

## Gradient-based stochastic depth with convolutional neural network for coconut tree leaf disease classification

Kavitha Magadi Gopalakrishna<sup>1</sup>, Raviprakash Madenur Lingaraju<sup>2</sup>, Ananda Babu Jayachandra<sup>3</sup>

<sup>1</sup>Department of Computer Science and Engineering, Kalpataru Institute of Technology, Visvesvaraya Technological University, Belagavi, India

<sup>2</sup>Department of Artificial intelligence and Machine Learning, Kalpataru Institute of Technology, Visvesvaraya Technological University, Belagavi, India

<sup>3</sup>Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India

### Article Info

#### Article history:

Received Mar 15, 2025

Revised Jan 20, 2026

Accepted Feb 6, 2026

#### Keywords:

Coconut tree disease  
Convolutional neural network  
Deep learning  
GrabCut segmentation  
Gradient magnitudes  
Gradient-based stochastic depth  
Inter- and intra-class variability

### ABSTRACT

The coconut palm (*Cocos nucifera*) is vital plantation crop, valued for their different uses, ranging from their fruit to its trunk. In recent times, it has been observed that many coconut trees are affected by diseases that reduce production and weaken the strength of the coconut. The classification of coconut leaf diseases is challenging because of intra-class and inter-class variability. This research introduces the gradient-based stochastic depth (GSD) with convolutional neural network (CNN) technique to coconut leaf disease classification to overcome these challenges. The GSD technique is incorporated into every layer of the CNN, where it calculates the probability using gradient magnitudes and skips layers that contribute minimally to the classification. The images are segmented using the GrabCut segmentation algorithm, which isolates the leaf from the background using graph-based segmentation, helping to differentiate between various disease classes. The GSD with CNN algorithm obtains an accuracy of 96.42%, precision of 96.15%, recall of 95.87%, and F1-score of 95.93%, while comparing with existing algorithms.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



### Corresponding Author:

Kavitha Magadi Gopalakrishna  
Department of Computer Science and Engineering, Kalpataru Institute of Technology  
Visvesvaraya Technological University  
Belagavi, India  
Email: kavitha.mg@kittiptur.ac.in

## 1. INTRODUCTION

In recent years, a significant decline in agricultural productivity has posed a threat to the global food supply [1]. The coconut tree is a vital plantation crop due to its diverse and useful products [2]. Globally, in many regions, the exceptional nutritional value and variety of coconut-based products have made it essential for daily life and numerous industries [3]. Despite the rising demand from consumers, many coconut farmers struggle to make ends meet due to high labor costs and the time-consuming process of producing coconut products [4]. Coconuts are a rich source of vitamins, minerals, and fiber, making them a healthy snack [5]. Additionally, they serve as an important source of nutrition and are widely used in the production of fuel, food, and medicines [6]. The coconut fruit, which develops on coconut palms, has been used for centuries, and humans have relied on it for their basic needs [7]. In recent times, people of various faiths and cultures have recognized coconut as a valuable medicinal and nutritional resource [8]. However, certain diseases that attack the coconut leaves weaken them over time, leading to crop losses. Farmers using conventional coconut farming techniques can only observe the lowest leaves of the plant crowns from the ground [9]. Conventional

imaging techniques, which capture photos from high altitudes over extensive areas, are both challenging and time-consuming [10]. This process requires manual involvement, which is costly, time-consuming, and prone to errors. Moreover, the trees are susceptible to various diseases that reduce their productivity and can ultimately lead to their death [11]–[13]. Early detection and diagnosis of these diseases are crucial for effective disease management and preventing further spread [14]. Machine learning (ML)-based algorithms are majorly utilized in agriculture to classify plants, trees, and crops, as well as to identify diseases [15]. Deep learning (DL)-based algorithms are effective neural networks for handling large datasets, helping to achieve better outcomes.

Kavithamani and UmaMaheswari [16] employed drones to capture images of damaged and healthy coconut palms. The images were analyzed for abnormal regions, such as symptoms of illness or whitefly infestation, using segmentation techniques. The prediction and classification of diseases focused on training a deep convolutional neural network (DCNN), which detected issues like root bleeding, blade pollution, and insect infestation by utilizing the segmented regions. The method quickly located abnormal boundaries using segmentation techniques. However, the method struggled to differentiate between the different disease classes due to limited number of layers in network. Megalingam *et al.* [17] suggested the modified Inception Net-based hyper tuning support vector machine (MIN-SVM) to classification of coconut tree, which relied on morphological elements such as inclination, height, and orientation. Features were extracted from pre-processed images by four different convolutional neural network (CNN) algorithms, included visual geometry group (VGG), ResNet, Inception Net, and MIN-SVM. These captured attributes were classified by SVM. The proposed method, which included the Inception Net feature extractor, yielded good results in classification. However, the suggested method does not enhance the contrast of the image, which limits the model's performance.

Thite *et al.* [18] introduced the MobileNetV2, ResNet50, and VGG16 architectures, that were renowned to its abilities in image recognition tasks. In the proposed method, the dataset was pre-processed, and data augmentation techniques were applied to improve generalization and fine-tune model to a specific classification task. Method achieved high classification performance with excellent accuracy. However, the method failed to eliminate noise in the images, which reduced the disease classification performance. Manoharan *et al.* [19] developed the YOLOv9 method for identifying macro and micronutrient deficiencies on coconut trees and introduced the image analysis-based severity detection (IASD) for assessing severity of these deficiencies. The severity index calculation model (SICM) was used to calculate severity index (SI) of the deficiencies. To every identified deficiency, relevant fertilizer and their application quantity were analyzed. The developed method utilized various augmentation and pre-processing algorithms to improve image quality, maximize count of images, and minimize overfitting. However, the method had limited feature representation, which hindered model's ability to differentiate among the various disease classes.

Divyanth *et al.* [20] presented Faster region-convolutional neural network (Faster R-CNN) to detecting coconut clusters, both non-occluded and those with leaf occlusion. To enhance accuracy, the attention mechanism was implemented in Faster R-CNN method. Presented method provided essential data for harvesting coconuts effectively and safely. However, the method had lower classification performance due to limited feature representation. From the above analysis, the existing algorithms have limitations, such as struggling to differentiate between different disease classes, not enhancing the contrast of the image, failing to eliminate noise, and having limited feature representation and classification performance due to issues like intra-class variability and inter-class similarity.

To mitigate these limitations, this article develops gradient-based stochastic depth (GSD) with CNN approach, which effectively classifies the different disease classes. Incorporating the GSD in the CNN method skips the less contributing layers, minimizing training time while achieving high classification performance. In the pre-processing phase, the contrast adaptive histogram equalization (CLAHE) and Gaussian blur techniques are used to enhance image contrast and eliminate noise, respectively. Through these processes, the issues of inter-class similarity and intra-class variability are minimized, and the representation of features for disease classification is enhanced. The contributions of this research are outlined as follows: CLAHE and Gaussian blur techniques are employed in the pre-processing stage to enhance color contrast and remove noise from the images. The GrabCut segmentation algorithm is used to segment the images, isolating the leaf portions from the background and improving the visibility of the disease, which helps differentiate between the various disease classes. GSD with CNN approach is developed to classify various disease classes with high accuracy and reduced training time. The GSD technique is incorporated into each layer of the network, which skips less efficient layers by calculating probabilities using gradient magnitudes. This process minimizes training time and enhances classification performance with higher accuracy.

This manuscript is organized as follows: section 2 shows process of proposed technique. Section 3 discusses results of the proposed technique. Section 4 concludes a manuscript.

## 2. METHOD

An efficient DL model is developed to classify various disease classes with high classification performance. Dataset utilized on this work is the coconut tree disease dataset, and the images are pre-processed using the CLAHE and Gaussian blur techniques, which enhance image contrast and remove noise. Pre-processed images are then segmented by GrabCut segmentation technique, which segments the images effectively through an iterative process. Finally, the segmented images are classified using the developed GSD-CNN algorithm, achieving high classification performance. Figure 1 illustrates the process of coconut tree disease classification.

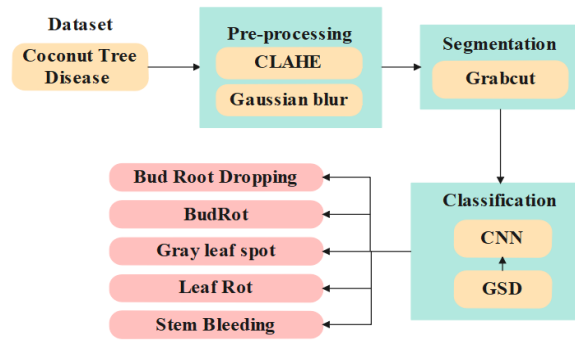


Figure 1. Process of coconut tree disease classification

### 2.1. Data collection

Data present in coconut tree disease dataset is gathered from farms in the districts of Pune, India, Taluka-Shirur, and Kendur. The images are captured on different scenarios, included leaves in its natural environment and detached from the plant. The dataset includes 5798 images, with every class containing varying number of images [21]. Table 1 presents the dataset description.

Name of disease	Number of samples
Bud rot	470
Bud root dropping	514
Stem bleeding	1006
Leaf rot	1673
Gray leaf spot	2135
Total	5798

### 2.2. Pre-processing

#### 2.2.1. Image denoising

This is a significant pre-processing phase for reducing distortions or noise in the image [22]. Conventional denoising techniques focus on smoothing images through applying unequal weights to pixels, which are inversely proportional to their distance from central pixel on image. Particularly, the Gaussian filter is linear smoothing filter which minimizes the weights applied to pixels as the distance from the central pixel increases, depended on Gaussian function. Mathematical formula for an input pixel in the Gaussian filter is given in (1). In the (1), the  $d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$  respective pixel distance from central pixel.

$$g(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}} \quad (1)$$

#### 2.2.2. Contrast adaptive histogram equalization

CLAHE is introduced to provide a more natural appearance for enhancing images. Additionally, the threshold element is utilized to limit contrast enhancement to chosen areas. To obtain this, the RGB color space is converted to hue, saturation, and value (HSV). The value component is processed through CLAHE, while the hue and saturation components remain unchanged. Finally, CLAHE is employed through redistributing the grey levels back to the RGB color space.

### 2.3. Segmentation

GrabCut segmentation approach is enhancement of GraphCut technique. GrabCut technique uses texture (color) and boundary (contrast) data in images, allowing interaction with users by rectangular boxes. It evaluates pixels within box and classifies pixels outside box as background. After several iterations, algorithm acquires the desired segmentation [23]. The conventional GrabCut technique performs iterative optimization in segmentation, simplifying the user interaction required to segment objects. It utilizes rectangular bounding box to evaluate pixels inside box and background pixels outside. Energy function  $E$  of GrabCut is given in (2). The  $U$  function defines regional information term of energy function,  $\theta$  denotes histogram model,  $\alpha$  is the transparency coefficient, and  $z$  denotes a single pixel. The mathematical formula for  $U$  is given in (3). The mathematical formula for smoothing item  $V(\alpha, z)$  is given as (4).

$$E(\alpha, \theta, z) = U(\alpha, \theta, z) + V(\alpha, z) \quad (2)$$

$$U(\alpha, \theta, z) = \sum_n -\log h(z_n; \alpha_n) \quad (3)$$

$$V(\alpha, z) = \gamma \sum_{(m,n) \in C} dis(m, n)^{-1} [\alpha_n \neq \alpha_m] \exp - (z_m - z_n)^2 \quad (4)$$

In (4),  $C$  represents a group of adjacent pixel pairs, the  $dis()$  represents Euclidean distance among adjacent pixels and choosing  $\beta$  ensured correct switching among high and low contrast on exponential direction that promoted the consistency on the same grayscale areas. While constant  $\beta = 0$ , smoothness term ensures smoothness everywhere defined through constant. This energy ensures coherence in areas with the same grey-level. Practically, better outcomes are attained through determining the pixels of neighboring, whether adjacent horizontally or vertically. The segmentation process of GrabCut image is described as:

- i) Initialization – user attains the initial trimap image through directly choosing target, where entire pixels outside box are considered background pixels, and entire pixels inside box are taken as target pixels  $T_U$ . When  $n \in T_B$ , pixel label  $\alpha_n = 0$ , and when  $n \in T_U$ , the pixel label  $\alpha_n = 1$ . Gaussian mixture model (GMM) to foreground and background is initiated with  $\alpha_n = 0$  and  $\alpha_n = 1$ . Pixels are next clustered into  $k$  classes, corresponding to  $k$  Gaussian models in GMM, utilizing K-means technique. Process of iterative reduction is described.
- ii) Employ components of Gaussian in GMM for every pixel. For every  $n$  in  $T_U$ ,  $k_n = \underset{k_n}{\operatorname{argmin}} D_n(\alpha_n, k_n, \theta, z_n)$ .
- iii) For given image data  $Z$ , iteratively optimize GMM parameters  $\theta = \underset{\theta}{\operatorname{argmin}} U(\alpha, k, \theta, z)$ .
- iv) Evaluate segmentation-relied on Gibbs energy term by evaluating the graph and measuring weights t-links and n-links. Next, max-flow/min-cut technique is utilized to segmentation.
- v) Repeat process until convergence occurs.
- vi) Utilize border cutout and matching for smoothing and post-processing in segmented boundaries.

### 2.4. Classification

Segmented images are fed to the classification stage to classify the various classes of coconut leaf diseases. In this phase, the GSD technique is included in CNN, skipping the layer that less contributes to coconut leaf disease classification and improves classification performance. This section explains the process of CNN and GSD techniques.

#### 2.4.1. Convolutional layers

Convolutional layer is a primary component of CNN which includes a group of individual filters. Every filter is individually convolved to image and feature map acquired [24], [25]. Generally, the size of the image  $M \times N$  is convolved with a filter size of  $w \times h$ , where the mathematical formulae for output feature map size is given in (5) and (6).

$$o_w = \frac{M-w+2p_w}{s_w} + 1 \quad (5)$$

$$o_h = \frac{N-h+2p_h}{s_h} + 1 \quad (6)$$

Where  $p_w$  and  $p_h$  defines padding of zeros on width and height, respectively. The  $s_w$  and  $s_h$  are strides in horizontal and vertical directions. Each resulting feature map is achieved through convolution of input maps to linear filter, including bias term, and next non-linear function is applied. Mathematical formula for output is generally given in (7).

$$X_j^l = f \left( \sum_{i \in I_j} X_i^{l-1} \times W_{ij}^l + b_j^l \right) \quad (7)$$

Where  $l$  is the number of layers,  $W_{ij}$  denotes a convolutional kernel,  $b_j$  denotes bias,  $I_j$  denotes group of input maps, and  $f(\cdot)$  is activation function.

Activation function is crucial on CNNs, enabling them to learn and process complex tasks. The activation function is a non-linear transformation employed to input. It generally determines whether data received by neuron is relevant enough to provide output or if it should be ignored. In this work, rectified linear unit (ReLU) is used, as it allows for easy backpropagation of errors and enables many layers of neurons to be activated through ReLU function. Primary benefits of ReLU process across other activation functions is that it doesn't activate entire neurons simultaneously. This means that, in any given time, only subset of neurons is activated, makes network sparse, efficient, and easier to compute. For negative inputs, gradients are zero, and weights are not updated in backpropagation. That lead to dead neurons that never get activated, and its mathematical formula is given in (8) and (9).

$$f(X) = \max(0, X) \quad (8)$$

$$f(X) = \begin{cases} X, & X \geq 0 \\ 0, & X < 0 \end{cases} \quad (9)$$

Initial convolutional layer captures various low-level features, like edges and textures. By stacking multiple convolutional layers, network learns more global features. Here, the GSD layer is applied to each convolutional layer to improve feature representation.

#### 2.4.2. Pooling layers

The pooling layer is included among consecutive convolutional layers in a CNN. Purpose of pooling layer is to minimize spatial size of representation. This process minimizes number of parameters and computational cost on model, which helps in handling overfitting. Pooling layer process on each input layer individually, resizing it spatially through pooling process. The most common form of the pooling layer, to filter size of  $2 \times 2$  and stride of 2, down-samples each depth slice of input across the width and height of activations. In this work, the max pooling layer is used, as it provides good results and reduces computational cost for the upper layers through removing non-maximal values, offering form of translation invariance. Additionally, it offers extra robustness for positional variations and effectively reduces the dimensionality of intermediate representations.

#### 2.4.3. Fully connected layers

This means that neuron in prior layer is integrated to each neuron on present layer. Its activations are calculated through matrix multiplication, followed through bias offset. Neurons in final fully connected layer correspond to amount of classes for prediction. Many of attributes from convolutional and subsampling layers are useful to classification, and integration of these features plays a key role. Fully connected layer integrates entire attributes extracted from prior convolutional and subsampling layers. Last fully connected layer utilizes SoftMax activation function, which is primarily used for multi-class classification.

#### 2.4.4. Stochastic depth technique

Shallow models enable quick forward and backward data propagation, allowing them to be trained in shorter time. Though, because of complex spatial distributions of pixels and multi-scale semantic data, shallow networks are insufficient to learning and capturing primary features. During the training process, issues such as long training times, data attenuation during forward propagation, and vanishing gradients arise. To overcome these challenges, a stochastic depth layer is incorporated into the CNN. Using a shallow network during training improves data transfer within the network, resulting in a deeper network with enhanced generalization capabilities. The network with random depth is achieved through including stochastic depth drop block, that is same to dropout process. Mathematical formula for inverted residual architecture is given in (10), where a random variable  $b_i$  follows Bernoulli distribution for inverted residual process  $F$ . In the (10),  $b_i = 1$ , where the fully inverted residual architecture is retained, while  $b_i = 0$  is when the residual architecture is not activated, while the whole architecture is degenerated for identifying the process, and its mathematical formula is given as (11). Where  $b_i = 0$  with probability of  $1 - p$  and  $b_i = 1$  with probability of  $p$ , which represents surviving probability of a residual module. Mathematical formula for the linear decay function  $p_i$ , as given in (12).

$$X_{i+1} = F b_i(X_i, W_i) + H(X_i) \quad (10)$$

$$X_{i+1} = H(X_i) \quad (11)$$

$$p_i = 1 - \frac{i}{l}(1 - p_i) \quad (12)$$

This setting allows shallow layers to capture low-level features to utilize in later layers, with the shallow layers not being discarded frequently. Because of random dropping scheme, certain layers on network are not activated in every training iteration, which efficiently develops ensemble of inexplicit approaches. Training process to random depths combines residual methods at various depths, randomly extracting deep residual features from shallow residual features. This ensures feature diverseness, reduces overfitting, and enhances generalization capability of method. Network's training time, when employing stochastic depth, no longer maximizes to network depth but is based on expected network depth. In training, *ith* inverted residual block have probability of inactive, making efficient inverted residual block random variable. Mathematical formula for this is given in (13). In linear decay rule,  $p_i = 0.5$  and for sufficient large  $l$ , count of activated residual blocks in training increases.

$$E(L) = \sum_{i=1}^l P_i \quad (13)$$

#### 2.4.5. Gradient-based stochastic depth technique

The GSD skips layers in the training process based on their gradient magnitude, making a more informed and adaptive technique when compared to the traditional stochastic depth layer. The detailed steps of this process are summarized in Algorithm 1. Each convolutional layer is evaluated based on its gradient magnitude, which calculates its contribution to the loss function. During the backward passing,  $G_i = \|\nabla_{w_i}\|_2$ , where  $\nabla_{w_i}$  represents the loss gradients corresponding to the layer parameters. The decision of skipping is performed dynamically; if  $G_i$  is less than the threshold, the layer is skipped with a probability  $P(G_i)$ , which decreases as  $G_i$  increases.

The CNN suffers from vanishing gradients and long training times. To address these problems, stochastic depth is introduced as a regularization method that randomly skips some layers during the training process. Conventional stochastic depth involves random skipping with a fixed probability. This random skipping discards layers without considering their significance, which may eliminate important layers and reduce performance. Additionally, the fixed probability remains static and does not adapt to the learning process. To mitigate these challenges, the GSD is proposed, which dynamically decides whether to skip a layer based on its gradient magnitude. The GSD is applied to convolutional layers, selectively skipping layers during the training process.

Algorithm 1. CNN with GSD

Input: Input images

Output: Classified disease type

Images are pre-processed using CLAHE and Gaussian blur technique.

GrabCut segmentation is used to isolate coconut leaves

Features are extracted using CNN with GSD

- For every layer,  $\|gt\|$  is measured
- Dynamic threshold  $T$  is executed
- The skip layer  $l$  with probability  $P_{skip}(l)$  is used for further processing

The output is fed to the fully connected layers for classification. Loss and backpropagation are executed. The weights are updated and the skipping probabilities are adjusted dynamically.

### 3. EXPERIMENTAL ANALYSIS

The developed algorithm is simulated in a python environment and the required configurations being i5 processor, windows 10 (64 bit) and 8 GB RAM. Performance of developed algorithm is validated in terms of various metrics including, F1-score, precision, recall, and accuracy. Mathematical formula for metrics is given from (14) to (17).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (16)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100 \quad (17)$$

In Table 2, performance of developed algorithm is validated using various classes of datasets with different metrics. The classes namely, bud rot, but root dropping, stem bleeding, leaf rot, and gray leaf spot are evaluated in Table 2. The developed method obtains an average accuracy of 96.42%, 96.15% of precision, 95.87% of recall, and 95.93% of F1-score on class-wise results. Table 3 presents the performance of GrabCut-based segmentation technique, as evaluated on the coconut tree disease dataset. The various existing algorithms used to evaluate the GrabCut technique are: K-means clustering, active contour fields, conditional random fields, and graph-based segmentation algorithms. The GrabCut segmentation algorithm obtains 96.42% accuracy, 96.15% precision, 95.87% recall, and 95.93% F1-score.

Table 2. Performance of class-wise results

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bud rot	96.20	96.00	95.70	95.80
Bud root dropping	96.35	96.10	95.80	95.85
Stem bleeding	96.50	96.20	95.90	95.90
Leaf rot	96.45	96.25	96.00	96.00
Gray leaf spot	96.60	96.20	95.95	96.10
Average	96.42	96.15	95.87	95.93

Table 3. Performance of segmentation algorithms

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
K-means clustering	93.29	93.02	92.93	92.99
Active contour fields	93.87	93.54	93.18	93.32
Conditional random fields	94.08	93.85	93.45	93.65
Graph-based segmentation	94.57	94.28	93.78	93.96
GrabCut	96.42	96.15	95.87	95.93

Table 4 presents CNN performance without the use of GSD technique on the coconut tree disease dataset. The different existing algorithms used to evaluate the CNN without GSD technique are: extreme learning machines (ELM), CapsNet, fully connected neural network (FCNN), and multi-layer perceptron (MLP). Conventional CNN attains 95.27% accuracy, 95.01% precision, 94.78% recall, and 94.92% F1-score, outperforming the existing approaches. In Table 5, performance of the CNN with GSD technique is validated using different metrics on the coconut tree disease dataset.

Table 4. Performance CNN without GSD model

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
ELM	93.53	93.38	93.19	93.22
CapsNet	93.89	93.66	93.57	93.61
FCNN	94.21	94.05	93.88	93.91
MLP	94.75	94.53	94.32	94.42
CNN	95.27	95.01	94.78	94.92

Table 5. Performance of CNN with GSD technique

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CapsNet	95.02	94.86	94.59	94.65
FCNN	95.31	95.18	94.83	94.92
MLP	95.75	95.52	95.37	95.42
CNN	96.06	95.84	95.64	95.77
GSD-CNN	96.42	96.15	95.87	95.93

The performance of the CNN with GSD technique is compared with several existing algorithms, including CapsNet, FCNN, MLP, and traditional CNN. The GSD with CNN algorithm achieved 96.42% accuracy, 96.15% precision, 95.87% recall, and 95.93% F1-score when compared to these existing algorithms. Figure 2 shows accuracy vs. epochs graph for the developed algorithm and Figure 3 shows loss

vs. epochs graph for the developed algorithm. Table 6 presents the k-fold validation results for the proposed algorithm and Table 7 presents the performance of the pre-processing techniques.

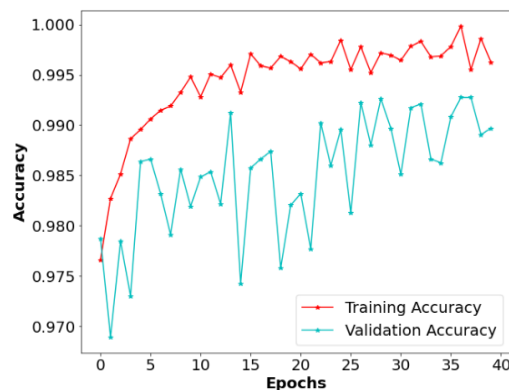


Figure 2. Accuracy vs epochs graph for the developed algorithm

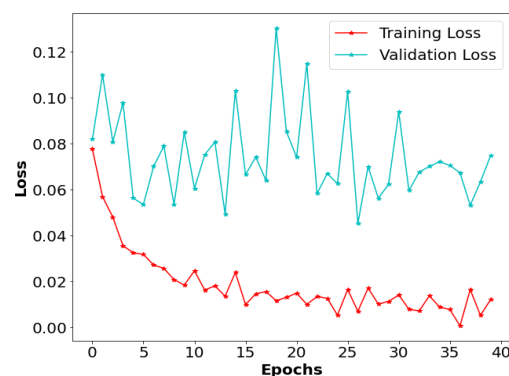


Figure 3. Loss vs epochs graph for the developed algorithm

Table 6. Performance of different K-fold values

K-values	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
K =2	93.02	92.78	92.47	92.15
K =3	93.67	93.37	93.02	92.78
K =4	94.33	94.05	93.79	93.23
K =5	96.42	96.15	95.87	95.93

Table 7. Performance of pre-processing techniques

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Median filter	94.89	94.56	94.27	94.40
Bilateral filter	95.28	95.02	94.68	94.93
Wavelet-based contrast enhancement	95.67	95.21	95.05	95.17
Anisotropic diffusion	96.03	95.87	95.43	95.65
CLAHE and Gaussian blur	96.42	96.15	95.87	95.93

### 3.1. Comparison analysis

In this section, performance of developed GSD-SNN algorithm is compared to existing algorithms like DCNN [16], MIN-SVM [17], and ResNet-50 [18] on coconut tree leaf images. The GSD with CNN algorithm achieved 96.42% accuracy, 96.15% precision, 95.87% recall, and 95.93% F1-score when compared to these existing algorithms. The GSD technique is incorporated into each layer of the CNN network, which skips the less efficient layers by calculating probabilities using gradient magnitudes. This process minimizes training time and improves classification performance with high accuracy. Table 8 presents comparative analysis of developed model.

Table 8. Comparative analysis of developed model

Methods	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DCNN [16]	Coconut tree leaf images	95.71	NA	NA	NA
MIN-SVM [17]		95.35	95	95	95
ResNet-50 [18]		94	92.25	93.42	93.88
Proposed GSD-CNN		96.42	96.15	95.87	95.93

### 3.2. Discussion

The performance of the developed algorithm is evaluated on coconut tree disease dataset using different metrics. Performance of developed algorithm is assessed based on class-wise results, using segmentation-based methods and classification algorithms. Additionally, the performance of developed algorithm is compared with existing algorithms: DCNN [16], MIN-SVM [17], and ResNet-50 [18]. The existing algorithms have limitations as noted further: struggles in differentiating different disease classes, no enhancement of image contrast, failure to eliminate image noise, lower representation of feature and classification performance because of issues like intra class variability and inter class similarity. To mitigate these limitations, this research proposes GSD approach with CNN method, which is classified into different disease classes effectively. By incorporating the GSD in the CNN method, the less contributing layer is skipped, while the training time is minimized with a high classification performance. In the pre-processing phase, the CLAHE and Gaussian blur techniques are used to enhance image contrast and eliminate noise, respectively. Through these processes, the issue of inter-class similarity and intra-class variability are eliminated and the representation of features for disease classification are enhanced. The GSD with CNN algorithm obtains accuracy of 96.42%, precision of 96.15%, recall of 95.87%, and F1-score of 95.93% while comparing with existing algorithms. High performance of proposed GSD-CNN is attributed to their adaptive layer-skipping mechanism depended on gradient magnitudes, that preserves high informative feature representations when minimizing redundant computations. Unlike traditional stochastic depth, gradient-aware skipping strategy prevents elimination of essential layers, by enhancing feature discrimination across inter-class variability and intra-class variability. Moreover, combination of CLAHE and Gaussian blur improves disease visibility and noise suppression, when GrabCut segmentation ensures learning on disease-affected regions, these improves classification accuracy. Rather than its strong performance, proposed model shows misclassification, especially where disease symptoms overlapping visual patterns with subtle texture variations. Same color and structural characteristics among certain disease classes like leaf rot and gray leaf spot leads to classification ambiguity.

## 4. CONCLUSION

This research develops a DL-based algorithm for coconut tree leaf disease classification. The coconut tree disease dataset used includes disease-labeled images. The images in dataset are pre-processed by CLAHE and Gaussian blur techniques, which enhance image contrast and eliminate noise, which helps differentiate different disease classes. Then, the images are segmented using GrabCut algorithm, which isolates the leaf images from the background through an iterative process to enhance feature representation. Finally, classification is performed using the developed GSD-CNN technique, which extracts hierarchical features through the convolutional layers and classifies the different disease classes with high classification performance. The GSD technique is incorporated into every layer of the CNN, skipping layers that contribute less to classification and minimizing the model's training time while maintaining high classification accuracy. The GSD with CNN algorithm achieved 96.42% accuracy, 96.15% precision, 95.87% recall, and 95.93% F1-score when compared to existing algorithms. In future work, other DL-based models can be explored to further enhance classification performance.

### ACKNOWLEDGMENTS

We would like to grateful to our Colleague's for their assistance with data collection and for their insightful discussions.

### FUNDING INFORMATION

Authors state no funding involved.

### AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kavitha Magadi	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Gopalakrishna														
Raviprakash Madenur		✓				✓		✓	✓	✓	✓	✓		
Lingaraju														
Ananda Babu	✓		✓	✓			✓			✓	✓		✓	✓
Jayachandra														

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at <https://doi.org/10.17632/gh56wbsnj5.1>, reference number [21].





## REFERENCES

- [1] R. Hayati, A. A. Munawar, E. Lukitaningsih, N. Earlia, T. Karma, and R. Idroes, "Combination of PCA with LDA and SVM classifiers: a model for determining the geographical origin of coconut in the coastal plantation, Aceh Province, Indonesia," *Case Studies in Chemical and Environmental Engineering*, vol. 9, 2024, doi: 10.1016/j.cscee.2023.100552.
- [2] R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, and S. Ruparelia, "Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform," *The Visual Computer*, vol. 38, no. 8, pp. 2923–2938, 2022, doi: 10.1007/s00371-021-02164-9.
- [3] L. Liu, W. Yan, and B. Liu, "Transcriptome sequencing of *Cocos nucifera* leaves in response to *Rhynchophorus ferrugineus* infestation," *Frontiers in Genetics*, vol. 14, 2023, doi: 10.3389/fgene.2023.1115392.
- [4] L. Lu *et al.*, "Integrated transcriptomic and metabolomic analyses reveal key metabolic pathways in response to potassium deficiency in coconut (*Cocos nucifera* L.) seedlings," *Frontiers in Plant Science*, vol. 14, 2023, doi: 10.3389/fpls.2023.1112264.
- [5] A. Sunpapao *et al.*, "Morphological and molecular identification of plant pathogenic fungi associated with dirty panicle disease in coconuts (*Cocos nucifera*) in Thailand," *Journal of Fungi*, vol. 8, no. 4, 2022, doi: 10.3390/jof8040335.
- [6] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas, "Machine learning and deep learning for plant disease classification and detection," *IEEE Access*, vol. 11, pp. 114352–114377, 2023, doi: 10.1109/ACCESS.2023.3324722.
- [7] S. K. Brar, R. Sharma, S. Vats, and V. Kukreja, "A smart approach to coconut leaf spot disease classification using computer vision and deep learning technique," in *2023 World Conference on Communication & Computing*, 2023, pp. 1–6, doi: 10.1109/WCONF58270.2023.10235251.
- [8] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A lightweight attention-based convolutional neural networks for tomato leaf disease classification," *Agriculture*, vol. 12, no. 2, 2022, doi: 10.3390/agriculture12020228.
- [9] K. Jaspin, S. Selvan, P. S. Packianathan, and P. Kumar, "Convolution neural network based classification of plant leaf disease images," in *Applications of Artificial Intelligence and Machine Learning*, Singapore: Springer, 2022, pp. 511–527, doi: 10.1007/978-981-19-4831-2\_42.
- [10] P. R. D. Silva, C. N. Perera, B. W. Bahder, and R. N. Attanayake, "Nested PCR-based rapid detection of phytoplasma leaf wilt disease of coconut in Sri Lanka and systemic movement of the pathogen," *Pathogens*, vol. 12, no. 2, 2023, doi: 10.3390/pathogens12020294.
- [11] H. D. Mafukidze, G. Owomugisha, D. Otim, A. Nechibvute, C. Nyamhere, and F. Mazunga, "Adaptive thresholding of CNN features for maize leaf disease classification and severity estimation," *Applied Sciences*, vol. 12, no. 17, 2022, doi: 10.3390/app12178412.
- [12] W. Li, L. Zhu, and J. Liu, "PL-DINO: an improved transformer-based method for plant leaf disease detection," *Agriculture*, vol. 14, no. 5, 2024, doi: 10.3390/agriculture14050691.
- [13] J. Li *et al.*, "Genome-wide identification, classification and expression analysis of MYB gene family in coconut (*Cocos nucifera* L.)," *Frontiers in Plant Science*, vol. 14, 2024, doi: 10.3389/fpls.2023.1263595.
- [14] A. Palananda and W. Kimpan, "Classification of adulterated particle images in coconut oil using deep learning approaches," *Applied Sciences*, vol. 12, no. 2, 2022, doi: 10.3390/app12020656.
- [15] Y. Chen *et al.*, "DFCANet: a novel lightweight convolutional neural network model for corn disease identification," *Agriculture*, vol. 12, no. 12, 2022, doi: 10.3390/agriculture12122047.
- [16] V. Kavithamani and S. UmaMaheswari, "Investigation of deep learning for whitefly identification in coconut tree leaves," *Intelligent Systems with Applications*, vol. 20, 2023, doi: 10.1016/j.iswa.2023.200290.
- [17] R. K. Megalingam, S. K. Manoharan, D. H. T. A. Babu, G. Sriram, K. Lokesh, and S. K. Sudheesh, "Coconut trees classification based on height, inclination, and orientation using MIN-SVM algorithm," *Neural Computing and Applications*, vol. 35, no. 16, pp. 12055–12071, 2023, doi: 10.1007/s00521-023-08339-w.
- [18] S. Thite, Y. Suryawanshi, K. Patil, and P. Chumchu, "Coconut (*Cocos nucifera*) tree disease dataset: a dataset for disease detection and classification for machine learning applications," *Data in Brief*, vol. 51, 2023, doi: 10.1016/j.dib.2023.109690.





- [19] S. K. Manoharan, R. K. Megalingam, G. A. G. Jogesh, A. K. and A. R. Kunnambath, "AI based early identification and severity detection of nutrient deficiencies in coconut trees," *Smart Agricultural Technology*, vol. 9, 2024, doi: 10.1016/j.atech.2024.100575.
- [20] L. G. Divyanth, P. Soni, C. M. Pareek, R. Machavaram, M. Nadimi, and J. Paliwal, "Detection of coconut clusters based on occlusion condition using attention-guided faster R-CNN for robotic harvesting," *Foods*, vol. 11, no. 23, 2022, doi: 10.3390/foods11233903.
- [21] K. Patil, S. Thite, Y. Suryawanshi, and P. Chumchu, "Coconut tree disease dataset," *Mendeley Data*, VI. 2023, doi: 10.17632/gh56wbsnj5.1.
- [22] A. Sharma and P. K. Mishra, "Image enhancement techniques on deep learning approaches for automated diagnosis of COVID-19 features using CXR images," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42649–42690, 2022, doi: 10.1007/s11042-022-13486-8.
- [23] S. Pang, T. H. G. Thio, F. L. Siaw, M. Chen, and Y. Xia, "Research on improved image segmentation algorithm based on GrabCut," *Electronics*, vol. 13, no. 20, 2024, doi: 10.3390/electronics13204068.
- [24] L. G. Manzano and J. G. Alfaro, "Software vulnerability detection under poisoning attacks using CNN-based image processing," *International Journal of Information Security*, vol. 24, no. 2, 2025, doi: 10.1007/s10207-025-00989-2.
- [25] S. Vidhanaarachchi, J. L. Wijekoon, W. A. S. P. Abey Siriwardhana, and M. Wijesundara, "Early diagnosis and severity assessment of weligama coconut leaf wilt disease and coconut caterpillar infestation using deep learning-based image processing techniques," *IEEE Access*, vol. 13, pp. 24463–24477, 2025, doi: 10.1109/ACCESS.2025.3537664.

## BIOGRAPHIES OF AUTHORS







**Kavitha Magadi Gopalakrishna**     is a research scholar with a background in electronics engineering, computer science, artificial intelligence (AI), and machine learning (ML). She is currently working as an assistant professor, with 10+ years of experience in the computer science field. Her current research focus lies in applying AI and ML techniques in the field of agricultural significance: disease prediction. Her current research delves into harnessing the power of AI and ML algorithms for the critical task of identifying and predicting diseases in trees. This innovative approach has the potential to revolutionize the coconut industry by enabling earlier detection and intervention, ultimately improving crop health and yield. Her specific areas of interest within AI and ML, IoT likely involve algorithms particularly well-suited to deep learning, CNN, image recognition, or pattern recognition. These techniques could be applied to analyze visual data of coconut trees, such as photographs, to detect signs of disease and predict future outbreaks. She can be contacted at email: kavitha.mg@kittiptur.ac.in.



**Raviprakash Madenur Lingaraju**     has completed his Ph.D. (Computer Science and Engineering) at Visvesvaraya Technological University. He has 13+ years of teaching experience in various fields such as machine learning and deep learning, data science, cloud computing, wireless ad hoc networks, and image processing. He has published various research papers in reputed journals and conferences such as IEEE and Springer. He has delivered various talks on machine learning, data science and its applications, and cyber security. He has reviewed many national and international journals and conference papers. He has received certificate of appreciation as a session chair in international conference, received certificate of appreciation from IEEE Bangalore section for outstanding dedication and support for IEEE International Conference. Research scholars have awarded Ph.D., under his guidance and he is guiding over 6+ research scholars in various fields of machine learning, network security, and deep learning. He has patent in an effective method for fake medicine detection and recommendation system and anomaly detection in cloud environment using deep learning application. He can be contacted at email: raviprakashml@kittiptur.ac.in.



**Ananda Babu Jayachandra**     worked as a research scholar in the research area wireless sensor networks. He has 13+ years of teaching experience in various fields. His research has been widely recognized, with publications in leading journals and conferences. His area of interest is data science, artificial intelligence, cloud computing, and wireless ad hoc networks. His strength is machine learning and deep learning, hands-on R-hadoop, hands-on CISCO packet tracer network simulator. He has received faculty excellence award by international conference of engineering trends in science, engineering and management for best contribution in academic research and development activity. He has contributed to the academic community by guiding 6+ Ph.D. students. He can be contacted at email: abj@mcehassan.ac.in.