

A feature-enhanced deep learning model for multi-class paddy ailment prediction in smart farming

Santhosh Byramangala Jayaramaiah¹, Balarengadurai Chinnaiah¹, Boddu Ravi Prasad²

¹Vidyavardhaka College of Engineering, Visvesvaraya Technological University, Belagavi, India

²Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology and Management, Hyderabad, India

Article Info

Article history:

Received Aug 19, 2025

Revised Jul 16, 2026

Accepted Aug 12, 2026

Keywords:

Attention mechanism

CNN-CBAM

Deep learning

Feature extraction

Paddy ailment detection

Smart agriculture

ABSTRACT

Paddy is an essential staple food crop eaten by more than 50% of the world population, particularly in Asia, where countries like China and India account for over 50% of both production and consumption. However, paddy cultivation is frequently threatened by numerous ailments that badly impact crop production and quality. Normal ailment identification approaches are often manual, time-consuming, and error-prone. To address these challenges, deep learning (DL)-based approaches have been widely adopted for automated paddy ailment detection. While these models have achieved commendable accuracy, many fail to focus on effective feature extraction, which is vital for accurate classify across diverse ailment categories. Hence, this study proposes an advanced convolutional neural network (CNN) based approach associated with convolutional block attention module (CBAM) for enhancing feature representation by focusing on both spatial and channel-wise importance. The approach was trained and executed on the widely used “paddy doctor” dataset, covering 2-class, 10-class, and 13-class classify scenarios. The proposed CNN-CBAM achieved outcomes with 99.75%, 99.93%, and 99.92% accuracy, respectively, outperforming existing models. These outcomes show the model’s effectiveness in capturing discriminative features and improving classify performance. This model contributes a foundation for future extensions in precision agriculture, including pest detection and real-time monitoring.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Santhosh Byramangala Jayaramaiah

Vidyavardhaka College of Engineering, Visvesvaraya Technological University

Belagavi, India

Email: santhoshb_12@rediffmail.com

1. INTRODUCTION

Paddy (*Oryza sativa* L.), majorly known as rice, is one among many essential staple crops worldwide, feeding more than 50% of the world’s people. It plays a vital role not only in ensuring food security but also in the socio-economic stability of several countries, especially in Asia and Africa [1], [2]. Paddy cultivation supports millions of smallholder farmers and is a significant contributor to rural employment and gross domestic product (GDP) in developing nations. As per the Food and Agriculture Organization (FAO), over 90% of the world’s rice is produced and spent in Asia, China, and India alone accounting for closely 50% of world’s production and intake [3], [4]. Southeast Asian countries like Indonesia, Vietnam, Thailand, and Philippines also contribute significantly. Outside Asia, rice is cultivated in parts of Africa, Latin America, and United States, though their shares are relatively smaller—Africa contributes about 6% to global production and Latin America about 4% [5]. Despite its critical importance, paddy cultivation is exposed to a wide range of that significantly reduce yield and quality. Common paddy

ailments include “bacterial-plant leaf-blight”, “bacterial-planet leaf-streak”, “blast”, “brown-spot”, “tungro”, “hispa”, “leaf-roller”, and several “stem-borers” [6], [7].

These ailments manifest through various symptoms such as yellowing of leaves, lesions, or stunted growth, and if not diagnosed early, can lead to large-scale crop failure. Traditionally, ailment identification has relied on manual visual inspection by experts, which is time-consuming, labor-intensive, and prone to human error due to the subtle visual differences between ailments [8]. In recent years, artificial intelligence (AI) and deep learning (DL) approaches have been increasingly applied in agriculture for automated ailment detection and classify [9]. These techniques, particularly convolutional neural networks (CNNs) [10], [11], have shown promise in accurately identifying different paddy ailments from leaf images. They offer scalability, speed, and high precision, allowing real-time diagnosis in the field. However, a significant limitation of many of these CNN-based models is their insufficient focus on feature refinement and extraction. While they achieve high classify accuracy on benchmark datasets, they often overlook the importance of learning where and what features in the image contribute most to the classify decision. This oversight can lead to reduced generalizability and performance in real-world, complex scenarios. Although various DL models have been developed for paddy ailment classify and have reported high accuracy [12]–[20], many of these models have not explicitly focused on enhancing the quality of feature extraction. As a result, the model may fail to learn fine-grained distinctions between visually similar ailments or may be distracted by background noise. This leads to decreased accuracy. Hence, there is a need for an approach that not only classifies ailments but also extracts and emphasizes the most relevant spatial and channel-wise features from the images to improve interpretability and robustness.

To address this issue, this work proposes a CNN model integrated with convolutional block attention module (CBAM). The CBAM model enhances traditional CNNs by sequentially applying channel and spatial attention mechanisms to the feature maps. This allows the model to learn “what” features to focus on (channel-attention) and “where” in the image those features are located (spatial-attention). By doing so, the model becomes more attentive to subtle ailment patterns and less sensitive to irrelevant background information, resulting in significantly improved classify accuracy and robustness. The contributions of the work are as follows novel CNN-CBAM model is proposed for efficient and accurate classify of paddy leaf ailments. The model is evaluated on the “paddy doctor” dataset using 2-class, 10-class, and 13-class configurations. The model shows strong generalization across varying class complexities, outperforming existing CNN-based methods.

The rest of the manuscript is organized as follows. Section 2 reviews related work and existing DL approaches for paddy ailment classify. Section 3 details the proposed CNN-CBAM architecture. Section 4 presents the experimental setup, outcomes, and a comparison with existing approaches. Finally, section 5 concludes the paper and outlines future research directions.

2. LITERATURE SURVEY

In current years, improvements in DL and computer vision have significantly transformed the area of precision agriculture, especially to detect and classify of crop ailments. Among these, the classify of paddy leaf ailments has emerged as a vital research area due to the crop’s economic importance and vulnerability to a wide range of ailments. Petchiammal *et al.* [12] presented a dataset called “paddy doctor” for the classify and prediction of paddy ailments. The dataset was created by taking red, green, blue (RGB) photos of rice paddies from the Tirunelveli region in Tamil Nadu. The dataset consisted of 16,225 paddy leaf images, which had 13 classes in which 12 classes (“bacterial-leaf-blight”, “bacterial-leaf-streak”, “bacterial-panicle-blight”, “black-stem-borer”, “blast”, “brown-spot”, “downy-mildew”, “hispa”, “leaf-roller”, “tungro”, “white-stem-borer”, “yellow-stem-borer”) were different paddy ailment classes, and 1 class was normal. All the images were in resolution of 256×256 pixels in .jpg format. For classify, five ML approaches were employed, which included residual network 34 layers (ResNet34), Xception, visual geometry group 16 (VGG16), mobile network (MobileNet), and deep-CNN, where ResNet34 achieved better outcomes for classify, achieving 97.50% accuracy. Their work did not focus on feature extraction. Ni *et al.* [13] conducted paddy ailment classify using the “Paddy Doctor” dataset, in which they only considered five classes, i.e., 5 ailment classes and 1 normal class (“blast”, “brown-spot”, “dead-heart”, “hispa”, “tungro”, and “normal”). First, in their work, utilized image enhancing approaches on the dataset for augmenting the dataset by modifying the saturation and contrast levels, flipping and adding blur for balancing samples of paddy leaves. Further, for classify, utilized re-parameterized visual geometry group (RepVGG) and enhanced it by adding an efficient channel attention (ECA) in RepVGG, providing a novel approach called RepVGG-ECA. Findings showed that RepVGG-ECA achieved 97.06% accuracy. This work considered feature extraction using feature maps of ECA.

Attallah [14] presented a model called as rice-paddy network (RiPa-Net), which was built using three CNN approaches, i.e., dark network 19 (DarkNet19), ResNet19, and MobileNet. For feature extraction, this study applied the dual-tree complex wavelet transform (DTCWT) for extracting features from the first

layer of DarkNet19, ResNet19, and mobileNet for extracting spectral-temporal information. Also, it utilized discrete cosine transform (DCT) and principal component analysis (PCA) for extracting features from the three CNN approaches, thereby reducing the first-layer feature dimensionality. Further, the second-layer spatial deep-features were fused with time-frequency deep-features. Further, a feature selection process was introduced for reducing the size of the feature vector and selecting important features for classify. For classify utilized cubic-support vector machine (CSVM) was utilized. For evaluation considered “paddy doctor” dataset was considered, which considered on 10 classes having 10,407 total samples (9 ailments: “blast”, “hispa”, “dead-heart”, “tungro”, “brown-spot”, “downy-mildew”, “bacterial-leaf-blight”, “bacterial-panicle-streak”, “bacterial-panicle-blight”, and 1 normal) and used augmentation (by scaling, shearing, flipping, and rotation) for avoiding overfitting and boosting CNN performance. Findings showed that using DarkNet19 + ResNet19 + MobileNet + PCA + CSVM achieved 97.47% accuracy, and using DarkNet19 + ResNet19 + MobileNet + DCT achieved 97.50% accuracy. El-Shahat *et al.* [15] presented a hybrid model that combined a deep neural network and mobile network (HMobileNetDNN) for paddy-ailment classify. The main aim was to reduce cost using MobileNet and extract complex features from the images using deep neural network (DNN). They also used four other models, i.e., DenseNet121, MobileNetV2, InceptionV3, and ResNetV2 for comparing their proposed HMobileNetDNN. For evaluation considered “paddy doctor” dataset was considered, which had 10 classes having 10,407 total samples (9 ailments: “blast”, “hispa”, “dead-heart”, “tungro”, “brown-spot”, “downy-mildew”, “bacterial-leaf-blight”, “bacterial-panicle-streak”, “bacterial-panicle-blight”, and 1 normal). Findings showed that HMobileNetDNN performed better in comparison with four other models, achieving 91.8% accuracy.

Dutta *et al.* [16] used the dataset presented in [17], [18], which had a total of 5,932 samples comprising 4 classes (“bacterial-blight”, “blast”, “brown-spot”, and “tungro”) of paddy ailment and 1 normal class. First, they pre-processed the image using bi-lateral filter to improve image quality. Further, used cascading-autoencoder attention-residual u-network (CAAR-U-Net), CNN, and MobileNetV2 for classify. Findings showed that among different models, CNN achieved better accuracy, i.e., 98. They failed to consider feature extraction. Mukherjee *et al.* [19] conducted a study on different AI-based approaches for rice/paddy leaf ailment prediction. This study discussed the important image acquisition approaches, imaging modalities and visual features required for classify and prediction. Also, discussed the different image processing approaches for feature extraction. Finally, compared different ML approaches’ outcomes using different datasets. Leelavathy and Kovvur [20] presented a model called as deep-CNN in which deep-CNN was used with hyperparameter tuning. The deep-CNN considered feature extraction before classify, providing better outcomes. For the evaluation considered “paddy doctor” dataset was considered, where findings showed that deep-CNN achieved 94.4% accuracy. The complete summary of literature survey is given in Table 1.

Table 1. Literature survey summary

Ref	Model used	Feature extraction	Accuracy (%)
[12]	ResNet34, Xception, VGG16, MobileNet, deep-CNN	No	97.50
[13]	RepVGG-ECA	Yes	97.06
[14]	RiPa-Net (DarkNet19 + ResNet19 + MobileNet + PCA)	Yes	97.50
[15]	HMobileNetDNN	No	91.80
[16]	CAAR-U-Net, CNN, MobileNetV2	No	98.00
[20]	Deep-CNN	Yes	94.40

From the literature survey [21], [22], it is seen that many researchers have explored various DL architectures and hybrid models using the “paddy doctor” dataset [23]. While several models have achieved high classify accuracy, a major limitation in many of these works is the lack of attention to effective feature extraction mechanisms [24], [25], which can greatly impact the model’s generalization and interpretability [26]. Hence, the proposed CNN-CBAM model addresses this gap by enhancing feature learning in an end-to-end and efficient manner, resulting in better accuracy and robustness in both binary and multi-class classify settings. The complete CNN-CBAM model is discussed in detail in the next section. Critical comparisons: in reviewing the various approaches used in recent studies, it is essential to include a critical comparison of different methods. It appears that CNN-based methods excel in image recognition tasks due to their DL capabilities. Hybrid methods merge traditional techniques with machine learning, leading to improved accuracy, particularly in structured datasets. Attention-based methods demonstrate strengths in focusing on relevant features but may come with increased computational demands. Meanwhile, handcrafted feature methods still maintain their relevance, especially in scenarios where data is limited. A table summarizing these comparisons would provide clarity on the strengths and weaknesses of each approach.

3. METHOD

This methodology first discusses the CNN-CBAM architecture, then the preprocessing steps considered, then the feature extraction process using CNN-CBAM and finally the prediction/classify approach used in CNN-CBAM. In Figure 1, the architecture of the proposed CNN-CBAM model is presented. The model is designed to receive RGB paddy leaves images with a fixed input shape of $128 \times 128 \times 3$, which represents height, width and color channels. In CNN-CBAM, the input image is first passed through a convolutional layer, where 32 filters are considered having a size 3×3 . These filters slide over the image for detecting low-level features, like simple textures, corners and edges. The output of this convolutional layer process provides feature maps, which are then processed using the rectified linear unit (ReLU) activation function for introducing non-linearity. Further, the pooling process reduces the spatial dimension by selecting important features from each local-region. This helps in reducing computation cost, providing better features, and preserving the most relevant patterns. The pooling layer outputs feature-maps which are then passed into CBAM, which consists of two sequential attention approaches, i.e., channel attention and spatial attention. The channel-attention helps in determining “what” features are important across all channels, and the spatial-attention helps in determining “where” in the image the critical information lies. Using these two attention forms, the model is to recalibrate feature maps, suppressing irrelevant information and enhancing meaningful features. Further, the CBAM-refined output is then further fed to another convolutional layer, which has 64 filters of 3×3 size again, which is followed by a pooling process and a second CBAM block. This stage provides the CNN-CBAM for capturing higher-level and more abstract features, like texture-specific features for different paddy ailments.

Following convolution and attention blocks, the feature maps are flattened into a one-dimensional vector in preparation for prediction/classify. This flattened vector is further passed to a fully connected layer (FCL) (dense), which has 128 neurons and is activated using ReLU. Further, in this work, a dropout layer was included for reducing overfitting by randomly setting 30% of input units to zero during training. The final layer of the CNN-CBAM model is a softmax classify layer, which maps the processed feature to a probability distribution over target classes (binary and multi). The entire model is trained using the Adam optimizer, which adjusts network weights for minimizing sparse categorical cross-entropy loss. This loss function is important for multi-class classify, where labels are encoded as integers rather than one-hot vectors. During training, the CNN-CBAM model learns to identify both high and low-level visual patterns related to different paddy ailments. Also, the CBAM helps dynamically in highlighting the most relevant channels and spatial locations at every network.

This section outlines the methodology used in this study, which comprises data preprocessing, model selection, training, and evaluation steps. Workflow diagram and pseudocode: a workflow diagram as shown in Figure 1, illustrates the complete process flow of the proposed method. This diagram outlines the key steps involved, from data collection and preprocessing to model evaluation. Algorithm 1 presents the pseudocode for the CNN-CBAM model used in this study.

Algorithm 1. CNN-CBAM algorithm

- 1: Load dataset.
- 2: Preprocess data (normalization, augmentation).
- 3: Split the dataset into training, validation, and test sets.
- 4: Initialize model with specified architecture.
- 5: Set hyperparameters (learning rate, batch size, epochs).
- 6: For each epoch:
 - i) Train model using training set.
 - ii) Validate model using validation set.
 - iii) Adjust hyperparameters if necessary.
- 7: Evaluate the model on the test set.
- 8: Generate performance metrics and a confusion matrix.
- 9: Visualize attention maps.

The selection of hyperparameters is critical to the performance of the model. In this study, a learning rate of 0.001 was chosen based on preliminary experiments indicating this rate allowed for stable convergence during training. A batch size of 32 was utilized to balance training speed and the stability of the gradient descent updates. Furthermore, the number of epochs was set to 50, as initial trials showed diminishing returns in performance improvements beyond this point. The computational complexity of the proposed method is primarily influenced by the model architecture and dataset size. The time complexity is $O(n^2)$, where n represents the number of data samples processed during each training iteration. This suggests that as the dataset grows, the processing time may increase quadratically. However, optimizations such as

mini-batch training and early stopping can mitigate some of this complexity, allowing for scalable implementation in larger datasets.

Preprocessing plays a critical role in preparing raw image data for effective training in DL models. The primary goal of preprocessing in this work is to ensure consistency in input dimensions and normalize data. In this work, all input images were resized to a fixed resolution of 128×128 pixels. This step provided a consistent input size for efficient batch processing and CNN-CBAM design. Also, resizing helped in standardizing the dataset, which can have images of different dimensions because of differences in capturing devices or by image cropping. By resizing, every image was fit into the expected input format for CNN, providing uniform-sized images across the dataset. As pixel intensity values in any raw images range from 0 to 255. In this work, these values were normalized by scaling them to the range of $[0,1]$. This was achieved by dividing every pixel value by 255. The normalization process helped in improving the training efficiency of the CNN-CBAM model by keeping input values bounded and small, which helped in preventing problems related to slow convergence. Also, it ensured all input features contributed equally to the CNN-CBAM model learning process, especially when using optimizers like Adam, which are sensitive to input scale. This preprocessing step ensured that the data is clean, consistent and in a format suitable for effective feature extraction and training of the CNN-CBAM model for classifying ailment accurately.

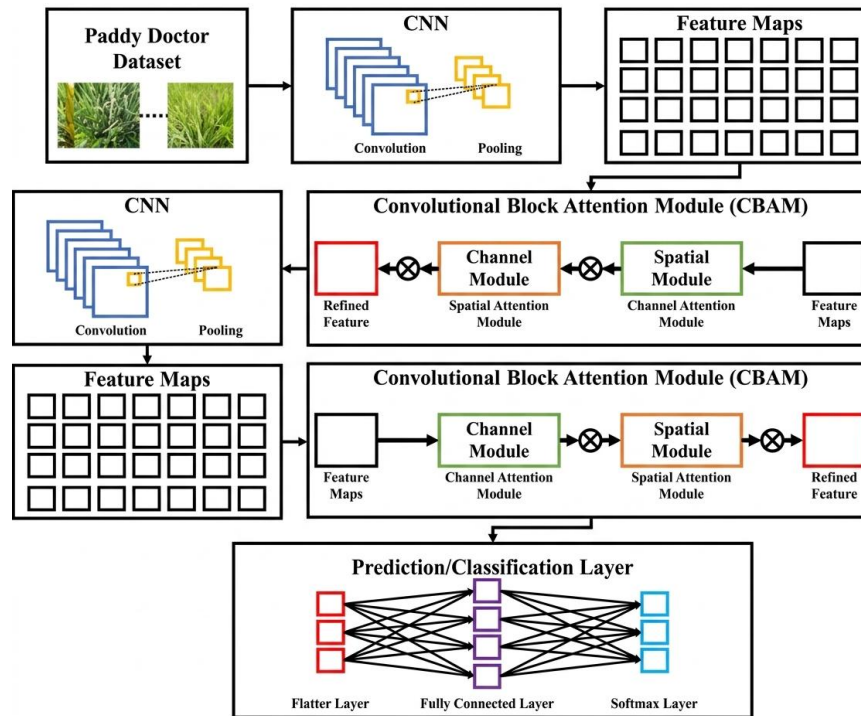


Figure 1. CNN-CBAM architecture for paddy ailment classify

In this work, the CNN architecture is enhanced using CBAM to improve CNN-CBAM's ability to focus on paddy ailment-relevant regions in paddy leaf images. The combination of CNN and CBAM enables the proposed CNN-CBAM for extracting both low-level (edge and textures) and high-level (ailment patterns features). In this work, the CNN extracts feature by applying a convolution operation which scan input images with trainable filters. Consider input image as $X \in \mathbb{R}^{H \times W \times C}$, where in this work, the image has been resized from $256 \times 256 \times 3$ to $128 \times 128 \times 3$. Hence, from this, the convolutional layer applies. K filters $\{W_k\}_{k=1}^K$, each creating a feature map F as presented in (1).

$$F_k = \phi(X \otimes W_k + b_k) \quad (1)$$

Where $\otimes$ denotes convolution, b_k denotes bias-term, W_k denotes the weights of the kernel, and $\phi(\cdot)$ denotes a non-linear activation function (ReLU). As the network deepens, subsequent layers detect increasingly abstract features, important for differentiating among different paddy ailments. After the extraction of feature maps from the convolutional layer, this work utilizes pooling-layer for reducing spatial-dimensions while retaining the most important features. The pooling operation is performed using (2).

$$F_{pool}(i, j) = \max_{(m, n) \in \Omega} F(i + m, j + n) \quad (2)$$

Where F_{pool} denotes feature maps attained by a pooling layer, i and j are pixels before the max function, m and n are pixels after the pooling function, and Ω denotes pooling window. This operation increases translational invariance and reduces computational complexity. After the pooling operation, the feature maps F_{pool} are passed on to CBAM, where the CBAM significantly improves feature extraction by refining intermediate feature-maps along two dimensions, i.e., spatial and channel-attention. The CBAM adaptively emphasizes informative features and suppresses less useful ones. Let $F_{pool} \in \mathbb{R}^{H \times W \times C}$ be an input feature-map for CBAM. In CBAM, first, the channel-attention focuses on “what” features are required for modelling inter-channel relationships. In this work, two pooling operations, global average pooling (GAP) and global max pooling (GMP), were applied across spatial dimensions for generating two descriptors. The operation of GAP and GMP is presented in (3) and (4). These descriptors are further passed through a shared multi-layer perceptron (MLP), which consists of two dense layers (with ReLU activation in between). The MLP process is defined as (5). In (5), σ denotes sigmoid activation function. The resulting vector M_c is then broadcast and multiplied with the original feature map using (6).

$$F_{avg} = GAP(F_{pool}) \quad (3)$$

$$F_{max} = GMP(F_{pool}) \quad (4)$$

$$M_c = \sigma \left(MLP(F_{avg}) + MLP(F_{max}) \right) \quad (5)$$

$$F' = F \cdot M_c \quad (6)$$

This operation enhances/suppresses channels based on their significance to the task. Further, after the channel-attention operation, the feature maps are passed through the spatial-attention, which determines “where” in the feature map the model should focus. Using the channel-refined feature map F' it computes spatial attention by applying average and max-pooling along the channel-axis using (7) and (8). The two resulting 2-dimensional feature maps are concatenated and passed through a convolutional layer with a kernel size 7×7 using (9). This produces a 2-dimensional spatial attention map M_s . The final output of the CBAM is obtained using element-wise multiplication, which is done using (10).

$$F_{avg}^{spatial}(i, j) = \frac{1}{C} \sum_{c=1}^C F'_{i,j,c} \quad (7)$$

$$F_{max}^{spatial}(i, j) = \max_{c=1}^C F'_{i,j,c} \quad (8)$$

$$M_s = \sigma \left(Conv_{7 \times 7}([F_{avg}^{spatial}; F_{max}^{spatial}]) \right) \quad (9)$$

$$F'' = F \cdot M_s \quad (10)$$

After this operation, the feature-map F'' is passed on to the convolutional layer, which has 64 filters of 3×3 size again (1), which is followed by pooling process (2) and a second CBAM block (3) to (10). Using the following operation, the CBAM-enhanced CNN dynamically adjusts both the importance of individual feature channels and spatial-regions of image, helping CNN-CBAM to better focus on ailment specific features, discolorations or fungal patterns. This is important in paddy leaf ailment detection, where such symptoms appear subtly or in small regions. Nevertheless, the CBAM also provides a better result by making more variants to background noise, lighting variations and irrelevant leaf structures. Once high-quality features are extracted using CNN-CBAM, the next step in this work was to predict/classify paddy ailment. For the prediction/classify, consider the feature map. F'' is converted to a one-dimensional vector using a flattening approach using (11).

$$v = Flatten(F'') \quad (11)$$

This step prepares feature-tensor for input to FCL. The v is then passed on through a dense layer to map it into latent-space. Let W_1 and b_1 be a weight matrix and a bias for the vector v . The transformation is evaluated using (12). In (12), the ϕ is the ReLU activation function. For preventing overfitting, dropout was applied to the output of FCL using (13). In (13), r denotes the rate of random zeroes, which was set as 30% of the neurons during training, forcing CNN-CBAM to learn more robust features. The final layer was a dense layer followed by softmax activation, which outputs class probabilities. Let C be the number of classes

in the dataset, such that $W_2 \in \mathbb{R}^{C \times h}$ and $b_2 \in \mathbb{R}^C$. From the following, the softmax function is defined using (14). In (14) provides a probability vector $\hat{y}$, where each element $\hat{y}_i$ corresponds to the likelihood of input belonging to the class i , i.e., (15). Moreover, the model is trained using a sparse categorical cross-entropy loss function. Hence, for the dataset for N samples, the loss is evaluated as (16).

$$z_1 = \phi(W_1 v + b_1) \quad (12)$$

$$z'_1 = \text{Dropout}(z_1, r = 0.3) \quad (13)$$

$$\hat{y} = \text{softmax}(W_2 z'_1 + b_2) \quad (14)$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}, i = 1, 2, \dots, C \quad (15)$$

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}_{i, y_i}) \quad (16)$$

In (16), $\hat{y}_{i, y_i}$ denotes predicted probability for true-class y_i of i^{th} sample. The Adam optimizer in this work was used for updating weights using gradient descent with adaptive learning rates. This ensured stable and efficient convergence. In CNN-CBAM, the model leverages CNNs in hierarchical feature learning and augments them with CBAM's dual-attention mechanism to isolate discriminative features critical for classifying paddy ailments. The final softmax layer then converts these refined features into actionable predictions, achieving accurate classify even under subtle symptom variations. The outcomes of the CNN-CBAM model are discussed in detail in the next section.

4. RESULTS AND DISCUSSION

The outcomes and discussion section presents the experimental setup and findings of the CNN-CBAM model. The implementation was carried out on a Windows 11 system with 16 GB RAM, NVIDIA GTX 1650 GPU, and Python programming in the Anaconda environment. The “paddy doctor” dataset was utilized, comprising 2-class, 10-class, and 13-class categories. This section discusses the hyper-parameter tuning parameters, CNN-CBAM performance in terms of performance metrics and finally, a comparative study is discussed.

For the evaluation of this work, the “paddy doctor” dataset was utilized [27], [28]. The dataset was created by taking RGB photos of rice paddies from the Tirunelveli region in Tamil Nadu. The dataset consisted of 16,225 paddy leaf images, which had 13 classes, in which 12 classes (“bacterial-leaf-blight”, “bacterial-leaf-streak”, “bacterial-panicle-blight”, “black-stem-borer”, “blast”, “brown-spot”, “downy-mildew”, “hispa”, “leaf-roller”, “tungro”, “white-stem-borer”, and “yellow-stem-borer”) were different paddy ailment classes, and 1 class was normal. All the images were in a resolution of 256×256 pixels in .jpg format. The complete details of the total samples of the 13 classes are given in Table 2. Further, from this dataset, a binary dataset was created, where all the 12 classes were considered as ailment, and the normal class was considered as healthy paddy. The complete details of the total samples of the 2 classes are given in Table 3.

Table 2. Total samples in the 13-class paddy doctor dataset

Classes	Samples count
Bacterial-panicle-blight	450
Bacterial-leaf-streak	505
Black-stem-borer	506
Bacterial-leaf-blight	648
Yellow-stem-borer	765
Downy-mildew	868
Leaf-roller	1,095
Brown-spot	1,257
White-stem-borer	1,273
Tungro	1,951
Hispa	2,151
Blast	2,351
Normal	2,405
Total	16,225

Similar to the work in [3], [4], [9], this work also considered the 10-class “paddy doctor” dataset, which consisted of 10 classes having 10,407 total samples (9 ailments: “blast”, “hispa”, “dead-heart”, “tungro”,

“brown-spot”, “downy-mildew”, “bacterial-leaf-blight”, “bacterial-panicle-streak”, “bacterial-panicle-blight”, and 1 normal). The total count of the samples considered for each class is given in Table 4. during CNN-CBAM training, hyperparameters were selected for the image-classify task and fine-tuned for optimal performance. The complete details of the hyperparameters used are discussed in Table 5.

Table 3. Total samples in the 2 class paddy doctor dataset

Classes	Samples count
Ailment	13,820
Normal	2,405
Total	16,225

Table 4. Total samples in 10 class paddy doctor dataset

Classes	Samples count
Bacterial-panicle-bight	377
Bacterial-panicle-streak	380
Bacterial-leaf-blight	479
Downy-mildew	620
Brown-spot	965
Tungro	1,088
Dead heart	1,442
Hispa	1,594
Blast	1,738
Normal	1,764
Total	10,407

Table 5. Hyperparameters used for CNN-CBAM

Hyperparameter	Value	Description
Image size	128×128	Resized input images for uniformity
Batch size	32	Number of samples per gradient update
Epochs	30	Total number of training iterations
Optimizer	Adam	Adaptive learning rate optimization algorithm
Loss function	Sparse categorical cross-entropy	Used for multi-class classify
Dropout rate	0.3	Regularization to prevent overfitting in dense layers
CBAM ratio	8	Channel reduction ratio in the CBAM attention block
Convolution filters	32, 64	Number of filters in the Conv2D layers
Activation function	ReLU/softmax	ReLU for hidden layers, softmax for output probability distribution

For the performance evaluation of CNN-CBAM, the four metrics, accuracy, precision, recall, and F1-score were used for prediction/classify as presented in (17) to (20), respectively, where TN denotes true negative, TP denotes true positive, FN denotes false negative, and FP denotes false positive. The outcomes of CNN-CBAM show better classify performance across all three dataset configurations, 2-class, 10-class, and 13-class, as presented in Figures 2 to 4, respectively. The attention-based enhancement in CBAM has significantly improved the model’s ability to focus on the most relevant features within both channel and spatial dimensions, which is reflected in the near-perfect evaluation metrics achieved across each class configuration. The consistent and high values of accuracy, precision, recall, and F1-score signify that the model not only predicts the correct class in most cases but also does so reliably across all classes with minimal FPs and FNs [29], [30].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (17)$$

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

$$Recall = \frac{TP}{TP+FN} \quad (19)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (20)$$

On the binary classify task distinguishing between “ailment” and “normal” classes, the CNN-CBAM achieved 99.75% accuracy, with 99.76% precision, 99.75% recall, and 99.76% F1-score. This performance is because of the reduced complexity of the binary classify task, where the model simply determines whether the leaf is healthy or affected by any ailment. The CBAM module strengthens the network’s ability to emphasize ailment-specific patterns, such as lesions, color distortions, or deformities in

the leaf structure, while suppressing irrelevant background noise. The high precision indicates that almost all leaves identified as ailment were indeed infected, while the equally high recall confirms the model's success in capturing nearly every ailment instance with very few missed cases.

Figure 2 shows the performance study of CNN-CBAM on the 2-class paddy doctor dataset. In the more challenging 10-class setup, as shown in Figure 3, which includes 9 ailment types and 1 normal class, the model attained 99.93% across all metrics. This is because the inter-class similarities among different ailments, such as blight, streaks, and fungal patches, may visually resemble each other. The enhanced feature extraction capability of CNN-CBAM allows the model to discriminate subtle variations between different ailment manifestations, such as the shape and spread of lesions or the presence of insect bites. The spatial attention submodule, in particular, contributes to localizing “where” in the image the ailment appears, while the channel attention captures “what type” of features are most significant, leading to highly effective multi-class classify.

Despite being the most complex configuration, the model also achieved 99.92% accuracy, precision, recall, and F1-score on the full 13-class paddy doctor dataset, as shown in Figure 3. This result underscores the robustness and generalization capacity of the CNN-CBAM. Handling 13 fine-grained categories, many of which share common visual traits, requires the model to extract highly discriminative and hierarchical features. The CBAM block aids this process by sequentially refining the feature maps at both the global (channel) and local (spatial) levels, enabling the model to understand important differences even in visually similar ailments like bacterial-leaf-blight vs. bacterial-leaf-streak, or white-stem-borer vs. yellow-stem-borer. Furthermore, such performance in a dataset with imbalanced class distributions demonstrates that the attention mechanisms help mitigate the bias toward overrepresented classes, ensuring consistent performance across all categories.

In all three configurations from Figures 2 to 4, the CNN-CBAM outperforms traditional CNNs by a substantial margin. It achieves better classify outputs, not merely through depth or width of layers but by incorporating biologically inspired attention mechanisms that emulate how human vision might focus on the most relevant portions of an image. These outcomes highlight the technique's strong potential for agricultural applications, offering scalable, accurate, and interpretable solutions for paddy ailment identification at both broad and granular levels.

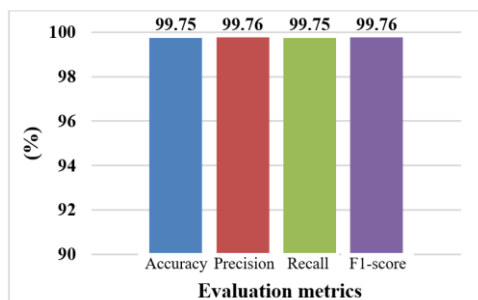


Figure 2. Performance study of CNN-CBAM on 2-class paddy doctor dataset

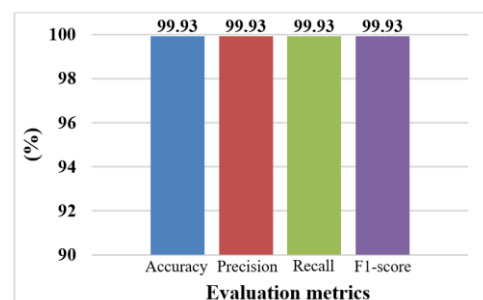


Figure 3. Performance study of CNN-CBAM on 10-class paddy doctor dataset

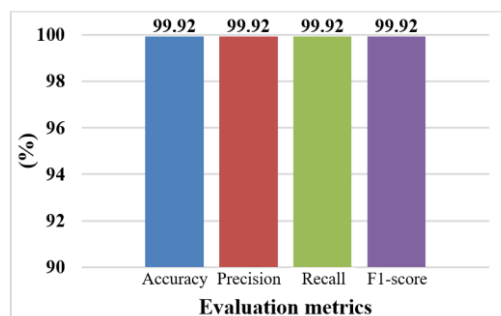


Figure 4. Performance study of CNN-CBAM on 13-class paddy doctor dataset

The proposed CNN-CBAM model demonstrates significant improvements in classify performance compared to existing approaches across all dataset configurations (2-class, 10-class, and 13-class), as presented in Table 6. In the 13-class classify task, CNN-CBAM achieved 99.92% accuracy, outperforming

ResNet [12] (97.5%) and complex hybrid models like MobileNet + ResNet + DarkNet + DCT + CSVM [14] (97.5%). Similarly, for the 10-class setup, CNN-CBAM attained 99.93% accuracy, substantially higher than HMobileNet DNN [15] (91.8%), deep-CNN [20] (94.4%), and EfficientNet B4 [26] (94%-96.91%). For the binary classify task (2-class), CNN-CBAM reached 99.75% accuracy, indicating its robustness even in simpler classify problems.

The high performance of CNN-CBAM is attributed to its capability to effectively fetch and emphasize particular features using CBAM. CBAM enhances feature representation by putting attention mechanisms in both spatial and channel dimensions, which helps in focusing on ailment-specific regions of the paddy leaf images. Unlike previous models that either ignored or partially implemented feature extraction, CNN-CBAM integrates attention directly within the network, leading to more precise learning and classify. This allows the model to distinguish subtle variations among ailment categories with higher precision, recall, and F1-score, establishing it as a better solution for paddy ailment diagnosis.

Table 6. Comparison of classification performance with existing methods

Reference	Model	Class	Accuracy	Precision	Recall	F1-score
[12]	DCNN	13	88.84	89.22	88.84	88.81
	MobileNet		92.42	96.63	92.42	92.39
	VGG16		93.19	93.49	93.19	93.2
	Xception		96.58	96.61	96.58	96.57
	ResNet		97.5	97.52	97.5	97.5
[14]	MobileNet + ResNet-18 + DarkNet + DCT	10	97.5	97.39	97.07	97.22
	MobileNet + ResNet-18 + DarkNet + PCA		97.4	96.8	96.25	96.66
	MobileNet + ResNet-18 + DarkNet + DCT + CSVM		96.99	96.76	96.57	96.66
	MobileNet + ResNet-18 + DarkNet + PCA + CSVM		97.47	97.5	97.24	97.36
[15]	MobileNetV2	10	90.4	91.3	89	90
	ResNetV2		84.1	83.3	82	82.1
	DenseNet121		79.8	83.7	77	79
	InceptionV3		62.8	65	54.8	56.5
	HMobileNetDNN		91.8	91.8	90.7	91.2
[20]	Deep-CNN	10	94.4	85	87	86
[26]	EfficientNet B4	10	94-96.91	97	93	95
Proposed	CNN-CBAM	2	99.75	99.76	99.75	99.76
		10	99.93	99.93	99.93	99.93
		13	99.92	99.92	99.92	99.92

5. CONCLUSION

This work presented a DL-based technique for the effective detection and bifurcation of paddy leaf using the CNN-CBAM model. Paddy, being one among most essential staple crops worldwide, particularly in countries like India, China, Indonesia, and other parts of Asia, faces significant threats from a variety of leaf ailments. Traditional approaches for identifying these diseases are labor-intensive, time-consuming, and prone to human error, which limits their scalability and accuracy. With the increasing application of AI in agriculture, especially DL, automated and precise ailment classify systems have become essential. However, existing models, though effective in classify, often lack robust feature extraction mechanisms, leading to reduced overall performance. Hence, the proposed CNN-CBAM model addresses this limitation by incorporating attention modules that enhance feature representation by focusing on the most relevant spatial and channel information. Experimental evaluation on the “Paddy Doctor” dataset across 2-class, 10-class, and 13-class setups demonstrated that CNN-CBAM outperforms existing approaches, achieving accuracies of 99.75%, 99.93%, and 99.92%, respectively. In future progress, we plan to expand the system to not only classify ailments but also detect pests in paddy fields, thereby offering a comprehensive smart agriculture solution.

ACKNOWLEDGMENTS

The authors extend their heartfelt thanks to their mentor for unwavering supervision, valuable insights, and support throughout the research work.

FUNDING INFORMATION

No funding is raised for this research.

AUTHOR CONTRIBUTIONS CHART

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
Santhosh Byramangala Jayaramaiah	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Balarengadurai Chinnaiiah	✓	✓			✓	✓		✓	✓	✓	✓	✓		
Boddu Ravi Prasad	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT

The author declares no conflict of interest.

DATA AVAILABILITY

The data supporting the findings of this work are available within the article, reference [20].

REFERENCES

[1] A. J. Magubika, F. K. Fukah, E. K. Nassary, and G. M. Tryphone, "Analysing rice (*Oryza sativa* L.) production trends—area harvested, quantity and yield stability in Tanzania," *Discover Agriculture*, vol. 3, no. 1, Apr. 2025, doi: 10.1007/s44279-025-00204-9.

[2] N. Hashim *et al.*, "Smart farming for sustainable rice production: an insight into application, challenge, and future prospect," *Rice Science*, vol. 31, no. 1, pp. 47–61, Jan. 2024, doi: 10.1016/j.rsci.2023.08.004.

[3] M. Joseph, S. Moonsammy, H. Davis, D. Warner, A. Adams, and T. D. T. Oyedotun, "Modelling climate variabilities and global rice production: a panel regression and time series analysis," *Heliyon*, vol. 9, no. 4, Apr. 2023, doi: 10.1016/j.heliyon.2023.e15480.

[4] W. Chen and X. Zhao, "Understanding global rice trade flows: network evolution and implications," *Foods*, vol. 12, no. 17, Sep. 2023, doi: 10.3390/foods12173298.

[5] N. A. Mohidem, N. Hashim, R. Shamsudin, and H. C. Man, "Rice for food security: revisiting its production, diversity, rice milling process and nutrient content," *Agriculture*, vol. 12, no. 6, May 2022, doi: 10.3390/agriculture12060741.

[6] K. Sujatha, T. K. Reddy, N. P. G. Bhavani, R. S. Ponmagal, V. Srividhya, and N. Janaki, "UGVs for agri spray with AI assisted paddy crop disease identification," *Procedia Computer Science*, vol. 230, pp. 70–81, 2023, doi: 10.1016/j.procs.2023.12.062.

[7] N. Paul, G. C. Sunil, D. Horvath, and X. Sun, "Deep learning for plant stress detection: a comprehensive review of technologies, challenges, and future directions," *Computers and Electronics in Agriculture*, vol. 229, Feb. 2025, doi: 10.1016/j.compag.2024.109734.

[8] B. Sambana *et al.*, "An efficient plant disease detection using transfer learning approach," *Scientific Reports*, vol. 15, no. 1, May 2025, doi: 10.1038/s41598-025-02271-w.

[9] K. Minhans, S. Sharma, I. Sheikh, S. S. Alhewairini, and R. Sayyed, "Artificial intelligence and plant disease management: an agro-innovative approach," *Journal of Phytopathology*, vol. 173, no. 3, May 2025, doi: 10.1111/jph.70084.

[10] A. Jafar, N. Bibi, R. A. Naqvi, A. S.-Niaraki, and D. Jeong, "Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations," *Frontiers in Plant Science*, vol. 15, Mar. 2024, doi: 10.3389/fpls.2024.1356260.

[11] S. U. Khan, A. Alsuhaibani, A. Alabduljabbar, F. Almarshad, Y. N. Altherwy, and T. Akram, "A review on automated plant disease detection: motivation, limitations, challenges, and recent advancements for future research," *Journal of King Saud University Computer and Information Sciences*, vol. 37, no. 3, May 2025, doi: 10.1007/s44443-025-00040-3.

[12] Petchiammal, B. Kiruba, Murugan, and P. Arjunan, "Paddy Doctor: a visual image dataset for automated paddy disease classification and benchmarking," in *6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, Jan. 2023, pp. 203–207, doi: 10.1145/3570991.3570994.

[13] H. Ni *et al.*, "Classification of typical pests and diseases of rice based on the ECA attention mechanism," *Agriculture*, vol. 13, no. 5, May 2023, doi: 10.3390/agriculture13051066.

[14] O. Attallah, "RiPa-Net: recognition of rice paddy diseases with duo-layers of CNNs fostered by feature transformation and selection," *Biomimetics*, vol. 8, no. 5, Sep. 2023, doi: 10.3390/biomimetics8050417.

[15] D. El-Shahat, A. Tolba, and A. K. Kumar, "An improved deep learning model for detecting rice diseases," *Optimization in Agriculture*, vol. 1, pp. 1–10, Jan. 2024, doi: 10.61356/j.oia.2024.1194.

[16] M. Dutta *et al.*, "Rice leaf disease classification—a comparative approach using convolutional neural network (CNN), cascading autoencoder with attention residual U-Net (CAAR-U-Net), and MobileNet-V2 architectures," *Technologies*, vol. 12, no. 11, Oct. 2024, doi: 10.3390/technologies12110214.

[17] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, Aug. 2020, doi: 10.1016/j.compag.2020.105527.

[18] P. K. Sethy, "Rice leaf ailment image samples," *Mendeley Data*, 2020, doi: 10.17632/fwcj7stb8r.1.

[19] R. Mukherjee, A. Ghosh, C. Chakraborty, J. N. De, and D. P. Mishra, "Rice leaf disease identification and classification using machine learning techniques: a comprehensive review," *Engineering Applications of Artificial Intelligence*, vol. 139, Jan. 2025, doi: 10.1016/j.engappai.2024.109639.

[20] B. Leelavathy and R. R. Kovvur, "Deep convolutional neural network (deep-CNN) for multi-class classification of biotic stress in paddy crop," *Journal of Computer Allied Intelligence*, vol. 3, no. 1, pp. 18–30, Feb. 2025, doi: 10.69996/jcai.2025002.

[21] R. S. Jesie, M. S. G. Premi, and T. Jarin, "Comparative analysis of paddy leaf diseases sensing with a hybrid convolutional neural network model," *Measurement: Sensors*, vol. 31, Feb. 2024, doi: 10.1016/j.measen.2023.100966.

- [22] M. Tasfe, A. Nivrito, F. Al Machot, M. Ullah, and H. Ullah, "Deep learning based models for paddy disease identification and classification: a systematic survey," *IEEE Access*, vol. 12, pp. 100862–100891, 2024, doi: 10.1109/ACCESS.2024.3419708.
- [23] J. B. C. T. and V. A., "Agricultural advancement: leveraging EfficientNetB4 for enhanced paddy disease classification," in *2024 IEEE Students Conference on Engineering and Systems* Jun. 2024, pp. 1–6, doi: 10.1109/SCES61914.2024.10652384.
- [24] M. S. H. Shovon, S. J. Mozumder, O. K. Pal, M. F. Mridha, N. Asai, and J. Shin, "PlantDet: a robust multi-model ensemble method based on deep learning for plant disease detection," *IEEE Access*, vol. 11, pp. 34846–34859, 2023, doi: 10.1109/ACCESS.2023.3264835.
- [25] N. Bharanidharan, S. R. S. Chakravarthy, H. Rajaguru, V. V. Kumar, T. R. Mahesh, and S. Guluwadi, "Multiclass paddy disease detection using filter-based feature transformation technique," *IEEE Access*, vol. 11, pp. 109477–109487, 2023, doi: 10.1109/ACCESS.2023.3322587.
- [26] J. Padhi, L. Korada, A. Dash, P. K. Sathy, S. K. Behera, and A. Nanthamomphong, "Paddy leaf disease classification using EfficientNet B4 with compound scaling and swish activation: a deep learning approach," *IEEE Access*, vol. 12, pp. 126426–126437, 2024, doi: 10.1109/ACCESS.2024.3451557.
- [27] V. Tanwar and S. Lamba, "An improved deep learning model for classification of the multiple paddy disease," in *2023 Third International Conference on Secure Cyber Computing and Communication*, May 2023, pp. 43–48, doi: 10.1109/ICSCCC58608.2023.10176446.
- [28] R. Suguna, S. Deepa, T. Kanimozhi, M. Chandru, G. Arun, and H. M., "Paddy plant disease detection using hybrid deep learning algorithms," in *2024 8th International Conference on Electronics, Communication and Aerospace Technology*, Nov. 2024, pp. 1382–1386, doi: 10.1109/ICECA63461.2024.10801067.
- [29] M. Tasfe, A. Nivrito, F. Al Machot, M. Ullah, F. A. Cheikh, and H. Ullah, "Deep learning-based models for paddy disease classification and segmentation: an experimental review," *IEEE Access*, vol. 13, pp. 152962–152996, 2025, doi: 10.1109/ACCESS.2025.3599355.
- [30] A. V-Duc, K. N-Vi, B. B-Quoc, and N. Kamel, "A comprehensive survey of super-resolution remote sensing image datasets: evolution, challenges, and future directions," *IEEE Access*, vol. 13, pp. 145350–145372, 2025, doi: 10.1109/ACCESS.2025.3599535.

BIOGRAPHIES OF THE AUTHORS



Santhosh Byramangala Jayaramaiah    is currently serving as an assistant professor in the Department of Computer Science and Engineering at BGS Institute of Technology, Faculty of Engineering, Management and Technology, Adichunchanagiri University, B.G. Nagara. He holds a B.E. in Computer Science and Engineering (Visvesvaraya Technological University or VTU, 2009) and an M. Tech. in Computer Networks and Engineering (VTU, 2012). He presently pursuing a Ph.D. in Computer Science and Engineering from Vidyavardhaka College of Engineering, Mysore, VTU, Belagavi, India with 13 years of teaching experience, he has made significant contributions to academia and research. His scholarly work includes: five conference papers published in IEEE proceedings. Three research papers in reputed journals. He is authorship of the textbook "Information and cyber security" (SIPH Publications). Contribution of two book chapters published by IGI Global. He is an active member of several professional bodies, including InSC, IEEE, ISTE, and IAENG. In recognition of his academic contributions, he received academic excellence award (2024) from institute of scholars (InSC). He can be contacted at email: santhoshbj221@gmail.com.



Dr. Balarengadurai Chinnaiyah    is a professor, Vidyavardhaka College of Engineering, India, with 20+ years of academic and research experience. He holds a Ph.D. in IT/CSE and has published 25+ research papers, six book chapters, and five patents. His expertise spans machine learning, blockchain, information security, OBE, and accreditation (NBA/NAAC), and he actively contributes as a reviewer for leading journals and organizer of academic programs. He can be contacted at email: balarengadurai@vvce.ac.in.



Dr. Boddu Ravi Prasad    has over 25 years of academic and administrative experience in engineering education. He is currently serving as dean (academics) and professor in Computer Science and Engineering (ratified by Jawaharlal Nehru Technological University – Hyderabad or JNTUH) at Marri Laxman Reddy Institute of Technology and Management, Hyderabad. He has previously worked as HOD and associate professor at reputed institutions, including Vasavi College of Engineering and RRS College of Engineering and Technology. He obtained his Ph.D. in Computer Science from GITAM University (2015), M.Tech. (Computer Science and Engineering) with distinction from JNTUH (2004), and B.Tech. (Computer Science and Engineering) from Nagarjuna University (1998). He has 24 papers in reputed international journals, 11 in international conferences, and 3 in national conferences. He published three patents. His research areas are deep learning, machine learning, artificial intelligence, and cloud computing. He can be contacted at email: rprasad.boddu@gmail.com.