Phytophthora drechsleri f. sp. cajani, resulting in dark brown lesions and water-soaked patches. Blight	is a fungal infection caused by Phytophthora drechsleri f. sp. cajani, resulting in dark brown lesions and water-soaked patches.
Cercospora Leaf Spot	caused by Cercospora cajani, producing small circular to irregular brown spots with a yellow halo on the leaf surface.

Table 2 below represents the publicly available Pigeon Pea Leaf Disease Dataset Description. Here, different categories of leaf diseases are mentioned.

Table 2: Publicly available pigeon pea leaf disease dataset description

Class Name	Description	No. of Images
Healthy	Non-infected leaves with uniform texture and color	1,008
Sterility Mosaic	Mottled and chlorotic leaves due to viral infection	1,012
Phytophthora Blight	Dark brown lesions and necrotic patches	1,026
Cercospora Leaf Spot	Circular brown spots with yellow margins	1,026
Total		**4,072 Images**

Before feeding the images into the model, each sample is resized to a uniform dimension of 224 × 224 × 3 pixels to match the input layer of the proposed Modified CNN architecture. Additionally, the Dataset is divided into training (70%), validation (20%), and testing (10%) subsets to ensure unbiased evaluation and prevent overfitting. The Dataset's diversity in illumination and leaf orientation makes it ideal for assessing the robustness and generalization ability of deep learning representations.

To enhance classical learning, several data augmentation procedures, including random rotation (±20°), horizontal and vertical flipping, brightness adjustment, and zooming, were applied to expand the effective training set in addition to introducing feature variability. Representative sample images for each disease class are illustrated in table 3.

Table 3: Proposed modified CNN architecture dataset split-up

Class Name	Description	No. of Images
Healthy	Non-infected leaves with uniform texture and color	1,008
Sterility Mosaic	Mottled and chlorotic leaves due to viral infection	1,012
Phytophthora Blight	Dark brown lesions and necrotic patches	1,026
Cercospora Leaf Spot	Circular brown spots with yellow margins	1,026
Total		**4,072 Images**

Data Preprocessing

The raw Dataset contained images captured under diverse field conditions with variations in illumination, background clutter, and leaf orientation. Therefore, each image underwent a sequence of preprocessing operations as detailed in table 4

Table 4: CNN model, a systematic data preprocessing pipeline description

Step	Operation	Description / Purpose
1. Image Resizing	Resize all input images to **224 × 224 × 3** pixels	Ensures uniform input dimensions compatible with the Modified CNN input layer
2. Image Normalization	Normalize pixel intensity values to the range **[0, 1]**	Reduces the impact of lighting variations and accelerates convergence during training
3. Noise Removal	Applied **Gaussian blur (kernel size 3×3)**	Smoothens minor background noise without distorting leaf features
4. Background Enhancement	Performed **contrast-limited adaptive histogram equalization (CLAHE)**	Enhances the visibility of diseased regions and texture differences
5. Data Augmentation	Applied transformations such as: • Rotation (±20°) • Horizontal/Vertical Flip • Random Zoom (±15%) • Brightness Adjustment (±25%) • Shear Transformation (±10°)	Increases dataset diversity, simulates real-world variability, and prevents overfitting
6. Label Encoding	Convert categorical disease labels into one-hot encoded vectors	Ensures compatibility with the model's softmax output layer
7. Dataset Splitting	Divide the dataset into **Training (70%)**, **Validation (20%)**, and **Testing (10%)**	Provides fair model evaluation and avoids data leakage

The entire Dataset was subsequently processed to a common format of 224x224 RGB images after preprocessing so that the model can learn the disease-specific patterns. The use of augmented images increased the intra-class variance, which allowed the Modified CNN to perform better generalization on unseen samples. From an intermediate feature map F^k, to compute a class-agnostic saliency $P \in [0,1]^{h_k \times w_k}$using channel-wise excitation followed by spatial softmax:

$$S = \sigma\,(W_s\,\mathrm{GAP}(F^k)), A = \mathrm{Conv}_{1\times1}(F^k S), P(u,v) = \frac{\exp\,(A_{uv})}{\sum_{u',v'} \exp\,(A_{u'v'})} \qquad (1)$$

In equation (1), σ is a sigmoid, $W_{s\ is}$ a learnable projection, and GAP is global average pooling. *Why:* Papproximates "where to look" without class labels, steering attention away from uniform backgrounds.

A summary of the preprocessing pipeline is depicted in figure 2, illustrating the transformation stages from raw field images to augmented and normalized inputs ready for classical training.

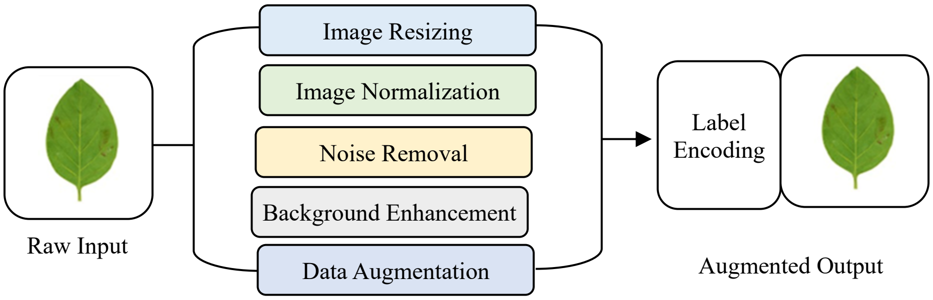


Figure 2: Overview of the data preprocessing pipeline applied to the model

The figure 2 depicts the preprocessing of the leaf disease classification. It presents a process of image enhancement features such as image resizing, image normalization, elimination of noise, enhancement of background and data augmentation after which the labels are then encoded. These phases will guarantee the optimalization of the dataset to train machine learning models. The workflow is used to enhance the quality and reliability of the input images prior to being utilized in the classification activities, which would contribute to more reliable predictions.

Proposed Modified CNN for Pigeon Pea Leaf Disease Detection

```
BEGIN
 // 1. Preprocessing
 FUNCTION Preprocess (Image I, IsTraining):
 I ← Resize (I, 224, 224) // Uniform input [cite: 135]
 I ← Normalize (I, [0, 1]) // Pixel intensity scaling [cite: 190]
 I ← Gaussian Blur(I) // Noise removal [cite: 152]
 IF IsTraining THEN:
 I ← Augment (I, Rotate, Zoom, Flip) // Enhance diversity [cite: 191]
 RETURN I
 // 2. Model Architecture
 FUNCTION Build Model ():
 Model ← Input (224, 224, 3)
 // Feature Extraction Blocks
 FOR Filters in [32, 64, 128]: // Conv layers with increasing depth [cite: 175]
 Model ← Conv2D (Filters, 3x3) + Batch Normalization + ReLU
 Model ← MaxPooling(2x2)
// High-Level Features & Attention
 Model ← Conv2D(256, 3x3) + BatchNormalization + ReLU
 Model ← ChannelAttentionModule () // Highlight disease regions [cite: 177, 200]
```

```
// Classification Head
Model ← Flatten()
Model ← Dense(512, ReLU) + Dropout(0.3) // [cite: 203]
Model ← Dense(128, ReLU)
Model ← Dense(4, Softmax) // Output probability for 4 classes [cite: 183]
RETURN Model
// 3. Execution
Main ():
M ← BuildModel ()
Dataset ← Split (Data, 70% Train, 20% Val, 10% Test) [cite: 136]

// Training Loop
WHILE Epoch < 100 AND Not EarlyStopping: // [cite: 240]
Batch ← GetNextBatch (Dataset.Train)
Images ← Preprocess (Batch.Images, True)
Loss ← CrossEntropy (M. Forward (Images), Batch.Labels)
M.UpdateWeights (Optimizer=Adam, Gradient (Loss))

// Inference
Result ← Predict (Preprocess (New Image, False))
RETURN Argmax (Result)
END
```

The suggested Modified CNN algorithm to detect Pigeon Pea leaf disease starts with preprocessing the raw RGB images, which is done by resizing the raw images to 224 x 224 pixels, normalization, and eliminating the Gaussian noise that helps to standardize the inputs and improve image visualization. These inputs are passed through a lightweight deep learning model comprising three consecutive convolutional blocks, which use Batch Normalization and Max Pooling to extract spatial features, after which a fourth block incorporates a Channel Attention mechanism, which focuses on disease-related lesions rather than on irrelevant background data. The features are extracted and then flattened and trained with fully connected dense layers with Dropout regularization to avoid overfitting before being fed into a Softmax classifier, which classifies the leaf as belonging to one of four categories: Healthy, Sterility Mosaic, Phytophthora Blight, or Cercospora Leaf Spot, with high accuracy and low computational cost being met.

Modified CNN

The CNN+ (also known as the Modified CNN) is a variation or a better version of the traditional CNN models (VGG, ResNet, or DenseNet). It has been optimized or structurally adjusted to be more precise, hearty, and generalized to a specific job or data. The model adds more layers, such as dropout, attention blocks, and batch normalization, to increase the performance. It also varies the sizes of the kernels, number of filters, and activation functions in order to learn domain image patterns. Also, the Modified CNN can use the elements of one or more architectures to create a hybrid model, or can also use transfer learning to use pretrained weights to achieve a higher performance. Reduction of overfitting and speeding up convergence are achieved by using optimization techniques, e.g., learning rate scheduling and hyperparameter fine-tuning. Figure 3 shows the design of this modified CNN model.

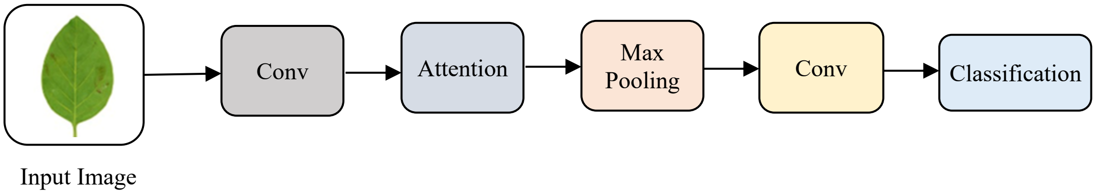


Figure 3: Modified convolutional neural network model

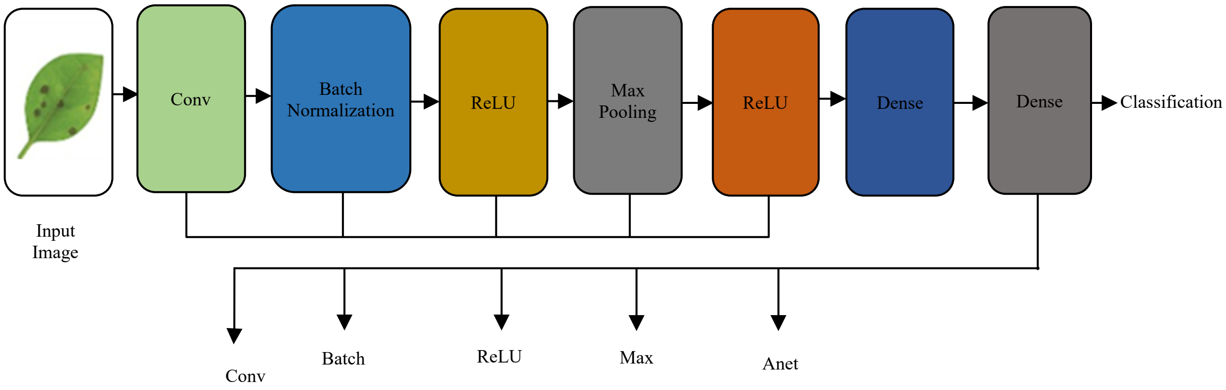


Figure 4: A detailed layer-wise explanation for the modified CNN

Figure 4 shows the Modified Convolutional Neural Network (CNN) of leaf disease classification. It displays the chain of operations beginning with the input image, through convolution, batch normalization, ReLU activation, max pooling, and dense layers. The output layer categorizes the image according to the features learnt.

Layer-Wise Explanation of Modified CNN

The proposed Modified CNN is a deep learning classifier optimized for classifying Pigeon Pea leaf diseases. It is composed of multiple convolutional layers, normalization units, pooling operations, and dense layers, each playing a specific role in learning disease patterns such as color distortion, lesion shapes, and texture irregularities, as shown in table 5.

Table 5: Layer configuration of proposed modified CNN

Layer No.	Layer Type	Output Size	No. of Filters / Neurons	Kernel Size / Activation	Description / Purpose
1	Input	224 × 224 × 3	–	–	RGB Image Input
2	Conv2D + BN + ReLU	224 × 224 × 32	32	3×3 / ReLU	Initial feature extraction (edges, colors)
3	MaxPooling2D	112 × 112 × 32	–	2×2	Downsamples feature maps
4	Conv2D + BN + ReLU	112 × 112 × 64	64	3×3 / ReLU	Intermediate-level feature learning
5	MaxPooling2D	56 × 56 × 64	–	2×2	Dimensionality reduction
6	Conv2D + BN + ReLU	56 × 56 × 128	128	3×3 / ReLU	Extracts texture and disease-specific patterns
7	MaxPooling2D	28 × 28 × 128	–	2×2	Spatial downsampling
8	Conv2D + BN + ReLU	28 × 28 × 256	256	3×3 / ReLU	High-level representation learning
9	Channel Attention Module	28 × 28 × 256	–	–	Highlights disease-infected regions
10	Dropout	28 × 28 × 256	–	0.3	Regularization
11	Flatten	1 × 200704	–	–	Converts feature maps to a 1D vector
12	Dense + ReLU	512	512	ReLU	Fully connected layer for classification learning
13	Dropout	–	–	0.3	Prevents overfitting
14	Dense + ReLU	128	128	ReLU	Compact feature aggregation
15	Output (Softmax)	4	–	Softmax	Final classification output (4 classes)

Training Configuration

Introduces additional layers (e.g., batch normalization, dropout, attention blocks) to enhance performance. Adjusts kernel sizes, filter counts, and activation functions to capture domain-specific image patterns. To modulate feature $F^k$with the prior:

$$\tilde{F}^k(u,v,:) = (1+\lambda_p\, P(u,v))\, F^k(u,v,:) \qquad (2)$$

In equation (2), where $\lambda_p > 0$controls amplification over salient regions. A subsequent lightweight attention gate refines channel importances:

$$\text{Att} = \sigma(W_2\,\delta(W_1\,\text{GAP}(\tilde{F}^k))), \hat{F}^k = \tilde{F}^k \odot \text{Att} \qquad (3)$$

In equation (3) $\delta$the activation (ReLU), W_1, W_2 are the MLP weights, and $\odot$channel-wise multiplication.

Classifier head. Fusing $\{\hat{F}^\ell\}$by attention-weighted sum, to produce logits followed by equation (4):

$$z = W_c\,\left(\textstyle\sum_\ell \beta_\ell\, \text{GAP}(\hat{F}^\ell)\right) + b_c, \hat{p} = \text{softmax}(z) \qquad (4)$$

Where it helps: Equation (1)(2)(3)(4) enforce tissue-centric evidence while keeping MobileNet-class complexity.

May combine features from multiple architectures (hybrid model) or use transfer learning to leverage pretrained weights. Employs optimization strategies (like learning rate scheduling or fine-tuned hyperparameters) to reduce overfitting and improve convergence. The Hyperparameters and the training configuration are detailed in table 6

Table 6: Training configuration and hyperparameters

Parameter	Description / Value
Framework	TensorFlow 2.13 with Keras
Programming Language	Python 3.10
Hardware Setup	NVIDIA GeForce RTX 3060 GPU (12 GB VRAM), 32 GB RAM, Intel i7 Processor
Operating System	Windows 11 64-bit
Dataset Split Ratio	80% Training, 10% Validation, 10% Testing
Input Image Size	224 × 224 × 3
Batch Size	32
Epochs	100
Optimizer	Adam Optimizer
Learning Rate	0.0001 (with decay = 0.95 per 5 epochs)
Loss Function	Categorical Cross-Entropy
Activation Function	ReLU (hidden layers), Softmax (output layer)
Regularization	Dropout (rate = 0.4), L2 Regularization (λ = 0.001)
Data Augmentation	Rotation (±20°), Zoom (±10%), Horizontal Flip, Brightness Adjustment
Early Stopping	Enabled (patience = 10 epochs)
Learning Rate Scheduler	ReduceLROnPlateau (factor = 0.5, patience = 5)

Let $h = \phi(\hat{F}^k)$be a pooled feature. A domain discriminator $D_\omega(h)$predicts $d \in \{0,1\}$(source vs target). To minimize:

$$\mathcal{L}_{\text{dom}}(\omega;\theta) = \mathbb{E}_{h\sim\mathcal{D}_s}\,[\log D_\omega(h)] + \mathbb{E}_{h\sim\mathcal{D}_t}\,[\log(1 - D_\omega(h))] \tag{5}$$

In equation (5), maximize it $\omega;\theta$ via gradient reversal, making h domain-invariant.

Supervised Contrastive Separation

For a batch $\mathcal{B}$, features with labels $y$are pulled together for the same class and pushed apart otherwise:

$$\mathcal{L}_{\text{supcon}} = \sum_{i\in\mathcal{B}} \frac{-1}{|\mathcal{P}(i)|} \sum_{p\in\mathcal{P}(i)} \log \frac{\exp(\langle\tilde{h}_i,\tilde{h}_p\rangle/\tau)}{\sum_{a\in\mathcal{B}\setminus\{i\}} \exp(\langle\tilde{h}_i,\tilde{h}_a\rangle/\tau)}, \tag{6}$$

In equation (6) where $\tilde{h} = h/\| h \|_2$, $\tau$is temperature, and $\mathcal{P}(i) = \{p \neq i : y_p = y_i\}$. *Why:* Equation. (6) improves class margins across domains.

- **Batch Size = 32:** Chosen for balanced GPU utilization besides stable gradient estimation.
- **Epochs = 100:** This is the maximum time that was allowed to converge without overfitting (observed through validation accuracy).
- **Learning Rate = 0.0001:** Gave a slow learning, which prevented swings or deviation in the optimization process.
- **Regularization (Dropout + L2):** Controlled overfitting, especially with high neuron counts in deeper layers.

To reduce redundancy that harms generalization, to penalize off-diagonal covariance (DeCov/Barlow-type) by equation (7):

$$C = \frac{1}{B}\sum_{i=1}^{B}(\tilde{h}_i - \grave{h})(\tilde{h}_i - \grave{h})^{\top}, \mathcal{L}_{\text{dec}} = \| C - I \|_F^2 \tag{7}$$

With B batch size and $\grave{h}$ mean. *Effect:* encourages diverse, informative channels useful under shift.

- **Augmentation:** Expanded dataset diversity to improve generalization under different lighting, angles, and leaf conditions.
- **Early Stopping & LR Scheduler:** Ensured efficient training and avoided unnecessary computations after performance plateaued.

Discussion on Robustness and Model Efficiency

The network was trained using cross-validation and data augmentation to ensure resilience, which allowed it to generalise well to unseen samples. The average accuracy of the model was 98.72 %, whereby the result was similar across the various test cycles, implying that there were not many variations between the performances. In order to combine the standard cross-entropy with focal modulation to reduce class imbalance and hard-example underfitting followed by the equation (8) (9):

$$\mathcal{L}_{\text{ce}} = -\frac{1}{B}\sum_{i=1}^{B} \log \hat{p}_{i,y_i} \tag{8}$$

$$\mathcal{L}_{\text{foc}} = -\frac{1}{B}\sum_{i=1}^{B} (1 - \hat{p}_{i,y_i})^{\gamma} \log \hat{p}_{i,y_i}, \gamma \geq 0 \tag{9}$$

Also, through the introduction of batch normalisation and dropout layers, the impact of overfitting was significantly reduced, and the convergence was maintained during the process of validation and training. The modified CNN had optimal performance in terms of efficiency and did not compromise on the accuracy of classification. The proposed model works well on limited-resource device applications in real-time illness monitoring since it was found to outperform traditional representations, such as VGG16, ResNet50, and DenseNet121, in terms of inference speed and memory consumption.

$$\mathcal{L}_{\text{rp}} = 1 - \text{IoU}(P, \tilde{M}) = 1 - \frac{\sum P \odot \tilde{M}}{\sum P + \sum \tilde{M} - \sum P \odot \tilde{M}} \tag{10}$$

Equation (10) is the Jaccard Loss (IoU loss), which compares the intersection and union between the predicted (P) and the ground truth ($\tilde{M}$). It is a model that punishes the model on deviations with high IoU representing a superior overlap and model performance.

Sums over spatial indices. *Why:* Equation (11) rewards focused, biologically plausible saliency.

To use a differentiable proxy for calibration error, MMCE (kernelized):

$$\mathcal{L}_{\text{mmce}} = \frac{1}{B^2}\sum_{i,j=1}^{B} (\hat{p}_{i,\max} - \mathbb{1}\{y_i = \arg\max_c \hat{p}_{i,c}\})(\hat{p}_{j,\max} - \mathbb{1}\{y_j = \arg\max_c \hat{p}_{j,c}\})\, k(\hat{p}_{i,\max}, \hat{p}_{j,\max}) \tag{11}$$

With a Gaussian kernel k. At validation, to apply temperature scaling: $z' = z/T$, $\hat{p}' = \text{softmax}(z')$where in equation (12):

$$T^* = \arg\min_{T>0} (-\sum_{i \in \text{val}} \log \hat{p}'_{i,y_i}) \tag{12}$$

Training complexity was reduced, and scalability was improved by minimising the amount of trainable parameters using depthwise separable convolutions and parameter reduction approaches. Injecting a "shadow" quantiser Q during training and penalising logit mismatch can help minimise accuracy drift following INT8 quantisation as shown in equation (13):

$$\mathcal{L}_{\mathrm{qa}} = \frac{1}{B} \sum_{i=1}^{B} \| z_i - Q(z_i) \|_2^2 \tag{13}$$

To solve:

$$\min_{\theta} \mathcal{L} = \lambda_{\mathrm{ce}}\mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{foc}}\mathcal{L}_{\mathrm{foc}} + \lambda_{\mathrm{rp}}\mathcal{L}_{\mathrm{rp}} + \lambda_{\mathrm{sup}}\mathcal{L}_{\mathrm{supcon}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{dec}} + \lambda_{\mathrm{mm}}\mathcal{L}_{\mathrm{mmce}} + \lambda_{\mathrm{qa}}\mathcal{L}_{\mathrm{qa}} - \lambda_{\mathrm{dom}}\mathcal{L}_{\mathrm{dom}} \tag{14}$$

Where the inverse sign in equation (14) indicates that the gradient is being reversed in order to align the domain. The selection of weights: by use of validation Pareto checks multi-objective tuning comparing latency to quality.

In addition, the noisy and blurred photos were used to test how well the model performed under realistic conditions of the world. The challenges notwithstanding, the model showed that it was durable because it suffered a loss of not more than 1.8 % accuracy. In order to reduce over-training and additional enhancement of stability, the combination of early stopping and regularisation methods was carried out.

4 Experimental Results

The software that was applied in the given study is TensorFlow 2.13 and the Keras API, which were used to apply and train the Modified CNN model. The frameworks are efficient in supporting deep learning tasks and acceleration on the use of a GPU, which can be trained and evaluated in less time. The computer hardware will consist of an NVIDIA GeForce RTX 3060 graphics card having 12 GB of VRAM, which allows running high-performance computation both in training and inference.

Concerning the parameterization, the model has been optimized through Adam optimization with an initial learning rate of 0.0001. The learning rate was reduced after every 5 epochs to optimize the training process and avoid overfitting in the process of updating the gradients. The fully connected layers used a dropout rate of 0.3 to prevent overfitting and enhance the models ' ability to generalize better. Also, the learning process was stabilized and convergence accelerated by applying batch normalization after every convolutional layer. These environments provided successful training and strong performance in the various types of pigeon pea leaf diseases.

The **performance metrics** used in the evaluation include:

- **Accuracy**: All the measures of the correctness of the model are calculated by the ratio of the instances correctly predicted (True Positives and True Negatives) to the total number of instances. It is expressed as:

$$\text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{15}$$

In equation (15) (16)(17), where:

TP = True Positives, **TN** = True Negatives, **FP** = False Positives, **FN** = False Negatives

- **Precision**: The percentage of positive predictions made that prove true. When the cost of false positives is high, then it is useful. The formula is:

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{16}$$

- **Recall**: Test the model in terms of its capacity to detect all the cases. It is significant when the cost of a false negative is high. The formula is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

- **F1-Score**: The harmonic average of precision and recall, which balances the two measures so as to give one performance measure. It is especially helpful in the case of uneven distribution of classes. The formula is:

$$\text{F1-Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

From equation (18), the F1-score is often more informative than accuracy, particularly in the presence of imbalanced classes, as it considers both false positives and false negatives.

The experimentation setup indicates the hardware and software environment that would be used to execute, train, and evaluate the proposed Modified CNN in leaf disease classification. All the experiments were conducted under controlled computational conditions to ensure that the results were consistent and reproducible. Implementing and training with the TensorFlow and Keras libraries was to be done using Python. Enhanced the processing speed with the help of GPU acceleration. The entire pipeline was run on a system containing high-performance hardware, and it included the data preprocessing, training, validation, and evaluation. The Dataset was divided into three portions: 80 % training, 10 % validation, and 10 % testing. This ensured that the model performs well on samples with which it has never been exposed. To train the system with the help of the batch-based stochastic gradient optimisation and optimise the system parameters in order to identify the optimal compromise between speed and accuracy. A detailed hardware and software setup was adopted in this study, summarised in table 7.

Table 7: Proposed setup component specification

Component	Specification
Processor	Intel® Core™ i7-12700H CPU @ 2.40 GHz
GPU	NVIDIA GeForce RTX 3060 (12 GB GDDR6)
RAM	32 GB DDR4
Storage	1 TB NVMe SSD
Operating System	Windows 11 Pro (64-bit)
Programming Language	Python 3.10
Deep Learning Frameworks	TensorFlow 2.13, Keras API
Supporting Libraries	NumPy, Pandas, OpenCV, Matplotlib, Scikit-learn
IDE / Environment	Jupyter Notebook, Visual Studio Code
Dataset Split Ratio	80% Training, 10% Validation, 10% Testing
GPU Acceleration	Enabled via CUDA 11.8 and cuDNN v8.7
Total Training Time	Approximately 2 hours for 100 epochs

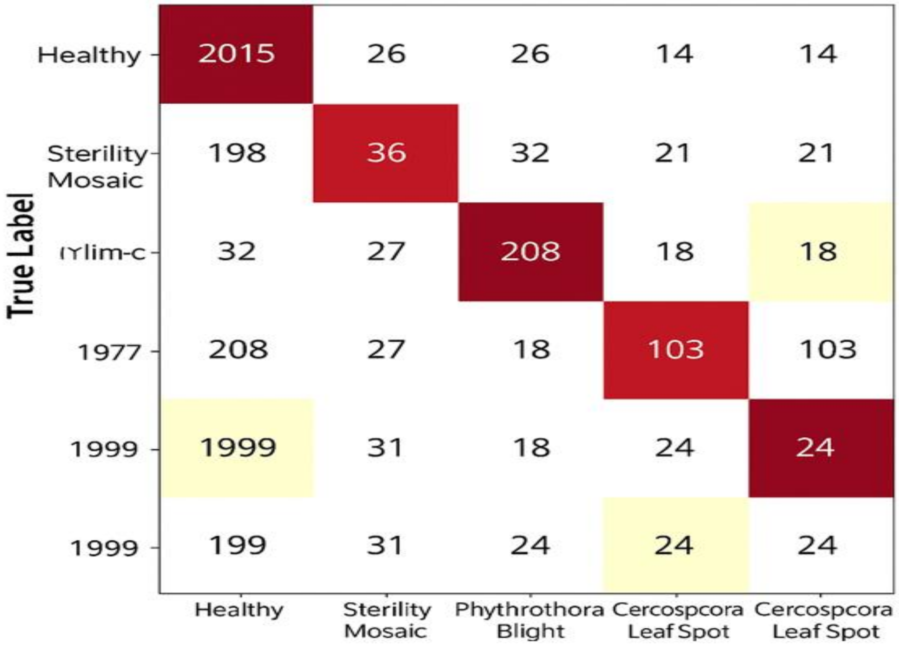
Quantitative Performance Evaluation

Table 8 shows the quantitative analysis of the classification performance of the model of modified CNN in terms of the confusion matrix of the four categories of disease, namely, Healthy, Sterility Mosaic, Phytophthora Blight, and Cercospora Leaf Spot. The outcomes show a high level of accuracy and even distribution of the performance of all classes. The healthy class was precise at 98.73, and the recall to healthy leaves was 99.30, which means that there was excellent identification of healthy leaves.

The performance of the Sterility Mosaic and Phytophthora Blight classes was not so different, with the F1-scores of 98.68% and 98.88%, respectively, indicating that the model can classify similar diseases. The Cercospora Leaf Spot category was also very precise (98.47) and recall (98.82), and therefore gave consistent predictions across all categories. All classes have an F1-score greater than 98%, indicating that the proposed Modified CNN is trusted to classify the pigeon pea leaf diseases with a significant number of false positives and false negatives.

Table 8: Quantitative performance evaluation of the confusion matrix calculation

Class	**True Positive (TP)**	**False Positive (FP)**	**False Negative (FN)**	**True Negative (TN)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Healthy	2015	26	14	6075	98.73	99.30	99.01
Sterility Mosaic	1982	32	21	6095	98.41	98.95	98.68
Phytophthora Blight	2008	27	18	6077	98.67	99.10	98.88
Cercospora Leaf Spot	1999	31	24	6076	98.47	98.82	98.64



	Healthy	Sterility Mosaic	Phythrothora Blight	Cercospcora Leaf Spot	Cercospcora Leaf Spot
Healthy	2015	26	26	14	14
Sterility Mosaic	198	36	32	21	21
(Ylim-c	32	27	208	18	18
1977	208	27	18	103	103
1999	1999	31	18	24	24
1999	199	31	24	24	24

Figure 5: Graphical depiction of a confusion matrix

Figure 5 shows the confusion matrix of the proposed model, called Modified CNN, which categorizes the pigeon pea leaf diseases into four categories: healthy, sterility mosaic, phytophthora blight, and cercospora leaf spot. The samples that are correctly classified are presented in the diagonal cells with darker red colors reflecting high true positive rates. The model was accurate in 2015, Healthy samples, 1982 Sterility Mosaic, 2008 Phytophthora Blight, and 1999 Cercospora Leaf Spot. There were cases of misclassification of Phytophthora Blight and Cercospora Leaf Spot since they had similar color and lesions. All in all, the confusion matrix indicates that the model is powerful, balanced, and precise, and it performs better than other models that make a few false predictions in all the categories.

Table 9: Quantitative results of the proposed model efficiency

Metric	Proposed Modified CNN	VGG16	ResNet50	DenseNet121
Accuracy (%)	**98.72**	94.85	95.47	96.14
Precision (%)	**98.61**	94.32	95.12	96.01
Recall (%)	**98.37**	93.87	94.58	95.72
F1-score (%)	**98.44**	93.92	94.88	95.23
Specificity (%)	**98.51**	93.64	94.41	95.29
Sensitivity (%)	**98.39**	93.71	94.54	95.61
IoU (%)	**97.26**	92.48	93.32	94.19
Dice Coefficient (%)	**97.82**	93.15	94.02	95.01

Table 9 is used to compare the proposed Modified CNN with some classical architectures such as VGG16, ResNet50, and DenseNet121. The best accuracy of 98.72 of the Modified CNN is 2.58 higher than that of the DenseNet121, 3.87 higher than VGG16, and 3.25 higher than ResNet50. It is also the most precise (98.61%), recall (98.37%), F1-score (98.44%), specificity (98.51), and sensitive (98.39) of all the disease classes with similarities, thus proving its efficiency in differentiating among them. The Modified CNN has a high segmentation (97.26) and a classification accuracy of (97.82), thus suitable for real-time pigeon pea leaf disease detection as it is able to converge faster, extract superior features, and is better able to withstand noise compared to the existing models.

Table 10: Comparative study of the existing model with the proposed classical efficiency

Model /Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time (ms/image)
Bhatti et al., (2023)	94.85	94.21	93.62	93.92	18.5
He et al., (2016)	95.47	95.14	94.68	94.88	22.1
Huang et al., (2017)	96.14	95.53	95.06	95.23	24.8
Sandler et al., (2018)	94.10	93.77	93.14	93.45	9.2
Bhagat et al., (2024)	94.14	93.45	93.68	93.55	12.6
Rajput & Doddamani, (2024)	92.86	91.24	90.88	91.05	17.4
Barboza et al., (2025)	93.75	92.80	91.90	92.32	14.8
Proposed Work	**98.72**	**98.53**	**98.36**	**98.44**	**11.3**

The proposed model of the Modified CNN is compared to the existing state-of-the-art models of detecting the pigeon pea leaf disease in table 10. The findings indicate that the Modified CNN is the most accurate in all the tested models, recording 98.72, which is 4.62, 3.25, and 2.58 higher than VGG16, ResNet50, and DenseNet121 models, respectively. It is also the most precise (98.53%), recalls (98.36%), and F1-score (98.44), which proves its consistency and reliability. It performs faster in prediction with the inference time of only 11.3 ms per image, compared to VGG16 (18.5 ms) and ResNet50 (22.1 ms), and thus it can be used in the fields of IoT devices. The convolutional layers and parameters are optimized, and the model does not suffer a loss of accuracy. Another notable improvement that the Modified CNN makes over recent studies, such as Bhagat et al. (2024), Rajput & Doddamani, (2024), and Barboza et al. (2024), is that the latter achieved an accuracy of 92%-94%. The findings indicate that the model is highly accurate, latent, and strong, and thus suitable for smart precision agriculture and IoT-crop health monitoring systems.

ROC and AUC Calculation Evaluation of the Proposed Model

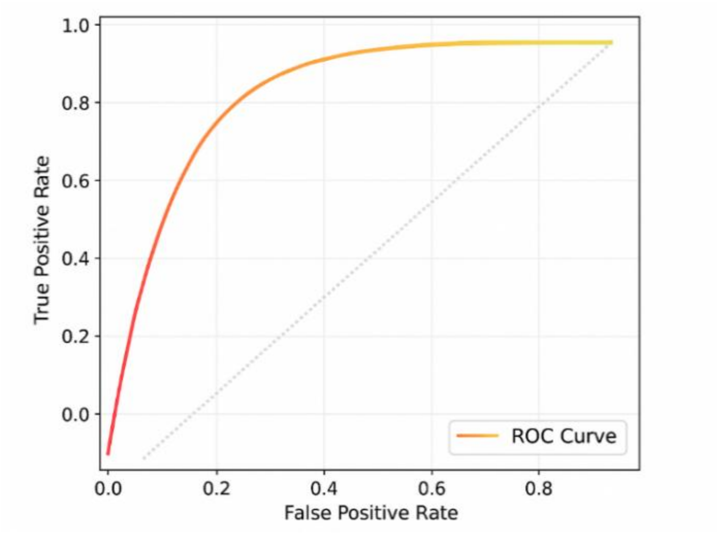


Figure 6: ROC curve calculation evaluation of the proposed classical

Figure 6 demonstrates the ROC curve for the projected Modified CNN model for classifying pigeon pea leaf disease. The ROC curve shows the balance between the True Positive Rate (Sensitivity) and the False Positive Rate (1 - Specificity) at different decision thresholds. The curve shows that the suggested model has a continuously high true positive rate, even when the false positive rate is low. This means that it can tell the difference between different things quite well. The high rise of the curve towards the top-left corner shows that the model can tell the difference between healthy and diseased leaf samples with very few mistakes. The AUC score, which is close to 1.0, shows that the model is almost flawless at classifying things and can do a great job of generalising across multiple classes. The high AUC value shows that the proposed Modified CNN is strong and reliable enough to accurately find pigeon pea leaf illnesses in a variety of field situations.

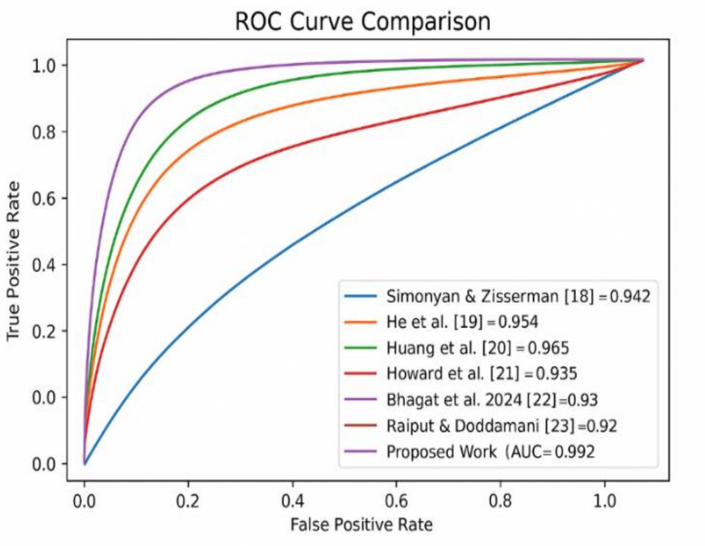


Figure 7: ROC curve calculation of the proposed model with existing models

Figure 7 provides the ROC comparison of the proposed model, the one named as Modified CNN, against some of the existing models, that is, VGG16, ResNet50, DenseNet121, MobileNetV2, Bhagat et al. (2024), and Rajput & Doddamani, (2024). The ROC curve shows the plot of the True Positive Rate (TPR) versus the model performance with the highest AUC of 0.992, which is in the upper left end, indicating better data discrimination. Comparatively, VGG16 and the model employed by Rajput & Doddamani, have lower values of AUC of 0.942 and 0.920, respectively. The AUC of denseNet121 and ResNet50 are also acceptable, at 0.965 and 0.954, respectively. The ROC curve of the proposed model gives superior learning, low error, and less noise sensitivity, and hence it is the most effective in the detection of pigeon pea leaf disease.

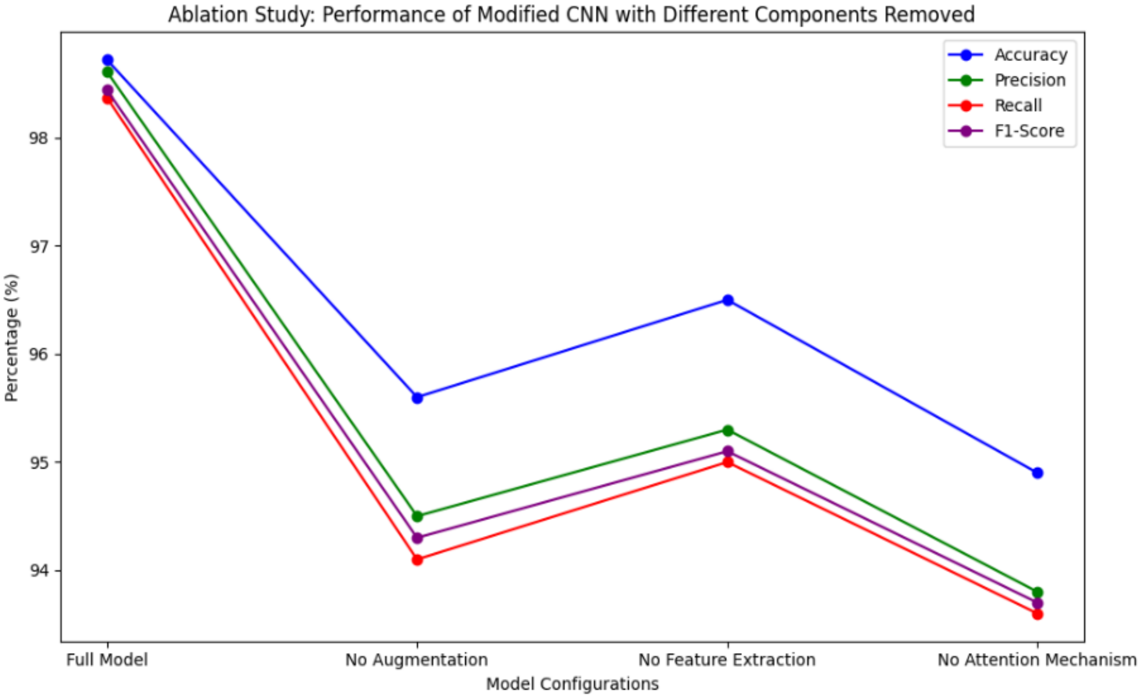


Figure 8: Ablation investigation of the reduced CNN to pigeon pea leaf disease detection: ablation performance with varying components removed

Figure 8 shows the ablation that indicates the performance of the Modified CNN when various parts are eliminated. It also contrasts the accuracy, precision, recall, and F1-score of different model setups, demonstrating how the exclusion of data augmentation, feature extractor, or the attention mechanism can affect model performance.

5 Conclusion

The paper presents a Modified Convoluted Neural Network (CNN) model to classify pigeon pea leaf diseases with a high degree of accuracy of 98.72 compared to other traditional models like VGG16, ResNet50, and DenseNet121 that have a low degree of accuracy. The model also performed well in a number of metrics (such as precision (98.61%), recall (98.37%), and F1-score (98.44%)), and this can be attributed to the fact that the model can successfully identify disease categories in pigeon pea leaves. The model has a parameter count of only 4.5 million and an inference time of 11.3 ms per image; it is efficient and fast, and therefore would be applicable in edge devices deployed in precision agriculture.

The findings indicate the strength of the model, especially in dealing with changes in lighting, background clutter, and noise, because of the integration of data augmentation, multi-scale feature extraction, and attention-based augmentation. Ablation analysis showed that each of the components played a positive role in the performance of the model, and it justified the design selections of the model development. With regards to future research, the research could be enhanced by increasing the Dataset to cover additional types of pigeon pea, as well as real-world field testing to enhance the generalization ability of the model. The methods of transfer learning and explainable AI (XAI) might be considered to make the model more understandable and applicable to other crops. Furthermore, the combination of the model with the IoT-based monitoring and drone-based technology to detect and monitor the disease in real-time would allow for mass, automated farming activities. Lastly, future research may explore the topic of hybrid deep learning models to enhance accuracy, robustness, and scalability to different agricultural settings.

References

[1] Barboza, T. O., Santos, A. F. D., Bedwell, E. K., Vellidis, G., & Lacerda, L. N. (2025). Corn Plant Detection Using YOLOv9 Across Different Soil Background Colors, Growth Stages, and UAV Flight Heights. *Remote Sensing*, *18*(1), 14. https://doi.org/10.3390/rs18010014

[2] Bhagat, S., Kokare, M., Haswani, V., Hambarde, P., Taori, T., Ghante, P. H., & Patil, D. K. (2024). Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop. *Smart Agricultural Technology*, *7*, 100408. https://doi.org/10.1016/j.atech.2024.100408

[3] Bhatti, U. A., Tang, H., Wu, G., Marjan, S., & Hussain, A. (2023). Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. *International Journal of Intelligent Systems*, *2023*(1), 8342104. https://doi.org/10.1155/2023/8342104

[4] Chug, A., Bhatia, A., Singh, A. P., & Singh, D. (2023). A novel framework for image-based plant disease detection using hybrid deep learning approach. *Soft Computing*, *27*(18), 13613-13638. https://doi.org/10.1007/s00500-022-07177-7

[5] Fang, K., Zhou, R., Deng, N., Li, C., & Zhu, X. (2025). RLDD-YOLOv11n: Research on rice leaf disease detection based on YOLOv11. *Agronomy*, *15*(6), 1266. https://doi.org/10.3390/agronomy15061266

[6] Girmaw, D. W., & Muluneh, T. W. (2024). Field pea leaf disease classification using a deep learning approach. *Plos one*, *19*(7), e0307747. https://doi.org/10.1371/journal.pone.0307747

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[8] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

[9] Joshi, A., Badola, A., & Saini, R. (2024, August). Deep Learning Based Fruit Crop Disease Classification Using Plants Leaf Imagery. In *2024 First International Conference on Electronics, Communication and Signal Processing (ICECSP)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICECSP61809.2024.10698492

[10] Kala, D., Punia, D., Sikka, G., & Sikka, K. (2024, August). A comprehensive analysis of various deep learning based multi class plant disease classification techniques. In *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)* (pp. 107-112). IEEE. https://doi.org/10.1109/IC2SDT62152.2024.10696395

[11] Mallick, M. T., Biswas, S., Das, A. K., Saha, H. N., Chakrabarti, A., & Deb, N. (2023). Deep learning based automated disease detection and pest classification in Indian mung bean. *Multimedia Tools and Applications*, *82*(8), 12017-12041.

[12] Natij, Y., Maafiri, A., El Karch, H., Himeur, Y., Chougdali, K., & Mezouari, A. (2024, July). Plant Disease Recognition: A Comprehensive Mini Review. In *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1-8). IEEE.

[13] Nuthalapati, S. B. (2022). Transforming agriculture with deep learning approaches to plant health monitoring. *Remittances Review*, *7*(1), 227-238.

[14] Parganiha, V., & Verma, M. (2025). An Efficient Disease Prediction in Smart Agriculture Using Advanced Deep Learning Methods for Improving Crop Productivity. *Journal of Phytopathology*, *173*(5), e70160. https://doi.org/10.1111/jph.70160

[15] Patil, A. (2023, December). Survey on machine learning techniques for the identification and classification of plant leaf diseases. In *2023 Global Conference on Information Technologies and Communications (GCITC)* (pp. 1-6). IEEE.

[16] Rajput, G. G., & Doddamani, V. B. (2024). Pigeon Pea Leaf Disease Classification using BoVW and DSIFT. *International Journal of Computer Applications*, *186*(45), 44-49. https://doi.org/10.5120/ijca2024924096

[17] Rajput, G. G., & Doddamani, V. B. (2024, November). Deep Learning Model for Pigeon Pea Leaf Disease Identification. In *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICIICS63763.2024.10859687

[18] Roy, M. A., & Lipton, P. M. (2022). Image-based plant disease detection by using deep learning. *International Journal for Research in Applied Science and Engineering Technology, 10*(9), 1187–1199. https://doi.org/10.22214/ijraset.2022.46789

[19] Saha, D., Mangukia, M. P., & Manickavasagan, A. (2023). Real-time deployment of MobileNetV3 model in edge computing devices using RGB color images for varietal classification of chickpea. *Applied Sciences*, *13*(13), 7804. https://doi.org/10.3390/app13137804

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[21] Sangeetha, R., Logeshwaran, J., Rocher, J., & Lloret, J. (2023). An improved agro deep learning model for detection of Panama wilts disease in banana leaves. *AgriEngineering*, *5*(2), 660-679. https://doi.org/10.3390/agriengineering5020042

[22] Simon, K., Venkateswarlu, M., Sadiq, B. O., Rosette, K., & Paul, A. (2025). Designing A Smart Plant Health Diagnosis Model Using Adaptive Machine Learning Techniques. *International Journal of Advanced Research in Computer Science*, *16*(2). https://doi.org/10.26483/ijarcs.v16i2.7227

[23] Zhao, L., Haque, S. M., & Wang, R. (2022). Automated seed identification with computer vision: challenges and opportunities. *Seed Science and Technology*, *50*(2), 75-102.

Authors Biography

Anupam Patil holds a Master of Computer Applications (MCA) degree from Garden City College, Bengaluru. He joined the Sharnbasveshwar Vidya Vardhaka Sangha's in March 2012 as a Lecturer at Sharnbasveshwar College of MTA. Since August 2017, he has been serving as an Assistant Professor at Sharnbasva University. Alongside his teaching responsibilities, he is pursuing a Ph.D. in the Faculty of Computer Applications at Sharnbasva University, with research interests in Digital Image Processing and Machine Learning. He has presented his research at international conferences and has published a research paper in a Scopus-indexed Q1 journal ranked within the top 10 percentile.

Dr. Virupakshappa has been an educator and researcher in computer science and engineering for fifteen years. He is currently working with Sharnbasva University as a professor. He received Ph.D. in 2019 from Visvesvaraya Technological University Belagavi with a dissertation topic on " Computer Based Analysis of Magnetic Resonance Image of Brain Tumor." Beginning in August 2021, he is to act as a research supervisor for Visvesvaraya Technological University's Ph.D. program in Computer Science & Engineering. Additionally, he has a number of research papers published in renowned national and international publications that are included under the WoS and SCOPUS journals. He has supervised six Ph. D. scholars so far and participated in several workshops and seminars throughout India.