Optimizing citrus disease detection: a transferable convolutional neural network model enhanced with the fruitfly optimization algorithm

Anoop Ganadalu Lingaraju, Asha Mangala Shankaregowda, Babu Kumar Sathiyamurthy, Santhrupth Budanoor Channegowda, Shruti Jalapur, Chaitra Palahalli Chennakeshava
Department of Computer Science and Engineering, School of Engineering and Technology, CHRIST University, Bengaluru, India

Article Info

Article history:

Received Sep 30, 2024
Revised Apr 9, 2025
Accepted Jun 8, 2025

Keywords:

Bacteria
Citrus fruit disease detection
Enhanced fruitfly optimization algorithm
Image analysis
Transferrable convolutional neural network

ABSTRACT

Fungal, bacterial, and viral diseases significantly threaten citrus production and quality worldwide, prompting producers to explore technological solutions to mitigate the financial impact of these diseases. Image analysis techniques have emerged as powerful tools for detecting citrus diseases by differentiating between healthy and diseased specimens through the extraction of discriminative features from input images. This paper introduces a valuable dataset comprising 953 color images of orange leaves from the species *Citrus sinensis* (L.) Osbeck, which serves to train, evaluate, and compare various algorithms aimed at identifying abnormalities in citrus fruits. The development of automated detection systems is crucial for reducing economic losses in citrus production, with this research focusing on twelve specific diseases and nutrient deficiencies. We propose a novel approach to citrus plant disease detection utilizing a hyper-parameter tuned transferrable convolutional neural network (TCNN) model, referred to as the enhanced fruitfly optimization algorithm (EFOA)-TCNN model. This model optimizes the parameters of TCNN using the EFOA and enhances architectural design by incorporating three convolutional layers alongside an energy layer instead of a traditional pooling layer. Experimental results demonstrate that the proposed EFOA-TCNN model outperforms existing state-of-the-art methods, achieving a sensitivity of 0.975 and an accuracy of 0.995.

Corresponding Author:

Asha Mangala Shankaregowda
Department of Computer Science and Engineering, School of Engineering and Technology, CHRIST University
Bengaluru, Karnataka, India
Email: ashagowda05@gmail.com

1. INTRODUCTION

Species that are indigenous to Australia, Melanesia, and certain regions of Asia are included in the genus *Citrus*, which is comprised of the fruit crops that are the most economically valuable on a global scale. Citrus fruits, which include sweet oranges and mandarins, are grown in more than 140 countries and are primarily cultivated for the purpose of serving the markets for fresh fruit and beverages. According to the Food and Agriculture Organization of the United Nations (FAO) in 2020 [1], sweet oranges account for 65 percent of the total citrus production worldwide [2], with Mediterranean countries being the leading exporters of fresh fruit. However, the persistent threat posed by a wide variety of fungal, bacterial, and oomycete diseases is a significant barrier to the efficient trade of citrus fruits, both domestically and internationally [3]. This is the case because citrus fruits are susceptible to a wide range of diseases. The presence of these pathogens, which include *Plenodomus tracheiphilus* and *Phytophthora* species, has the potential to result in yield losses of thirty to fifty percent during crucial phases of the plant's life cycle [2]. The susceptibility of citrus plants to diseases is exacerbated by their acid pH and high water content, which leaves them vulnerable during the pre-harvest and post-harvest stages-6 and 7 respectively [4].

The purpose of this study is to investigate the issue of citrus disease detection, which is a significant obstacle that affects citrus production all over the world. Because of the demands placed on computational resources and the constraints imposed by network design, the machine learning and deep learning techniques that are currently in use have limitations when it comes to detecting these diseases with the required level of efficiency. The reason for this is that there is a growing demand for agricultural products that are of high quality and safe for the environment [5]–[8]. Therefore, advanced detection systems are essential in order to minimize economic losses in the citrus industry. The approach that has been proposed provides fresh perspectives by presenting a transferrable convolutional neural network (TCNN) model that has been optimized with the enhanced fruitfly optimization algorithm (EFOA). Using this method, disease detection is improved while maintaining low computational costs, which makes it accessible and practical for applications in the real world [9].

As stated in the introduction, the paper's main goal is to identify citrus diseases through image analysis and machine learning. However, it should be noted that the research presents the EFOA-TCNN model to identify 12 different kinds of citrus plant illnesses and deficits [10]–[12]. Although the financial burden of citrus illnesses is mentioned in the abstract, the introduction will be extended to discuss the importance of automated detection. The financial ramifications will be highlighted, including avoiding large crop losses, maintaining quality control in the citrus sector, and the shortcomings of conventional manual inspections in detecting these illnesses. The primary goal of this research is to develop the EFOA-TCNN model to identify twelve distinct types of citrus diseases and nutritional deficiencies through image analysis and machine learning. Although the abstract references the financial burden of citrus diseases [13]–[15], the introduction will be expanded to emphasize the importance of automated detection. Key financial implications include the prevention of substantial crop losses, the maintenance of quality control in the citrus sector, and the limitations of traditional manual inspection methods in effectively identifying these diseases [16].

Moreover, this introduction will delve deeper into the EFOA and its role in fine-tuning the TCNN. It will also elucidate the innovative incorporation of an energy layer in place of the conventional pooling layer and discuss how hyperparameter optimization enhances overall model performance. By evaluating the EFOA-TCNN model using a publicly available dataset of orange leaves, this paper aims to present a comprehensive solution for accurate citrus disease classification.

Specifically, the paper is organized as follows. The literature that is pertinent to the topic is discussed in section 2. The methodology is presented in section 3. The results are discussed in section 4. Lastly, the findings are summarized in section 5.

2. RELATED WORKS

In their investigation of an automated system for citrus disease classification, Butt *et al*. [17] employed deep learning combined with optimal feature selection techniques. The initial phase of their approach involved data augmentation, which entails generating new images for the training dataset from existing examples. Leveraging transfer learning, the authors retrained two pre-existing models—DenseNet-201 and AlexNet-using the enhanced dataset derived from leaf images. Their experiments achieved a remarkable precision level of 99.6%. At each stage, the proposed framework was compared to state-of-the-art methodologies, demonstrating superior performance.

Yadav *et al*. [18] developed a computer vision system capable of automatically categorizing fruits and leaves, thereby facilitating efficient disease management in orchards. This study utilized features generated by CNNs and machine learning classifiers to effectively detect citrus black spot (CBS)-infected fruits and leaves exhibiting canker symptoms. The custom shallow CNN combined with radial basis function (RBF) support vector machine (SVM) achieved an overall accuracy of 92.1% for fruits affected by CBS and four other conditions (greasy spot, melanose, wind scar, and marketable). For leaves showing canker symptoms alongside four other conditions (control, greasy spot, melanoses, and scab), the VGG16 model with RBF SVM achieved an impressive overall accuracy of 93%.

According to Dhiman *et al*. [3], an effective citrus fruit disease prediction model can be developed using hyperspectral imaging (HSI) systems and features extracted through both deep and shallow convolutional neural networks, combined with machine learning classifiers. Their proposed model integrates edge computing with deep learning architectures, specifically CNN and long short-term memory (LSTM) networks. This model incorporates a feature-fusion subsystem, a down-sampling method, and an advanced feature-extraction mechanism to ensure accurate disease detection in citrus fruits while enabling substantial identification capabilities. The study utilized 2,950 labeled images of citrus fruits identified as affected by melanosis, scabs, cankers, black spots, or greening, drawn from online Kaggle and village datasets. Performance metrics such as precision, recall, F-measure, and support were employed to compare the proposed model with existing ones, assessed both with and without feature pruning. The research included two phases: the first involved experimental analysis using magnitude-based pruning (MBP), while the second combined MBP with post-quantization. The CNN-LSTM model, enhanced by these techniques, outperformed the current state-of-the-art CNN method, achieving accuracy rates of 97.18% and 98.25%, respectively. The CNN-LSTM Model hybrid model used for maize disease classification [19].

Uğuz *et al.* [8] introduced CitrusNet, a novel model based on CNNs for classifying damaged and abnormal citrus fruits. The study gathered 5,149 fruit images from citrus groves in Antalya, Turkey. Among the four CNN models tested, CitrusNet and ResNet50 yielded the best classification results. The second phase of their research evaluated five different CNN models for detecting two common diseases in Turkish citrus: alternaria alternata and thrips. Experimental results indicated that YOLOv5 and Mask R-CNN were the most effective in detecting citrus diseases, achieving an average precision (AP) of 0.99. Wang *et al.* [20] built a citrus yellow shoot disease recognition model based on the YOLOv5s and achives an accuracy of 91.3%

Saini *et al.* [21] proposed a deep CNN model for classifying citrus plant leaf and fruit diseases into seven categories. The model's performance was evaluated using optimizers such as Adam, stochastic gradient descent (SGD), and RMSprop, with Adam achieving the highest precision of 98.6%. The study also showed that data augmentation, along with variations in epochs, batch size, and dropout, improved model accuracy. This approach demonstrates the potential of AI in enabling fast and accurate plant disease detection. Arthi *et al.* [22] proposed a novel approach named duck optimization with enhanced capsule network (DOECN) based citrus disease detection for sustainable crop management (CDDCM) was proposed to detect and classify citrus diseases effectively. The method integrates preprocessing steps, uses DOECN for feature extraction and hyperparameter tuning, and employs a deep stacked sparse autoencoder (DSSAE) for classification. A CNN-SVM hybrid model was proposed for citrus disease detection, using CNN for feature extraction and SVM for classification [23]. With an accuracy of 92.34%, the model effectively identifies multiple citrus diseases, supporting precision agriculture and sustainable crop management.

Chowdhury *et al.* [24] developed a lightweight CNN model for citrus leaf disease detection and compared with pre-trained models like ResNet-50, VGG16, and DenseNet variants. Trained on an augmented dataset of 2800 images, the model achieved a 97.84% validation accuracy and 96% F1-score. Shastri *et al.* [7] proposed a new approach for reliable and automated disease identification using CNNs. By analyzing a substantial dataset of images depicting diseased citrus fruits and leaves, their suggested E-CNN model demonstrated exceptional results in both recognition and classification accuracy. Qiu *et al.* [25] explored semantic embedding methods were investigated for disease images and structured descriptive texts. Visual features of leaves were extracted using convolutional networks of varying depths, including VGG16, ResNet50, MobileNetV2, and ShuffleNetV2. Shermila *et al.* [26] proposed a tailored approach that integrates a CNN with an LSTM, achieving an efficiency of 96%.

2.1. Problem statement

Citrus diseases, caused by various fungal, bacterial, and viral infections, lead to significant financial losses in the citrus industry worldwide. Manual inspection methods are time-consuming, prone to errors, and require expert knowledge, making them inefficient for large-scale detection. Therefore, there is a critical need for automated systems that can accurately detect and classify multiple diseases in citrus plants to mitigate economic losses and ensure high-quality production. The problem is the inefficiency of traditional manual and non-optimized machine learning methods in detecting multiple citrus diseases, which can lead to economic losses and reduced production quality. Challenges in citrus disease detection: citrus diseases, caused by fungal, bacterial, and viral agents, are a major threat to global citrus production. Traditional methods of disease detection rely on manual observation, which is time-consuming and error-prone. The proposed method tackles the challenge by automating the detection process using image-based analysis, allowing early detection and minimizing crop losses.

3. PROPOSED SYSTEM

The paper proposes an EFOA integrated with a TCNN. The strategy behind using the EFOA is to optimize the hyper-parameters of the TCNN, allowing for better feature extraction and classification performance. This directly addresses the problem by improving the accuracy and sensitivity of disease detection. This research provides an innovative solution for automated citrus disease detection, with the EFOA-TCNN model outperforming conventional methods. The detailed analysis and comparison of models validate the robustness of the proposed approach, offering significant potential for real-world applications.

The research paves the way for future work in optimizing similar models for other plant diseases and agricultural challenges. Our research demonstrated that the EFOA-TCNN model significantly enhances the detection of citrus diseases compared to traditional methods. We found that the sensitivity of the EFOA-TCNN model reached 0.975, indicating a high true positive rate for identifying diseased samples.

3.1. Data description

Gathered in orange orchards in the northeastern Mexican states of San Luis Potosi, the collection contains 953 color photos of *Citrus* species leaves. Orange leaves in the dataset are categorized into 12 groups, as shown in Table 1. These groups include healthy, sick, nutrient deficient, and pests. In addition, Figures 1(a) to 1(l) displays examples of each anomaly, showing how the leaves' texture and color patterns change as predicted [27].

Table 1. Class-wise image delivery in the dataset

Class name	Abnormality type	# Images
Greasy spot	Disease	100
Fe	Iron deficiency	100
Mg	Magnesium lack	100
Zn	Zinc deficiency	100
Healthy	Not abnormal	100
HLB	Disease	43
Texas mite	Pest	100
Red scale	Pest	30
Red scale sequelae	Pest	100
Citrus leafminer	Pest	100

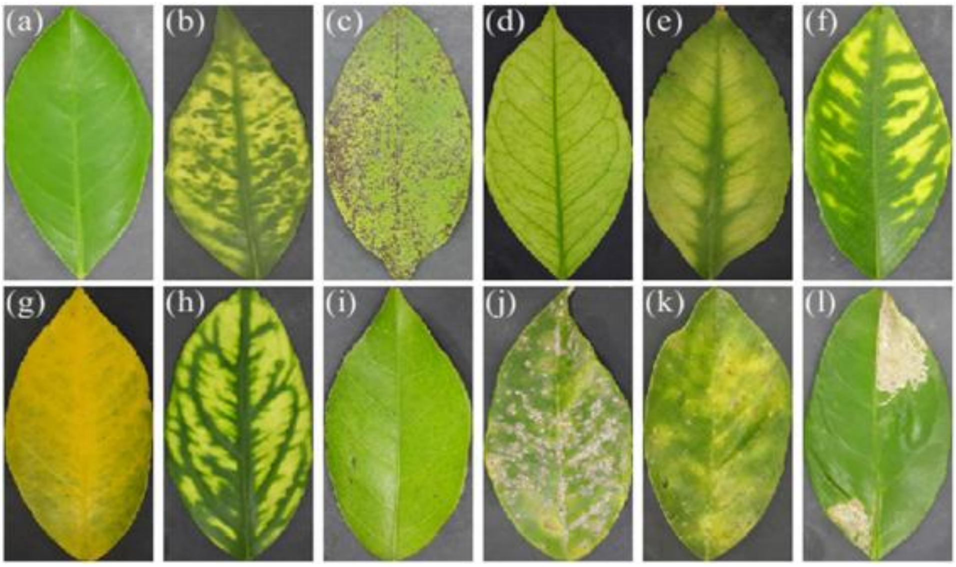


Figure 1. Here are the leaf samples included in the dataset: (a) strong, (b) huanglongbing, (c) greasy spot, (d) iron deficiency, (e) magnesium lack, (f) manganese deficiency, (g) nitrogen deficit, (h) zinc deficiency, (i) the citrus mite of Texas, (j) citrus leafminer, (k) red scale and its aftereffects, and (l) red scale itself

3.2. Classification using deep learning network construction

For the purpose of plant disease categorization, the EFOA-TCNN architecture is shown. There are three key aspects of the picture that the recommended deep CNN takes into account: If the size of the description pattern is equal filter will be able to identify it. Second, various portions of the input picture use of the same forms or patterns [27]. Convolving the whole source image is additional models. Finally, the geometry of the source picture is unaffected by down sampled pixels, which layer.

Two convolution layers make up the suggested EFOA-TCNN, with the pooling layers and the EL being directed by a third convolution layer. After that, the fully connected (FC) layer is paired with a SoftMax layer. By taking an average of the output, EL summarizes the feature maps of the final convolutional layer. Each feature map gets a value back from this, which is the same as the energy bank. This design not only reduces the number of layers, but it also uses less memory and computation time when learning texture functions, and it does it quite well. This performance-computing time trade-off is made possible by EL. To layer's data flow, this layer is added. Shortly following the last pooling layer, the EL's flattened output is routed to the concatenation layer. In order to impart information on the image's shape and texture to the completely linked layer, this connection generates a fresh, flattened vector. The mathematical computation of the convolution layer's output size is given by (1) as:

$$Output = \frac{I_a - I_b + 2S}{\varrho + 1} \tag{1}$$

where I_a besides I_b characterizes the input besides filter size correspondingly, S signifies the padding, besides ϱ is the stride worth.

Next, the first two layers of the three convolution layers are trained using a 5×5 kernel size, and their outputs are 16 and 32 channels, correspondingly. The third convolutional layer, which has 64 output channels besides a kernel size of 3×3, is used as a transitional layer to extract texture attributes [28]. The convolution layer may only learn 31,744 parameters, which are determined using the formula in (2) and (3):

$$\xi_v = \zeta_V \times (I_k \times \varrho + 1) \tag{2}$$

$$\xi_v = \zeta_V + X_k \times \varrho \times \zeta_V \tag{3}$$

where ξ_v means the CNN layer learnable limits, I_k characterizes the kernel size, then ζ_V signifies the channel sum.

The output of the input neuron is computed by each convolution layer. Its weight plus the least input field associated with it are multiplied by a dot product to get the computation. A 16-kernel output with dimensions of 32×32×16 is shaped by the initial convolution layer. According to (4), the first convolution layer neurons' output is:

$$S_\vartheta = \sum_\vartheta C_\vartheta \times T_\vartheta + P_\vartheta \tag{4}$$

Where S_ϑ represents the maps, C_ϑ characterizes the feature maps that were supplied, and T stands for the weighted map. The last layer energy description as its output. After the third convolution layer, energy layers are mixed criteria. Like a descriptor, it describes the texture in a similar way. In (5) provides relationship as:

$$EL(\xi, \vartheta) = \rho\left[\sum_{i=1}^{j} T_i^{\omega} \vartheta_i + P\right] \tag{5}$$

Where EL(ξ,ϑ) stands for the EL layer, j for the input influences, and T for the EL weighted vector. There are fewer parameters that can be learned because the connection among the EL and FC layers is narrower than the last traditional convolution layer connectivity [29]. Forward and backward propagation also allow EL to learn, and it remembers the energy data from the previous layer. In addition to enhancing the learning capability and simplifying the projected system, EL helps to decrease the vector size of layer. To determine the EL parameters that can be learned, use (6) as:

$$\xi_{EL} = \eta^m \times \eta^{m-1} \tag{6}$$

where ξ_{EL} is the EL learnable limits, η^m is the current neuron, besides η^{m-1} is the neuron.

Between the convolution the batch activation purpose is utilized to process. To eliminate the internal covariate shift, batch normalization is employed. The standard deviation and mean can be normalized to achieve this. In (7) and (8) are utilized in the bulk normalization procedure to determine the mean and variance.

$$\tau_Q = \frac{1}{n} \sum_i^n l_i \tag{7}$$

$$v_Q = \frac{1}{n} \times \sum_i^n \left(l_i - \tau_Q\right)^2 \tag{8}$$

Where τ_Q and v_Q characterizes the mean besides variance correspondingly, n is the size of l_i component of attributes. A value of 64 is used for n in our work. When using (9) to determine the batch normalization, the result is:

$$\lambda_i = \frac{\vartheta_{i-\tau}}{\sqrt{\nu^2+\phi}}\alpha + A \tag{9}$$

where α and A are the values of the initializable parameters for each output function. By plugging rectified linear unit (ReLU) into the activation function in (10) and then calculating the output of (11) as:

$$\lambda_{i,j,k} = max0, \vartheta_{i,j,k} \tag{10}$$

$$\lambda_{ReLU} = ReLU(Bnorm(Conv(w, x))) \tag{11}$$

Where $\lambda_{i,j,k}$ signifies features and $\vartheta_{i,j,k}$ stands for the input element's attribute. As a consequence of the control network's overfitting, the pooling layer subsequently shrinks the feature mappings, weights, and computations. In (12) is used to calculate the layer numerically:

$$M_{pool} = max\left(0, \sum_Q \vartheta^{k-1} T_\vartheta\right) \tag{12}$$

Where M_{pool} stands for the feature maps that will be output, ϑ for the feature maps that will be input, Q for the size of the pooling, and T for the maximum layer for the vector. Two max layers, each with a 2×2 kernel scope, are utilized in this study.

For each weighted update, the dropout layer removes a fraction of randomly chosen parameters in order to prevent overfitting of the training data [30]. In order to prevent training data from being overfit, drop editing is employed in to continually eliminate a parameter. Over-compatibility of training data is a problem for FC layers because they have the most parameters in the network. Because of this, the dropout layer is decided upon subsequent to the FC layer. A classifier that makes use of the loss function is the SoftMax layer. For SoftMax, the possible outcomes might take on values between zero and one. In (13) the loss function is expressed mathematically as:

$$k_l = \delta_j + log \sum_i exp(\delta_i) \tag{13}$$

Where k_l stands for the overall loss and δ_j with the i-th vector element's class d. As shown in (14) using the SoftMax function, the classifier's goal is to minimize the probability discrepancy between the actual and estimated labels.

$$\lambda_i = \frac{exp^{\delta_j}}{\sum_i exp(\delta_j)} \tag{14}$$

When this step is finished, EFOA-TCNN moves on to the hyper-parameter tuning process, which will be described in the subsection that follows. As you can see from Table 2, the input and output dimensions of the projected network are fully labelled.

Table 2. Proposed EFOA-TCNN architecture layers

Types	Padding	Kernel size to form each feature map	Stride	Output size	Input size
Convolutional layer 1	[1 1 1 1]	5×5	[1 1]	62×62×16	64×64×1
Max pooling layer 1	[1 1 1 1]	2×2	[2 2]	32×32×16	62×62×16
Convolutional layer 2	[1 1 1 1]	5×5	[1 1]	30×30×32	32×32×16
ReLU					
Max pooling layer 2	[1 1 1 1]	2×2	[2 2]	16×16×32	30×30×32
Convolutional layer 3	[1 1 1 1]	3×3	[1 1]	16×16×64	16×16×32
ReLU					
EL	-	-	-	128×1	16×16×64
Dropout	-	-	-	128×1	128×1
FC1	-	-	-	1024×1	128×1
Dropout	-	-	-	1024×1	1024×1
FC2	-	-	-	2×1	1024×1
SoftMax layer	-	-	-	-	-
Classification layer	-	-	-	-	-

3.2.1. Fine-tuning using EFOA

This section begins by looking into the origins of logarithmic spiral pathways. After that, an adaptable switch (ratio) is built to strike the right combination of exploration and exploitation based on variations in the values of expressions on the face. EFOA is used for optimizing the hyper-parameters of the TCNN model, such as the learning rate, batch size, number of filters, and dropout rates. Optimization of these hyper-parameters is critical to achieving the best possible model performance, as improper hyper-parameter tuning can lead to poor generalization or overfitting. In EFOA, each fruit fly represents a candidate solution (a set of hyper-parameters), and the algorithm explores the search space to find the best combination of parameters that minimizes the classification error on the validation dataset.

The search process in EFOA involves both local and global search phases to balance exploration and exploitation. The local search refines the search around promising solutions, while the global search ensures the algorithm does not get stuck in local minima. The enhanced version of FOA introduces improvements in its search strategies, including adaptive parameter tuning and multi-dimensional exploration, which help in faster convergence to the optimal hyper-parameters. The integration of a new adaptable switch (ratio) and an enhanced FA is then shown.

3.2.2. Design of the logarithmic spiral path

The Sphere function $f(x) = \sum_{i=1}^{d} X_i^2$ and Schwefel function $f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} X_j)^2$(xi varies between -100 to 100, and d is set to 10) are chosen as the benchmark functions. Our research involves running tests 50 times to calculate the mean performance. By doing this, the final results are less affected by unpredictability in population position initialization. In these 50 runs, the appropriate variables for every algorithm are all given the same value. The quantity of searchers is 15, $\alpha = 0.2$, $\beta_0 = 1 \, and \, \gamma = 1$.

The potential for local space utilization has gone unnoticed. How to sustain exploration and exploitation of firefly is an intriguing topic in the search process. When the problem is solved, the optimizer's effectiveness will increase and the computational load will decrease. We solve this conundrum by considering large raptors that travel in a logarithmic spiral, like peregrine falcons in search of food. This strategy is based on experimental biology. We've also noticed a similar flight pattern for fireflies at night. The logarithmic spiral path is one technique that could be used to improve FA exploitation. The design of a novel position updating method based on this is how the logarithmic spiral looks,

$$X_{i,t+1} = X_{i,t} + \beta_0 . e^{-\gamma . r_{ij}^2} . (X_{j,t} - X_{i,t}) \otimes e^{b.l} \otimes \cos(2\pi . l) \tag{15}$$

In (15), l is an even random vector in d dimensions in $[-1, 1]^d$; the logarithmic spiral's shape is determined by the constant b, which has a default value of 1. To describe the position update methodology in (16), two causes determine this position variation: logarithmic spiral paths and brightness intensity. The coefficients in mathematics represent the latter $e^{b.l} \otimes \cos(2\pi . l)$.

3.2.3. Adaptive switch design

In order to equalize both exploring and exploiting modes, this research provides the adaptive switch (ratio) technique. Which approach will be employed in the following iteration:

$$\begin{cases} Exploration\ Mode\ for\ Global\ Search, if\ u > R_{t,} \\ Exploitation\ Mode\ for\ Local\ Search, if\ u \leq R_t, \end{cases} \tag{16}$$

Where u is a generated number randomly with uniform distribution [0, 1] and the R_t is computed in the previous iteration. The exploitation method must be selected with a greater probability than a mode of exploration in order to hasten the optimizer's convergence. Thus, we describe the changeover R_(t+1) ranging from [0.5, 1]. Setting the beginning ratio to 0.5 results in the value of the adaptable switch:

$$R_{t+1} = \begin{cases} \dfrac{1}{1+\exp\left(-\frac{f_t^*}{f_{t-1}^*}\right)}, \lfloor \lg|f_t^*| \rfloor \neq \lfloor \lg|f_{t-1}^*| \rfloor, \\ \dfrac{1}{1+\exp\left(-\dfrac{f_t^* - \theta . \left\lfloor \frac{f_t^*}{\theta} \right\rfloor}{f_{t-1}^* - \theta . \left\lfloor \frac{f_{t-1}^*}{\theta} \right\rfloor}\right)}, else \end{cases} \tag{17}$$

In (17), f_t^* at the *t*-th iteration, is the best function's fitness value; $lg(\cdot) = log_{10}(\cdot)$; $\lfloor \cdot \rfloor$ is the floor's purpose. The calculation of the (18) is done using the formula by the adaptable scale parameter threshold:

$$\theta = 10^{\lfloor \lg|f_t^* - f_{t-1}^*| \rfloor + 1} \tag{18}$$

We shall examine three situations in order to comprehend how this formula works in greater detail:

- The first condition, $f_t^* \gg f_{t-1}^*$. As a result, there is a sizable gap between two iterations. The process of optimisation now has a fresh, better option thanks to search agents. Hence, an adaptive ratio R_{t+1} becomes 1 and the next iteration's a mode of exploration is selected;
- If $f_t^* \ll f_{t-1}^*$, then shift to exploration with regard to of performance degradation. The flexible ratio R_{t+1} will be 0.5, and we'll make using a mode of exploration more likely the following searches.
- The final prerequisite is $\lfloor \lg | f_t^* | \rfloor = \lfloor \lg | f_{t-1}^* | \rfloor$, imply the discovery of a closest minimum. By the item, we adjust the ratio by (19) to increase the sensitivity of the adaptive switch. In this scenario, there will be a high likelihood of search agents escaping potential traps.

$$\frac{f_t^* - \theta . \left\lfloor \frac{f_t^*}{\theta} \right\rfloor}{f_{t-1}^* - \theta . \left\lfloor \frac{f_{t-1}^*}{\theta} \right\rfloor} \tag{19}$$

The scaling factor $\lfloor \lg | f_t^* - f_{t-1}^* | \rfloor$ can automatically detect the finding status. Thus, our adaptive switch improves convergence even more. Following that, the logistic function is used to convert the variation to a probability. The adaptable switch ratio is then determined. Based on the idea of a flexible switch layout, we discover that this particular switch has a greater capacity: agents that search activity has increased in choosing a mode of exploration when there is an interruption during the process of searching to ensure the optimization algorithm to find a greater ideal.

3.2.4. Firefly's updated algorithm

The updated location formula is now displayed as (20):

$$X_{i,t+1} = \begin{cases} X_{i,t} + \beta_0 . e^{-\gamma . r_{ij}^2} . \left(X_{j,t} - X_{i,t} \right) + \alpha . sign[rand - 0.5] \otimes Levy, u > R_{t,} \\ X_{i,t} + \beta_0 . e^{-\gamma . r_{ij}^2} . \left(X_{j,t} - X_{i,t} \right) \otimes e^{b.l} \otimes \cos(2\pi . l), u \le R_t . \end{cases} \tag{20}$$

Our revised firefly method, termed the adaptable logarithmic spiral firefly algorithm, which integrates the advantages of FA with a modernized the logarithmic spiral pathway controlled by a smart adaptive switch. The pseudo-code demonstrates the modifications that were made for the conventional FA framework to accommodate our flexible switch. The most recent value of the fitness function f_t^* is logged to configure the flexible switch. We will employ a traditional technique in which the value of the switch is fixed at 0.5 to assess the effectiveness of the capability of the adaptive switch and the logarithm spiral path. In (21) is the method for updating position:

$$X_{i,t+1} = \begin{cases} X_{i,t} + \beta_0 . e^{-\gamma . r_{ij}^2} . \left(X_{j,t} - X_{i,t} \right) + \alpha . sign[rand - 0.5] \otimes Levy, u > 0.5, \\ X_{i,t} + \beta_0 . e^{-\gamma . r_{ij}^2} . \left(X_{j,t} - X_{i,t} \right) \otimes e^{b.l} \otimes \cos(2\pi . l), u \le 0.5. \end{cases} \tag{21}$$

4. RESULTS AND DISCUSSION

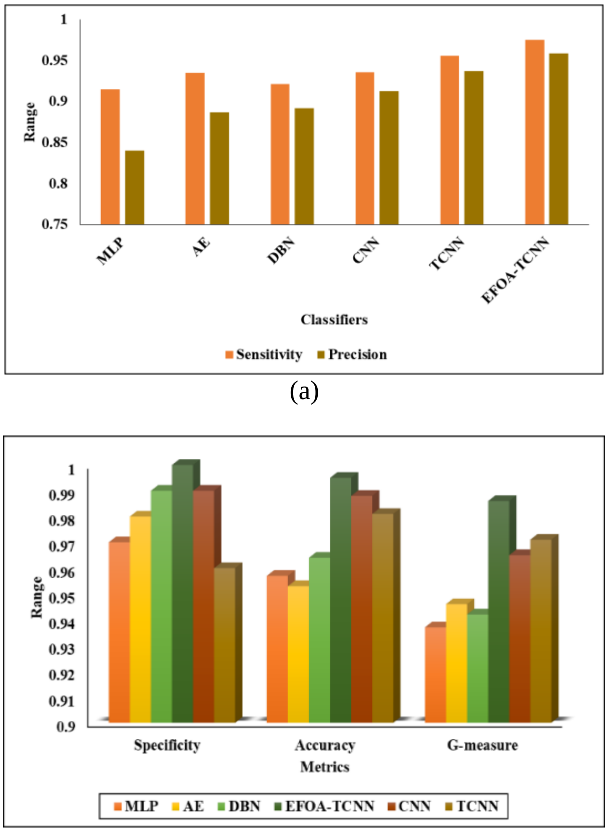
To conduct the research, an Intel Core i5-7200 CPU besides 8 GB of internal memory is utilized. The processor is accomplished of running at 2.7 GHz. Devoted User Interface (UI) besides Jupyter Notebook (Python 3.7) perform the operations on Windows 10, a 64-bit operating system natural setting.

4.1. Validation analysis of proposed model with existing procedures

Table 3 provides the experimental investigation of predictable faultless with existing procedures in terms of different metrics. In Table 3 means that the validation study of projected faultless with existing techniques. In this investigation, the MLP technique attained sensitivity as 0.915 besides specificity as 0.97 and accuracy of 0.957 and F-measure of 0.937 and precision as 0.840 correspondingly. Then the autoencoder technique attained sensitivity as 0.935, specificity as 0.98, accuracy of 0.953, F-measure of 0.946, and precision as 0.887 correspondingly. Then the deep belief network (DBN) technique attained sensitivity as 0.921, specificity as 0.99, accuracy of 0.964, F-measure of 0.942, and precision as 0.892 correspondingly. Then the CNN technique attained sensitivity as 0.936, specificity as 0.99, accuracy of 0.988, F-measure of 0.965, and precision as 0.913 correspondingly. Then the TCNN technique attained sensitivity as 0.956, specificity as 0.96, accuracy of 0.981, F-measure of 0.971, and precision as 0.937 correspondingly. Then the EFOA-TCNN technique attained sensitivity as 0.975, specificity as 1.00, accuracy of 0.995, F-measure of 0.986, and precision as 0.959 congruently. The Figures 2(a) and 2(b) shows the visual representation of projected classical and comparison analysis of proposed with existing models for plant disease detection respectively.

Table 3. Validation analysis of proposed textbook with existing techniques

Classifiers	Sensitivity	Specificity	Accuracy	F-measure	Precision
MLP	0.915	0.97	0.957	0.937	0.840
AE	0.935	0.98	0.953	0.946	0.887
DBN	0.921	0.99	0.964	0.942	0.892
CNN	0.936	0.99	0.988	0.965	0.913
TCNN	0.956	0.96	0.981	0.971	0.937
EFOA-TCNN	0.975	1.00	0.995	0.986	0.959



(a)

(b)

Figure 2. Performance and comparative analysis, (a) visual representation of projected classical and (b) comparison analysis of proposed with existing models for plant disease detection

The improvement in precision and F-measure indicates better balance in handling both true positives and false positives. This table provides a concise yet comprehensive comparison, clearly demonstrating the effectiveness of the proposed model. This approach keeps the manuscript manageable while showcasing the significant performance gains of the EFOA-TCNN.

Validation analysis comparing the performance of the proposed EFOA-TCNN model with existing techniques, specifically the reference model [4], is presented in Table 4. The analysis highlights key metrics such as sensitivity, specificity, accuracy, F-measure, and precision. The reference model achieved a sensitivity of 95%, specificity of 96%, accuracy of 96%, F-measure of 95%, and precision of 96%. In contrast, the EFOA-TCNN model outperformed the reference model with a sensitivity of 97.5%, perfect specificity of 100%, an accuracy of 99.5%, an F-measure of 98.6%, and a precision of 95.9%. This indicates that the EFOA-TCNN model demonstrates superior performance across all metrics, significantly enhancing detection capabilities in the context of the study. Such findings underline the effectiveness of integrating optimization techniques like EFOA in deep learning models for improved classification tasks, as supported by various studies in the field of machine learning and image processing.

Table 4. Validation analysis of proposed textbook with existing techniques

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)	F-measure (%)	Precision (%)
Reference [4]	95	96	96	95	96
EFOA-TCNN	0.975	1.00	0.995	0.986	0.959

4.2. Discussion

This research presents the development of an optimized model for citrus disease detection, focusing on improving accuracy and efficiency through the integration of the EFOA with a TCNN. The experimental results demonstrate that the proposed EFOA-TCNN model significantly outperforms traditional machine learning and deep learning techniques in detecting diseases from images of orange leaves. The EFOA-TCNN model achieved remarkable performance across key metrics, with an accuracy of 0.995, sensitivity of 0.975, and precision of 0.959. These results indicate the model's high capacity to correctly identify diseased leaves while minimizing false positives and negatives. The model's use of an energy layer in place of a traditional pooling layer allowed for better feature extraction, preserving crucial image details that contributed to its superior classification performance. This modification, combined with the hyper-parameter tuning provided by EFOA, resulted in improved sensitivity and accuracy compared to the baseline CNN model and other classifiers such as MLP and AE.

The optimization of TCNN parameters using EFOA played a critical role in the model’s success. By fine-tuning hyper-parameters such as learning rate, filter size, and dropout rates, the EFOA significantly enhanced the TCNN’s ability to capture intricate disease patterns in the input images. This demonstrates the importance of applying optimization algorithms to improve deep learning architectures for specific applications, such as agricultural disease detection.

In comparison to other classifiers, the proposed EFOA-TCNN model consistently outperformed MLP, AE, DBN, and CNN in all key metrics. The closest competing model, CNN, achieved an accuracy of 0.988, but the EFOA-TCNN model still surpassed it with a notable 0.995 accuracy. The improvement in sensitivity (0.975) and specificity (1.00) shows the model's exceptional balance between correctly identifying diseased leaves and avoiding false positives, which is critical in real-world agricultural settings where over-diagnosing diseases can lead to unnecessary treatments and costs.

The research findings provide strong evidence that the proposed EFOA-TCNN model is a highly effective tool for automated citrus disease detection. By integrating the EFOA, the model significantly outperforms conventional techniques and demonstrates potential for real-world implementation. The inclusion of optimization algorithms such as EFOA helps bridge the gap between theoretical model development and practical applications, making this approach scalable for broader.

5. CONCLUSION

Deep learning has achieved significant advancements in agriculture, particularly in automating the identification of plant diseases while minimizing the reliance on extensive human involvement. This research focused on developing an automated system for disease identification in citrus leaves through deep learning and optimal feature selection. Initially, data augmentation was employed to expand the dataset, enhancing the robustness of deep learning representations. To further improve the precision and efficiency of plant disease detection, this study introduced a novel approach that integrates multiple methodologies. Specifically, a hyperparameter-tuned TCNN was utilized to enhance classification performance. This approach streamlined the architecture by replacing conventional pooling layers with just three energy layers, thereby making the model more accessible and effective. The use of the EFOA facilitated efficient hyperparameter tuning, which contributed to improved model performance. The extracted features were subsequently categorized using various supervised learning algorithms. Although the fusion of selected deep features significantly enhanced detection accuracy, it was accompanied by a trade-off in increased computational time. Looking ahead, this research lays the groundwork for implementing real-time monitoring systems for citrus plant disease detection. Future work will explore the integration of this model with internet of things (IoT) devices and edge computing systems, enabling real-time disease detection in agricultural fields. Such integration would facilitate rapid on-site analysis of plant health without relying on centralized servers, allowing farmers to receive immediate feedback and make timely interventions. Moreover, future studies could focus on enhancing the model's efficiency through optimization techniques that reduce computational overhead while maintaining accuracy. The exploration of adaptive learning systems that continuously improve with new data can also be a key area for development. This research underscores the potential of AI-powered disease detection systems in agriculture and sets the stage for further advancements in precision farming technologies. By combining advanced image analysis and optimization techniques, we can unlock new avenues for sustainable crop protection methods, ultimately benefiting both producers and consumers.

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
Anoop Ganadalu Lingaraju		✓		✓	✓	✓		✓	✓		✓	✓	✓	
Asha Mangala Shankaregowda	✓	✓	✓		✓		✓		✓	✓	✓			
Babu Kumar Sathiyamurthy	✓	✓		✓	✓	✓				✓		✓		
Santhrupth Budanoor Channegowda			✓		✓	✓	✓		✓	✓	✓	✓		
Shruti Jalapur					✓		✓		✓	✓				
Chaitra Palahalli Chennakeshava	✓		✓	✓		✓	✓	✓		✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

The authors confirm that all data used in this study, including images of tomato leaves, were publicly available. No human participants or personal data were involved in this study.

ETHICAL APPROVAL

The authors affirm that the research was conducted in accordance with ethical standards and institutional guidelines.

DATA AVAILABILITY

The data that support the findings of this study are openly available at https://zenodo.org/records/8294078.

REFERENCES

[1] P. Dhiman *et al.*, "A novel deep learning model for detection of severity level of the disease in citrus fruits," *Electronics*, vol. 11, no. 3, 2022, doi: 10.3390/electronics11030495.

[2] P. K. Pareek, I. M. Ramya, N. B. Jagadeesh, and M. H. LeenaShruthi, "Clustering based segmentation with 1D-CNN model for grape fruit disease detection," *2023 IEEE International Conference on Integrated Circuits and Communication Systems, ICICACS 2023*, 2023, doi: 10.1109/ICICACS57338.2023.10099916.

[3] P. Dhiman, A. Kaur, Y. Hamid, E. Alabdulkreem, H. Elmannai, and N. Ababneh, "Smart disease detection system for citrus fruits using deep learning with edge computing," *Sustainability*, vol. 15, no. 5, 2023, doi: 10.3390/su15054576.

[4] H. Çetiner, "Citrus disease detection and classification using based on convolution deep neural network," *Microprocessors and Microsystems*, vol. 95, 2022, doi: 10.1016/j.micpro.2022.104687.

[5] N. Ghanei Ghooshkhaneh and K. Mollazade, "Optical techniques for fungal disease detection in citrus fruit: a review," *Food and Bioprocess Technology*, vol. 16, no. 8, pp. 1668–1689, 2023, doi: 10.1007/s11947-023-03005-4.

[6] G. L. Anoop and C. Nandini, "Classification of rice blast, brown spot, leaf blight and hispa paddy leaf diseases in transformed YUV color space," *NeuroQuantology*, vol. 20, pp. 4172–4182, 2022.

[7] R. Shastri, A. Chaturvedi, B. Mouleswararao, S. Varalakshmi, G. N. R. Prasad, and M. K. Ram, "An automatic detection of citrus fruits and leaves diseases using enhanced convolutional neural network," *Remote Sensing in Earth Systems Sciences*, vol. 6, no. 3–4, pp. 123–134, 2023, doi: 10.1007/s41976-023-00086-9.

[8] S. Uğuz, G. Şikaroğlu, and A. Yağız, “Disease detection and physical disorders classification for citrus fruit images using convolutional neural network,” *Journal of Food Measurement and Characterization*, vol. 17, no. 3, pp. 2353–2362, 2023, doi: 10.1007/s11694-022-01795-3.

[9] R. K. Ranjan and V. Kumar, “A systematic review on fruit fly optimization algorithm and its applications,” *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13015–13069, 2023, doi: 10.1007/s10462-023-10451-1.

[10] S. Palei, S. K. Behera, and P. K. Sethy, “A systematic review of citrus disease perceptions and fruit grading using machine vision,” *Procedia Computer Science*, vol. 218, pp. 2504–2519, 2022, doi: 10.1016/j.procs.2023.01.225.

[11] P. K. Pareek, C. Srinivas, S. Nayana, and D. S. Manasa, “Prediction of floods in kerala using hybrid model of CNN and LSTM,” *2023 IEEE International Conference on Integrated Circuits and Communication Systems, ICICACS 2023*, 2023, doi: 10.1109/ICICACS57338.2023.10099867.

[12] P. Dhiman, A. Kaur, V. R. Balasaraswathi, Y. Gulzar, A. A. Alwan, and Y. Hamid, “Image acquisition, preprocessing and classification of citrus fruit diseases: a systematic literature review,” *Sustainability*, vol. 15, no. 12, 2023, doi: 10.3390/su15129643.

[13] A. Elaraby, W. Hamdy, and S. Alanazi, “Classification of citrus diseases using optimization deep learning approach,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/9153207.

[14] S. F. Syed-Ab-Rahman, M. H. Hesamian, and M. Prasad, “Citrus disease detection and classification using end-to-end anchor-based deep learning model,” *Applied Intelligence*, vol. 52, no. 1, pp. 927–938, 2022, doi: 10.1007/s10489-021-02452-w.

[15] P. K. Yadav, T. Burks, Q. Frederick, J. Qin, M. Kim, and M. A. Ritenour, “Citrus disease detection using convolution neural network generated features and Softmax classifier on hyperspectral image data,” *Frontiers in Plant Science*, vol. 13, 2022, doi: 10.3389/fpls.2022.1043712.

[16] R. Hadipour-Rokni, E. Askari Asli-Ardeh, A. Jahanbakhshi, I. Esmaili paeen-Afrakoti, and S. Sabzi, “Intelligent detection of citrus fruit pests using machine vision system and convolutional neural network through transfer learning technique,” *Computers in Biology and Medicine*, vol. 155, 2023, doi: 10.1016/j.compbiomed.2023.106611.

[17] N. Butt, M. Munwar, I. Iftikhar, H. Akbar, and U. Khadam, “Citrus diseases detection using deep learning,” *Journal of Computing & Biomedical Informatics*, vol. 06, no. 02, pp. 23–33, 2024.

[18] P. K. Yadav *et al.*, “Automated classification of citrus disease on fruits and leaves using convolutional neural network generated features from hyperspectral images and machine learning classifiers,” *Journal of Applied Remote Sensing*, vol. 18, no. 1, 2024, doi: 10.1117/1.jrs.18.014512.

[19] N. Thapliyal, M. Aeri, A. Kumar, V. Kukreja, and R. Sharma, “Combining spatial and temporal analysis: a CNN-LSTM hybrid model for maize disease classification,” *Proceedings-International Conference on Computing, Power, and Communication Technologies, IC2PCT 2024*, pp. 1529–1533, 2024, doi: 10.1109/IC2PCT60090.2024.10486456.

[20] S. Wang, Y. Zhang, and R. Li, “Citrus yellow shoot disease detection based on YOLOV5,” *Highlights in Science, Engineering and Technology*, vol. 39, pp. 1291–1300, 2023, doi: 10.54097/hset.v39i.6758.

[21] A. K. Saini, R. Bhatnagar, and D. K. Srivastava, “Citrus fruits-leaves diseases detection and classification with optimized deep CNN,” *Lecture Notes in Networks and Systems*, vol. 812, pp. 89–99, 2024, doi: 10.1007/978-981-99-8031-4_9.

[22] A. Arthi *et al.*, “Duck optimization with enhanced capsule network based citrus disease detection for sustainable crop management,” *Sustainable Energy Technologies and Assessments*, vol. 58, 2023, doi: 10.1016/j.seta.2023.103355.

[23] V. Kumar, D. Banerjee, R. Chauhan, S. Kukreti, and K. S. Gill, “Optimizing citrus disease prediction: a hybrid CNN-SVM approach for enhanced accuracy,” *2024 3rd International Conference for Innovation in Technology, INOCON 2024*, 2024, doi: 10.1109/INOCON60754.2024.10511309.

[24] M. F. Chowdhury, A. Nondi, S. U. Akhter, T. B. Pathan, and D. Z. Karim, “Optimizing citrus leaf disease detection: an efficient custom CNN leveraging efficient training parameters,” *6th IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2024*, pp. 579–584, 2024, doi: 10.1109/IICAIET62352.2024.10730688.

[25] X. Qiu *et al.*, “Detection of citrus diseases in complex backgrounds based on image–text multimodal fusion and knowledge assistance,” *Frontiers in Plant Science*, vol. 14, 2023, doi: 10.3389/fpls.2023.1280365.

[26] J. P. Shermila, A. Victor, S. O. Manoj, and E. A. Devi, “Automatic detection and classification of disease in citrus fruit and leaves using a customized CNN based model,” *Boletin Latinoamericano y del Caribe de Plantas Medicinales y Aromaticas*, vol. 23, no. 2, pp. 180–198, 2024, doi: 10.37360/blacpma.24.23.2.13.

[27] W. Gómez-Flores, J. J. Garza-Saldaña, and S. E. Varela-Fuentes, “CitrusUAT: a dataset of orange Citrus sinensis leaves for abnormality detection using image analysis techniques,” *Data in Brief*, vol. 52, 2024, doi: 10.1016/j.dib.2023.109908.

[28] S. Baswaraju, V. U. Maheswari, K. K. Chennam, A. Thirumalraj, M. V. V. P. Kantipudi, and R. Aluvalu, “Future food production prediction using AROA based hybrid deep learning model in agri-sector,” *Human-Centric Intelligent Systems*, vol. 3, no. 4, pp. 521–536, 2023, doi: 10.1007/s44230-023-00046-y.

[29] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.

[30] I. Salehin and D. K. Kang, “A review on dropout regularization approaches for deep neural networks within the scholarly domain,” *Electronics*, vol. 12, no. 14, 2023, doi: 10.3390/electronics12143106.

BIOGRAPHIES OF AUTHORS

Anoop Ganadalu Lingaraju holds a Ph.D. in Computer and Information Science from Visvesvaraya Technological University, Belagavi, Karnataka, India. Currently, he is an Assistant professor at the Department of Computer Science and Engineering, CHRIST University. His research areas include digital image processing, image classification, video processing, pattern recognition, neural network, deep learning, and machine learning. He has also co-authored the textbook quantum computing: a beginner's journey to quantum computing. He can be contacted at email: gl.anoop1@gmail.com.

Asha Mangala Shankaregowda is currently working as an Assistant Professor in the Department of Computer Science and Engineering at CHRIST (Deemed to be University) Bangalore. She received her B.E. degree in ISE from Visvesvaraya Technological University, Belgaum and M.Tech. degree in CCT from Mysore University. She is currently pursuing her Ph.D. at Visvesvaraya Technological University (VTU), Belgaum. Her research interests include image processing and deep learning, where she actively explores advancements in these fields. With a strong academic background and a passion for teaching, she contributes significantly to both research and education in computer science. She can be contacted at email: ashagowda05@gmail.com.

Babu Kumar Sathiyamurthy is currently working as an Assistant Professor in the Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore. He has 10 years of teaching experience and 3 years in research. He is a Certified AWS Associate Architect and also a Life member of ISTE. He received his B.E. degree in ISE and M.Tech. degree in CNE from Visvesvaraya University, Belgaum. He is currently pursuing his Ph.D. at RVITM, a research center by VTU. His research interests include image processing, deep learning, and cloud computing. He has published 12 technical papers in the fields of image processing, deep learning, and cloud computing at international conferences, SCI, Scopus, and indexed journals. He can be contacted at email: babukumar.sbk@gmail.com.

Santhrupth Budanoor Channegowda is an Assistant Professor in the Department of Computer Science and Engineering at CHRIST University, Kengeri Campus. He holds a UG and PG from Vishveshwariya Technological University and has over 12 years of experience in the field. His expertise lies in image processing and machine learning, and he is currently pursuing a Ph.D. in this domain. In addition to teaching, he actively mentor’s students and is involved in several academic initiatives, including extended realities and related technologies. He can be contacted at email: santhrupth23@gmail.com.

Shruti Jalapur is currently working as an Assistant Professor at the Department of Computer Science and Engineering, CHRIST University, Bangalore. Her research interests include the internet of things, machine learning, deep learning, and parallelism. She can be contacted at email: shruti.jalapur@christuniversity.in.

Chaitra Palahalli Chennakeshava is an Assistant Professor at CHRIST University, Bangalore, and is currently pursuing her Ph.D. in Computer Science. She holds an M.Tech. from Dr. Ambedkar Institute of Technology and B.E. from B.G.S.I.T. She has presented research on topics like cancer detection and machine learning at national and international conferences. She actively participates in faculty development programs and has a strong interest in artificial intelligence and data science. She can be contacted at email: chaitra.pc@christuniversity.in.