IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 November 2023, accepted 4 January 2024, date of publication 9 January 2024, date of current version 22 January 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3351912*

RESEARCH ARTICLE

# Palm Leaf Health Management: A Hybrid Approach for Automated Disease Detection and Therapy Enhancement

**S. M. NURUZZAMAN NOBEL[1], MD. ASIF IMRAN[1], NAHIDA ZAMAN BINA[1], MD. MOHSIN KABIR[2], (Member, IEEE), MEJDL SAFRAN[3], SULTAN ALFARHOOD[3], AND M. F. MRIDHA[4], (Senior Member, IEEE)**

[1]Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh
[2]Superior Polytechnic School, University of Girona, 17071 Girona, Spain
[3]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Computer Science, American International University-Bangladesh, Dhaka 1229, Bangladesh

Corresponding author: Mejdl Safran (mejdl@ksu.edu.sa)

This research is funded by the Researchers Supporting Project Number (RSPD2024R1027), King Saud University, Riyadh, Saudi Arabia.

**ABSTRACT** Deep Learning and computer vision have become potent agricultural technologies in recent years. These technologies are essential for identifying hazardous plant leaf diseases, significantly impacting crop quality and productivity. The precise distinction between healthy and damaged palm leaves is at the core of this research. Our study marks a significant improvement in the area by introducing a novel method for identifying palm leaf disease using a hybrid model. Our hybrid model's central component combines the Efficient Channel Attention Network (ECA-Net) with reliable transfer learning techniques utilizing ResNet50 and DenseNet201. In addition to improving disease diagnosis accuracy, this fusion sets a new performance bar compared to earlier models. Our hybrid model maintains a validation accuracy of 98.67% while achieving an amazing 99.54% training accuracy in precisely identifying diseases. Compared to its contemporaries, it also performs exceptionally well in F1 score values, highlighting its remarkable prowess in agricultural technology. This research provides a breakthrough method for disease detection in palm leaves. It will revolutionize the agriculture sector.

**INDEX TERMS** Palm leaf disease, deep learning, CNN, transfer learning, hybrid model, K-fold, disease detection, automated diagnosis, agricultural sustainability.

## I. INTRODUCTION

Palm leaves, or Phoenix dactylifera as it is formally named, play a crucial role in many different cultures, economies, and ecosystems. These robust and adaptable leaves have been used for various tasks for ages, enhancing people's subsistence and way of life worldwide. The date palm trees and their foliage are notably harmed by the "Dubas" insect, also known as the Dubas bug (Ommatissus lybicus). This bug is particularly significant in areas where date palm farming is a substantial agricultural activity. A critical component of precision farming is the identification and classification of plant diseases [1]. One of the difficulties farmers face is identifying the illnesses that afflict date palms and figuring out the extent of infestation, especially when the symptoms are incredibly similar and picking the right strategy and treatment is challenging and, frequently, incorrect. Agriculture's automated detection and diagnosis of plant pests and diseases has undergone a revolutionary change because of the development of machine vision and deep learning [2]. The sap-sucking pests known as palm date scale feed on the leaflets of palm trees and establish colonies

The associate editor coordinating the review of this manuscript and approving it for publication was Liandong Zhu.

**FIGURE 1.** **Palm leaf with Dubas Bug insect.**

there [3]. Palm leaf diseases seriously endanger the health and productivity of palm trees. Different fungi, bacteria, viruses, and pests can all cause illnesses in palm leaves, as shown in Figure 1.

Dubas Bug (Obmatissus lybicus), shown in 1, often known as the Dubas insect, is a severe pest of palm trees, especially the date palm. By puncturing palm fronds and sucking out sap, these tiny sap-eating insects can seriously harm plants by drying out and deforming leaves. They can affect the overall health of the palm tree and decrease fruit yield in cases of severe infestation. In areas where Dubas bugs are common, effective pest management and routine monitoring are crucial for maintaining the health of date palm orchards.

**There are several ways to control palm leaf diseases. These include:**

- Cultural controls: These policies cover things like good hygiene, crop rotation, and resistant cultivars.
- Chemical remedies: Fungicides, bactericides, and insecticides can be used to combat illnesses that affect palm leaves.
- Biological controls: Pest insects that attack palm leaves can be managed biologically with the help of parasites and insect predators.

Palm trees' unique fronds and critical ecological functions make them stand out in various environments and landscapes. These trees provide vital resources for food, shelter, and economic value for many areas of the world. However, several illnesses that can seriously restrict palm tree growth, reduce agricultural yields, and even cause tree mortality continually threaten their health and productivity [4]. Prompt and precise detection of these diseases is essential for efficient disease control and maintenance of palm tree populations. Agronomists or plant pathologists with experience visually evaluate palm trees as part of traditional disease diagnosis and assessment methods. This strategy has drawbacks, including the possibility of human error, inconsistent results, and the need for specialized knowledge, even though it has some potential for success. Additionally, the spread of palm trees across the globe has increased the demand for automated and scalable solutions to quickly and reliably diagnose and categorize palm leaf diseases [5]. Recent developments in machine learning and intense learning have demonstrated fantastic ability in several areas, including computer vision and medical diagnosis. It has been shown that these methods offer a great deal of potential for automating the identification and classification of plant diseases, especially those that impact palm palms.

For the prediction of palm plant diseases, there are various benefits to combining an Efficient Channel Attention (ECA-Net) with a hybrid model like DenseNet201 and ResNet50. Convolutional neural networks (CNNs) are a powerful tool for computer vision and deep learning, quickly evolving fields. Researchers and practitioners are constantly looking for new approaches to improve the performance of CNNs [6]. The creation of hybrid models, which combine the advantages of many architectural components to produce superior outcomes, has been one of the most significant movements in recent years. In this research [7], the Efficient Channel Attention Network (ECA-Net), ResNet50, and DenseNet201 are three well-known architectures that are combined in a novel hybrid model is presented. This fusion seeks to achieve unseen levels of efficacy and accuracy in image recognition tasks by utilizing the complementing qualities of two architectures. Using DenseNet201 and ResNet50 with an ECA-Net attention mechanism to predict palm plant diseases is successful because it takes advantage of each architecture's advantages. It is crucial to remember that the success of any model depends on several different things, including the data's quality, how it was processed, how its hyperparameters were tuned, and the methods used to evaluate the model. The main contributions of our study are as follows:

- Develop a significant contribution to the palm leaf disease classification field by leveraging state-of-the-art deep learning architecture.
- This study's methodological approach is methodically constructed around a hybrid model that employs ResNet50, DenseNet201, and ECA-Net architectures, those deliberately harnessed for their distinct and specialized goals.
- Modifications in image pixel, pooling layers, and optimizer choices significantly boost our approach, enhancing efficiency and reducing time complexity, aligning well with the discussed hardware and software specifications in the results section.
- Through meticulous experimentation and data augmentation techniques, we achieve the highest accuracy and demonstrate the effectiveness of combining our proposed models to address the complex challenge of palm leaf disease classification.

The remainder of this paper is organized as follows: section II outlines related work in automated plant disease detection. Section III methodology details the dataset and pre-processing steps used for the experiments. It describes the methodology, including the architecture of the hybrid model and the transfer learning technique. Section IV presents the experimental results and their analysis. Section V discusses this research's implications and future directions. Finally, section VI conclusions and future work of the paper.

## II. RELATED WORKS

Over recent years, the agricultural industry has faced ongoing challenges in identifying and assessing the severity of diseases affecting date palm leaves. One prominent approach in disease classification and plant leaf categorization was introduced by Aakanksha Rastogi et al. utilizing an Artificial Neural Network (ANN) for training [8]. This method, involving rigorous testing and training steps, aimed to reduce mean square error and enhance accuracy. Challenges, however, arose due to the resource-intensive nature of ANN training, demanding high-quality input images [9]. The thesis had a fundamental level. To begin with, A novel technique proposed by KYAMELIA ROY et al. integrated Principal Component Analysis (PCA) DeepNet with Generative Adversarial Networks (GANs) and a customized Deep Neural Network (DNN) [10]. Employing the Faster Region-Based Convolutional Neural Network (F-RCNN) for disease classification, this approach exhibited exceptional outcomes [11]. To identify and classify indications of illnesses affecting palm oil leaves, Masazhar and Kamal [12] devised an automated method using digital image processing and the extreme learning machine (ELM). This method successfully identified two palm oil diseases using k-means clustering and a multiclass Support Vector Machine (SVM) classifier based on leaf symptoms. Thirteen distinct features extracted through k-means clustering aided in disease categorization. The proposed method by Belal A. M. Ashqar et al. developed a Convolutional Neural Network (CNN) model for image-based tomato leaf disease identification, demonstrating improved performance for full-color images compared to grayscale ones [13]. Additionally, Mrs. Shruthi U et al. highlighted the potential of Convolutional Neural Networks in identifying various crop diseases through machine learning algorithms [14]. A hybrid learning model In 2021, Anindita Septiarini et al. proposed a technique focusing on diagnosing diseases in oil palm leaves, utilizing pixel quantification and color attribute extraction [15]. This method involved Otsu thresholding in the Lab color space for Region of Interest (ROI) detection, followed by preprocessing and classification using k-nearest neighbors (KNN) [15]. A comprehensive method for identifying leaf diseases in tomato plants was implemented by Sunil S. Harakannanavar et al. [16]. This method integrated several techniques, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Convolutional Neural Network (CNN), Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and Gray Level Co-occurrence Matrix (GLCM). Their preprocessing workflow included K-means clustering and texture-based feature extraction using GLCM, providing feature vectors for SVM, KNN, and CNN classifiers. Deep learning made significant strides in image recognition systems in 2022 [17]. Using pre-trained models like DenseNet-121, ResNet-50, VGG-16, and Inception V4, researchers effectively identified plant diseases, evaluated through classification accuracy, sensitivity, specificity, and F1 score [18]. Additionally, Al-Mahmood et al. introduced a dataset employing image processing for feature extraction and classification, enabling efficient identification of Palm leaf diseases through Convolutional Neural Network (CNN) utilization [19]. Despite recent technological advances, little research has been conducted on the early identification and classification of date palm disease. Al-Mahmood provided an image dataset that used image processing for feature extraction and classification [19]. The dataset offers valuable data for calculating the number of insects present in a particular region and assessing the severity and scope of the infestation. In our model, we have developed a machine learning system that leverages a new dataset and maintains high standards for image quality. This approach efficiently identifies Palm leaf diseases using a Convolutional Neural Network (CNN). Notably, our model exhibits superior accuracy to other models in this context. Our system employs a hybrid model that seamlessly integrates profound learning principles. We experimented with several models during the training and validation phases, including VGG16, VGG19, ResNet50, ResNet101, DenseNet121, DenseNet169, and DenseNet201. Among these models, the Hybrid model is exceptional, boasting an astonishingly high accuracy rate in precise leaf disease identification. This result underscores our approach's unique capabilities and effectiveness in tackling this critical agricultural challenge.

## III. METHODOLOGY

The methodology section of a palm leaf disease study is an integral part of the research process. It describes the method and strategies used for data collection, data preprocessing, data augmentation, the proposed model, and their architecture study. This section serves as a comprehensive guide for researchers, scientists, and practitioners seeking to advance our knowledge of palm leaf diseases and develop effective strategies for their control.

### A. DATA COLLECTION

The image dataset [19] used in this study was meticulously curated to represent the diverse flora found in various regions within the Aoun district, located in the Karbala Governorate of Iraq. The dataset exclusively focuses on date palm leaves infected with various conditions, categorized into four distinct classes. These categories are based on the health status of the palm leaves and the observed insect growth stage. The classifications are as follows: "healthy," "infected with insects only," "infected with honeydew only," and "infected by a combination of insects and honeydew." It is worth noting that the images of insect-infested leaves encompass a wide range of life cycle stages, from the third generation of nymphs to fully formed adult insects, including those in the fifth nymphal stage. The dataset was gathered using two different types of drone cameras, showcasing the depth and diversity of the image collection process. The acquired dataset consists of around 3,000 images belonging to 4 different classes. The dataset includes images of palm leaf diseases that can be caused by the Dubas bug (Ommatissus lybicus).

### B. DATA SET PREPROCESSING

The actual data collection was precisely scheduled to correspond with the spring and autumn seasons, aligning with the varying life cycles of the problematic insects [19]. Our primary goal was to identify diseased palm specimens accurately using a sophisticated methodology. During the time-consuming image processing, precision cropping painstakingly delineated the virus-infected areas. This laborious post-processing work resulted in the generation of the final dataset images, which all have a pixel resolution of 896 by 896 and are organized in a 3-channel layout.

The stratification of the dataset was deliberately intended to match the seasonal growth dynamics of the insects. This separation produced four different groupings, each of which was meticulously organized into its own designated folder. In autumn, small insects and their eggs embellished the damaged leaves in these photographs. As spring progresses until early summer, the production of honeydew on the leaves becomes more noticeable.

The meticulous curation of the dataset deserves special mention. Images affected by noise, shadows, or dust were rigorously eliminated, ensuring the repository met the highest quality and relevancy criteria. This rigorous curation approach ensured that the dataset is a valuable and reliable resource for future palm leaf disease research and analysis.

### C. DATA AUGMENTATION

A dataset must be enhanced before it can be used for deep learning and machine learning applications. It entails transforming the current data in various ways to produce new training instances. By boosting the diversity and quantity of the training sample, the model becomes more robust and is better equipped to generalize to fresh, unexplored data. Here are some data augmentation techniques with the dataset [19], which contains images of both healthy and afflicted date palm leaves in Table 1 impacted by the Dubas pest:

- Rotation: Rotate the images by angles (such as 90, 180, or 270 degrees) to represent various palm leaf orientations. This can assist the model in learning to identify Dubas pest damage from different perspectives.
- Flip: Flip the images horizontally. Regardless of whether they emerge on the left or right side of the leaf, this can help the model learn to recognize pests and illnesses.
- Zoom: Zoom in and out on the images at random. This can replicate different vantage points from whence the images were taken, such as up close and far away.

**TABLE 1. A comprehensive analysis of original data collection and augmentation data.**

| Class | Original data | Augmented data | Total data |
|---|---|---|---|
| Bug | 600 | 3900 | 4500 |
| Dubas | 800 | 3700 | 4500 |
| Health | 800 | 3700 | 4500 |
| Honey | 800 | 3700 | 4500 |

- Brightness and Contrast Adjustments: Randomly zoom in and out on the images. This can replicate different vantage points from whence the images were taken, such as up close and far away. Change the images' brightness and contrast levels. This can aid the model's ability to adjust to various illumination situations in the field.
- Noise Injection: Add random noise to the images to increase their resistance to noisy environments found in real-world scenarios.
- Color Manipulation: Adjust the images' hue, saturation, and color balance with color manipulation software. This can aid the model's ability to generalize to differences in leaf color more accurately.
- Crop and Resize: Resize and crop the images to create the appearance of various framings and resolutions. This can facilitate the model's handling of images with various degrees of detail.
- Data balancing: Data augmentation can be applied selectively to the ''Dubas bug'' class in cases with fewer samples than other classes to achieve dataset balance.

  The overall dataset size expanded to 18,000 images from the original 3,000 images following augmentation with methods including rotation, flipping, zooming, etc., shown in Table 1. Out of 18000 images, data from 80% are used for training, and 20% are used for validation during data preprocessing. Figure 2 enhances and displays some images.

### D. PROPOSED MODEL

A specific model has been painstakingly created in response to the rising demand for effective and automated palm leaf disease detection systems. Locating and detecting palm leaf diseases is specifically suited for this approach. The creation of the model marks a crucial turning point in efforts to use cutting-edge technology to solve the urgent need for more precise, efficient, and scalable plant disease detection and management tactics. Figure 3 shows a graphic representation that captures the substance and importance of this ground-breaking development.

The given neural network architecture is designed for image processing tasks using 224 $\times$ 224 pixel input images. Efficient Channel Attention Network (ECA-Net) and two transfer learning models, ResNet50 and DenseNet-201,

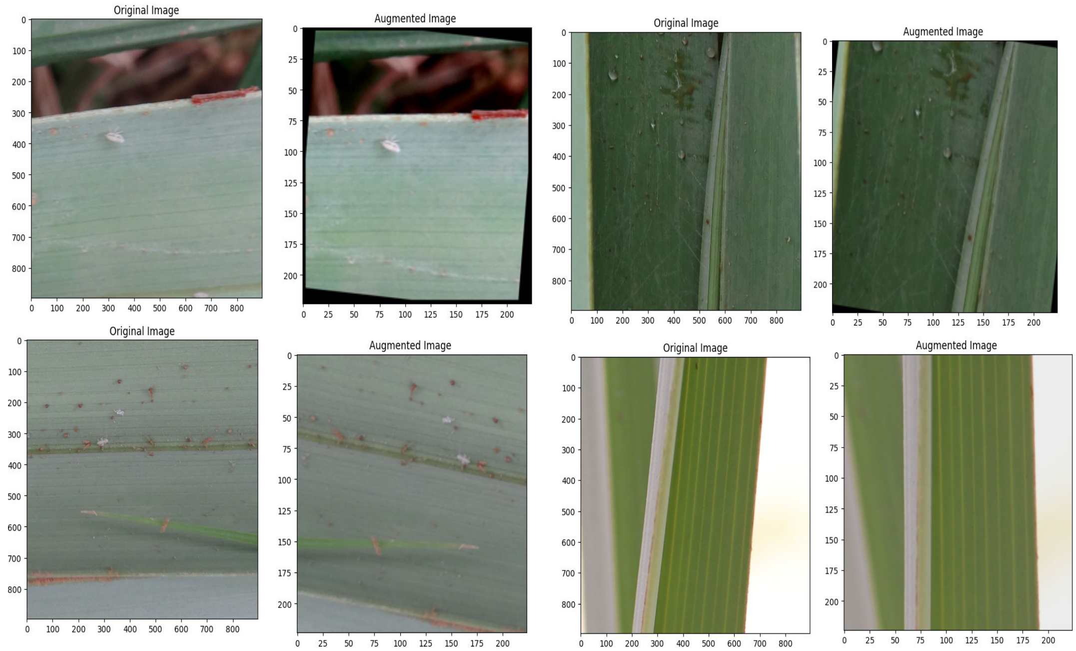



**FIGURE 2.** Augmented visuals of this image were created using various data augmentation techniques.

handle feature extraction and hierarchical representation learning before splitting into two parallel branches. ResNet50 can be modeled as the following function:

$$H_{ResNet50}(x) = F(x) + x_{input} \tag{1}$$

Here in equation 1 $x_{input}$ is the input image and F(x) represents the residual information computed by ResNet50. DenseNet-201 is illustrative of:

$$x_{DenseNet201}(l) = H_l \tag{2}$$

Here, equation 2 $H_l$ encapsulates the features extracted by DenseNet-201. After that, an ECA-Net module further improves the features from ResNet50 by likely using channel-wise attention techniques to boost pertinent information. In the ECA-Net module, channel-wise attention techniques are introduced.

$$F_{Eca-Net}(x) = A(x) \tag{3}$$

Here, equation 3 represents the attention-augmented features. Now ECA-Net module came from ResNet50. So the result will be,

$$E_x = [F(x), H(l)] \tag{4}$$

The results from DenseNet 201 and ECA-Net are then combined by concatenating these enhanced features.

$$X_{concate} = [E(x), A(x)] \tag{5}$$

The spatial dimensions are then reduced via global average pooling to produce a condensed representation of the combined features.

$$X_{pool} = GlobalAvgPool(X_{pool}) \tag{6}$$

The neural network ends with an output layer designed for a particular purpose, such as image classification, and uses the features processed to generate the outcome. By combining attention processes with the benefits of ResNet-50 and DenseNet-201, this architecture may improve performance on challenging visual recognition tasks.

$$X_{output} = Conv(X_{pool}) \tag{7}$$

The neural network architecture is illustrated by this equation-based representation, which effectively combines the advantages of DenseNet-201 and ECA Net to produce the final haze-free images.

### *E. RESNET50*

Deep convolutional networks known as residual networks (ResNets) are designed to primarily utilize shortcut connections to skip entire blocks of convolutional layers [20]. It shines the brightest when traditional deep networks face the enormous obstacle of training complexity. ResNet is a robust solution in these situations, improving classification accuracy and achieving parameter efficiency. We wisely used ResNet50 as a critical part of our research, utilizing it as a potent deep convolutional feature extractor as a testament to its strengths.

Our methodology significantly uses the deployment of ResNet50 for feature extraction, as illustrated in Figure 4. It is important to note that we train the network's initial weights on the Image dataset in recognition that the information acquired in later layers tends to be more specialized and catered to particular classes, as demonstrated by ResNet50's fully connected layer. We investigated the discriminative potential of output vectors from earlier convolutional layers due to

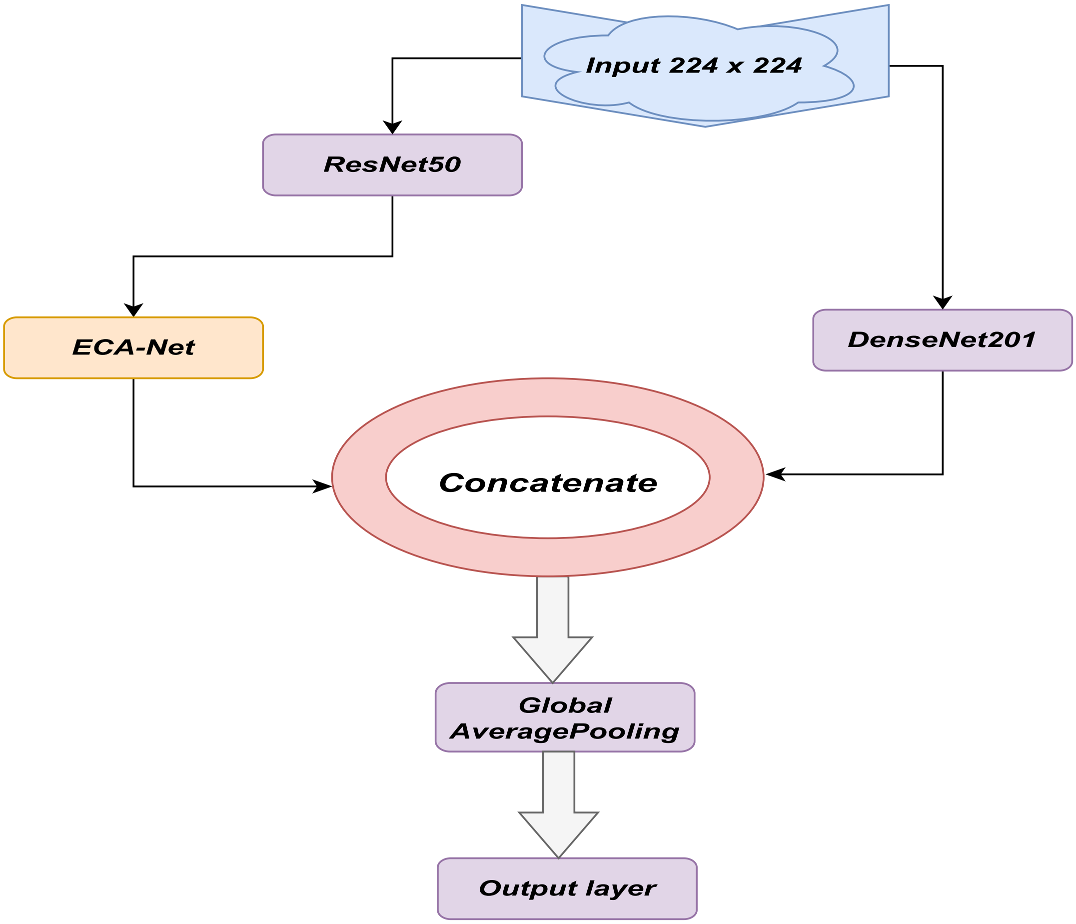



**FIGURE 3.** Within the bounds of this scholarly exposition, the layers of research methodology are unveiled, meticulously elucidating its fundamental principles and intricately interwoven procedures.

our curiosity. Our attention was particularly drawn to the results from the final residual units in convolutional layers 3, 4, and 5. The feature dimensions of these layers vary, with the third layer's features having a more compact dimension than those in the fifth layer. This reinforces the strategic decision we made to use ResNet50 in the context of our investigation. Assuming that the network input is x and the expected mapping is H(x),

$$F(x) = H(x) - x \tag{8}$$

where x represents the higher layer network's characteristic mapping. The observation value of this layer network is H(x), and the residual of this layer network is F(x). The three have a relationship that is

$$H(x) = F(x) + x \tag{9}$$

Although the effects of H(x) and F(x) + x are identical, F(x) is significantly easier to optimize than H(x). Considering that layer L is where the relationships between layers can be represented,

$$x_{L+1} = x_L + F(x_L) \tag{10}$$

$$x_L = x_L + \sum_{i=1}^{L-1} F(X_i) \tag{11}$$

The network model has improved, and the network loss error has decreased, assuming that the network reaches a particular depth. Network degradation could result from an increase in network depth. We can use the residual network to put the network in the ideal condition, whereby the residual F(x) value is 0.

The gradient of the loss function loss concerning $x_k$ in the k-layer network can be represented as

$$\frac{\partial loss}{\partial x_k} = \frac{\partial loss}{\partial x_L} + \frac{\partial loss}{\partial x_L} * \frac{\partial}{\partial x_k} \sum_{i=1}^{L-1} F(x_i) \tag{12}$$

From equation 12, it is clear that it cannot remain constant at -1 throughout the training process and that the network is unaffected by the gradient disappearance issue. The residual module can significantly reduce the network's weight, and the backpropagation is quicker and more versatile.

### F. DENSENET201

A fascinating idea of direct connections between layers is introduced by DenseNet, a cutting-edge neural network

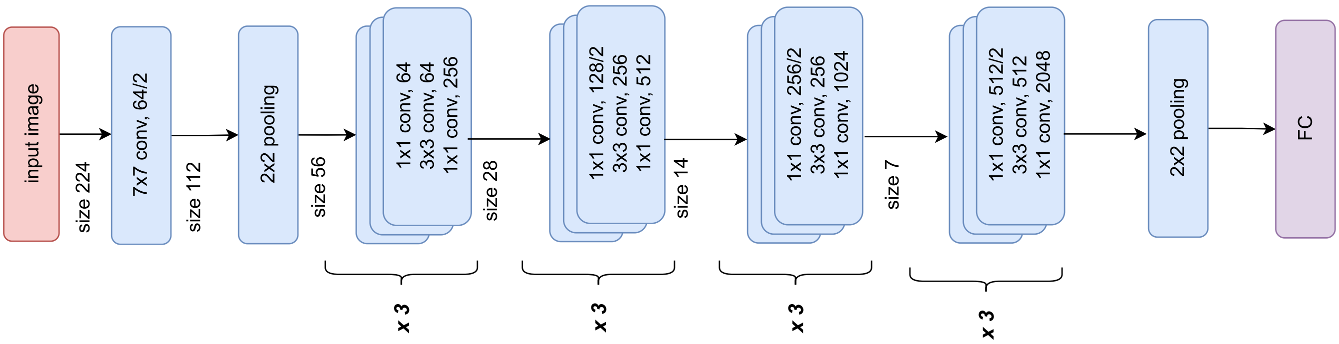



**FIGURE 4.** The ResNet50 architecture is shown in this figure, a deep learning model renowned for its deep residual blocks.

design, creating thorough connectivity where each layer receives feature maps from all preceding layers up to the current depth level. The network's information flow is considerably improved by this interconnectivity. However, allowing this seamless connectedness becomes difficult when the dimensions of feature maps are altered. The architecture includes downsampling techniques targeted at downsizing feature maps to get around this. The network architecture incorporates an intentional design decision to handle the high number of input channels. In particular, the "bottleneck layer"a $1 \times 1$ convolutional layeris deftly introduced before the next $3 \times 3$ convolutional layer. One of the most important functions of this intermediate bottleneck layer is to efficiently reduce the dimensionality of the feature maps while simultaneously increasing computational effectiveness. Here let,

- x0: Single image
- L: Number of layers
- Hl: Non-linear transformation

l = 1,. . . .,L which stands for the layer index
Now,

$$x_l = H_l(x_{l-1}) \tag{13}$$
$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{14}$$
$$x_l = H_l([x_0, x_1, \ldots.., x_{l-1}]) \tag{15}$$

Here, l stands for the layer index and H means the non-linear operator, and $x_l$ represents the features from the l layer.

A sophisticated engineering approach has been developed in the field of neural network design, specifically within the framework of DenseNet (Densely Connected Convolutional Networks), to solve the difficult challenge of effectively managing a profusion of input channels. This clever method introduces a "bottleneck layer," which takes the form of a $1 \times 1$ convolutional layer placed before the $3 \times 3$ convolutional layer within the network. This bottleneck layer's clever design is based on its ability to drastically reduce the number of feature maps or channels, effectively saving computational resources while maintaining vital information flow.

Furthermore, to augment the overall compactness of the model, a concept known as a "transition layer" is ingeniously incorporated. This transition layer reduces the number of feature maps produced within a certain DenseBlock. By using the compression factor $\theta\epsilon$ (0,1], it achieves this. If this factor is set to 1, an equivalent situation emerges, keeping the number of feature maps constant. However, a careful reduction occurs when is smaller than 1, in which the feature maps are reduced to twice their initial count. As a result of these careful architectural decisions, DenseNet's convolutional neural network design is more simplified yet incredibly powerful.

### G. ECA-NET

To overcome this difficulty, a crucial element known as the ECA-Net, which takes inspiration from the pyramid pooling block, is implemented. The main goal of the ECA-Net is to make sure that the final, haze-free images harmoniously incorporate the decoded features derived from various scales. To improve feature alignment after the decoding process in picture restoration activities, details across various scales must be recovered. As a result, the EB is carefully created using the receptive field paradigm and is skilled in extracting information at various scales. Figure 6 shows how the decoded feature maps go downscaling using global average pooling and scaling factors of 4x, 8x, 16x, and 32x to create a flexible four-scale pyramid. The network may reconstruct an image at different scales thanks to this pyramidal structure's range of varied receptive fields. Then, using $1 \times 1$ convolutions, a dimension-reduction step is carried out on each scale. Implementing a $3 \times 3$ convolutional procedure, acting as the final transformation step, completes this complex process and produces the long-desired haze-free images.

$$V = \sum_z avg - pool(F_c) = \sum_{i=0} \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{i=1}^{W} X_z(i, j) \tag{16}$$

In equation 16, avg_pool() stands for the global average pooling. The value of the z-th channel at position (i, j) is denoted by the symbol Xz(i, j). Each feature map's length and width are denoted by H and W, respectively, and the compressed channel-wise vector is denoted by V.

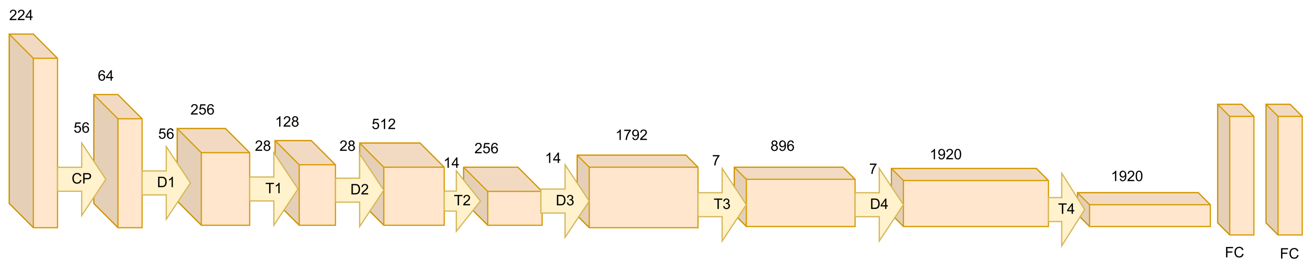



**FIGURE 5.** DenseNet-201 architecture, a powerful deep learning model recognized for its dense connection patterns that enable feature reuse and efficient training.

### H. TRANSFER LEARNING

In machine and deep learning, transfer learning is an effective method that can drastically shorten training periods and boost model performance in various applications. For the classification, we applied the idea of transfer learning. Transfer learning [21] has succeeded with various visual identification tasks, such as detection and image classification.

We experimented with a variety of transfer learning techniques in the ECA-Net design, drawing inspiration from transfer learning methodologies, including VGG16, VGG19, ResNet50, ResNet101, DenseNet121, DenseNet169 and DenseNet201. Through these studies, we discovered that using transfer learning with ResNet-50 and Desnet201 produced positive results. We greatly improved efficiency and segmentation performance by including ECA-Net in our architecture. As a result, we developed a completely new architecture known as ECA-net.

## IV. EXPERIMENT AND RESULT ANALYSIS

A thorough study was conducted to validate the Transfer Learning DeepNet classifier model. The entire training process was executed on a Windows 10 computing environment with an Intel(R) Core(TM) i5CPU, 20GB RAM, and 16GB GPU. Python 3.6.9 version with TensorFlow 2.2.1 was used to implement all offensive image classification models. Anaconda software provides a streamlined environment for managing Python libraries such as TensorFlow, which are commonly used for building image classification models.

This study used pre-processed augmented data containing four distinct classes as the input dataset. During the training phase, various model configurations were iteratively generated to develop an effective classifier model using transfer learning principles. The objective was to leverage pre-existing knowledge and patterns learned from a different domain or task to enhance the model's ability to classify data within the specific target domain, in this case, the four predefined classes.

### A. EXPERIMENT IMAGE PIXEL

These Tables 2, 3, 4 and 5 compare how well a hybrid model performs when using different optimizers (Nadam, RMSprop, SGD, Adamax, and Adam) on various image sizes ($128 \times 128$, $224 \times 224$, $256 \times 256$, and $512 \times 512$). Both training accuracy and validation accuracy are used to measure performance.

In the context of a hybrid image classification model with a fixed image size of $128 \times 128$ pixels Table 2, various optimization algorithms were evaluated based on their performance metrics. With the highest validation accuracy of 85.67% among these optimizers, Adamax stood out and showed its potency in generalizing well to new data. Nadam achieved training accuracy scores of 98.54%, however, Nadam's validation accuracy score was just slightly lower at 84.00%.

When analysis these results, it becomes evident that the Adam optimizer consistently outperforms the other optimizers across all image sizes, yielding the highest validation accuracies (91.02% for $224 \times 224$) Table 3, it is clear from the analysis of these findings. It routinely delivers greater generalization and accuracy performance, making it the best performer among optimizers. Nadam also performs admirably, seconding Adam regarding validation accuracy (87.00%). As picture size increases, RMSProp shows a propensity to overfit, which lowers validation accuracy (82.17%). SGD has the lowest validation accuracy performance (83.83%), which suggests poor learning. Although Adamax operates admirably, its accuracy results fall short of Adam's. These accuracy-based results emphasize the significance of optimizer selection in getting the optimal model performance, with Adam demonstrating to be the most trustworthy option in this situation, independent of image size.

In $256 \times 256$ pixels, Table 4 is based on their training and validation accuracy. The Nadam optimizer demonstrated a very low validation accuracy of 85.50% despite achieving a high training accuracy of 99.33%.

The accuracy of different optimization techniques during training and validation was evaluated in $512 \times 512$ pixels Table 5. Nadam attained a validation accuracy of 86.50% while demonstrating a robust training accuracy of 99.56%.

The $224 \times 224$ image size strikes a good mix between giving the model just enough image detail to pick up useful features and patterns while not overburdening it with computation, as is the case with bigger image sizes. This image size is the best option among the tested image sizes since it enables the model to generalize successfully and attain the maximum validation accuracy.

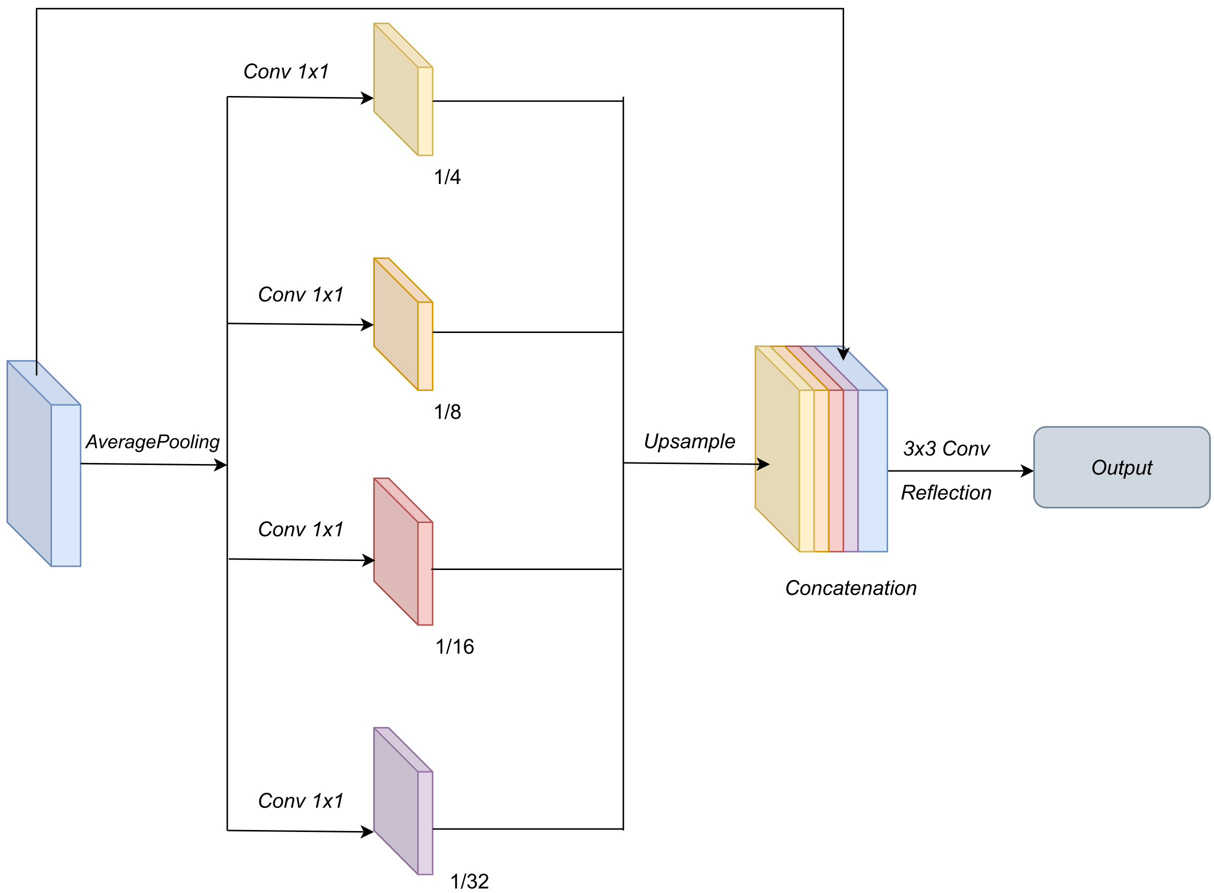



**FIGURE 6.** This figure presents ECA-Net, a leading-edge deep learning framework for advanced computer vision feature learning.

## B. PERFORMANCE ANALYSIS PARAMETERS

We evaluated the effectiveness of our system using four essential metrics: recall, which indicates the capacity to recognize actual positives; accuracy, which counts overall correctness; precision, which quantifies true positives among positive predictions; and the F1-score, which provides a balanced assessment of precision and recall. This thorough assessment offers a complete picture of our system's capabilities and guarantees that it is appropriate for use in practical applications.

### 1) ACCURACY

This numerical measurement, which indicates precision, ensures that our categorizations were correct and precisely measured. It indicates the consistency of our system's actual outcomes across many samples. Precision encapsulates our system's capacity to produce exact and consistent classifications, emphasizing its ability to reliably identify real positive cases with a low percentage of false positives. This measurement dependability is critical when data accuracy and integrity are critical, such as medical diagnosis or quality control in industrial processes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

**TABLE 2.** When image size (128 × 128), comparison results of our hybrid model with different optimizer based on Tran accuracy and valid accuracy.

| Optimizer | Train_accuracy | Valid_accuracy |
|---|---|---|
| **Nadam** | **98.54%** | 84.00% |
| RMSProp | 97.93% | 82.47% |
| SGD | 81.92% | 80.25% |
| Adamax | 98.16% | **85.67%** |
| Adam | 98.46% | 82.17% |

where, $TP$= Ture Positive; $FP$= False Positive; $TN$= True Negative; $FN$= False Negative.

### 2) PRECISION

A precision measure quantifies the system's capacity to produce accurate optimistic predictions concerning the complete set of expected positive observations. It provides valuable insights into the system's effectiveness by calculating the percentage of actual positive instances successfully detected out of all expected positive instances. Practically speaking,

**TABLE 3.** When image size (224 × 224), comparison results of our hybrid model with different optimizer based on Tran accuracy and valid accuracy.

| Optimizer | Train_aaccuracy | Valid_accuracy |
|---|---|---|
| Nadam | 98.08% | 87.00% |
| RMSProp | 99.12% | 82.17% |
| SGD | 83.13% | 83.83% |
| Adamax | 99.29% | 86.00% |
| **Adam** | **99.54%** | **91.02%** |

**TABLE 4.** When image size (256 × 256), comparison results of our hybrid model with different optimizer based on Tran accuracy and valid accuracy.

| Optimizer | Train_accuracy | Valid_aaccuracy |
|---|---|---|
| **Nadam** | **99.33%** | 85.50% |
| RMSProp | 98.37% | **86.17%** |
| SGD | 87.71% | 84.03% |
| Adamax | 97.79% | 84.83% |
| Adam | 99.21% | 85.17% |

this statistic represents the system's ability to avoid producing false optimistic predictions, highlighting its accuracy in providing findings that can be trusted. High precision is essential in applications where the consequences of false positives can be costly or where maintaining the integrity of optimistic predictions is critical, such as in healthcare diagnostics, fraud detection, or any other field where accuracy and reliability are essential.

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

### 3) RECALL

Recall, also known as "sensitivity" or "true positive rate," measures a system's ability to properly detect positive observations inside a class. It is all about minimizing missed positive instances, which is critical in applications where missing such instances might have major effects, such as medical diagnosis or quality control.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{19}$$

### 4) F1-SCORE

The F1-score represents the weighted average of recall and accuracy. So, both false positives and negatives are considered while calculating this score. Although accuracy may not be as intuitively clear, F1-score typically outperforms accuracy, especially if the distribution of students in the class is uneven. The cost of false positives and false negatives should be equal for accuracy to perform optimally. It is

**TABLE 5.** When image size (512 × 512), comparison results of our hybrid model with different optimizer based on Tran accuracy and valid accuracy.

| Optimizer | Train_accuracy | Valid_accuracy |
|---|---|---|
| **Nadam** | **99.56%** | 86.50% |
| RMSProp | 97.33% | 88.17% |
| SGD | 85.89% | 84.43% |
| Adamax | 98.75% | 85.35% |
| Adam | 98.78% | **89.85%** |

desirable to pay attention to both accuracy and recall if the costs of false negatives and false positives are significantly different.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

## C. BASELINE EVALUATION

Using the Python machine learning package, a baseline test was conducted. Several well-known deep learning architectures, such as VGG16, VGG19, ResNet50, ResNet101, DenseNet121, DenseNet169, and DenseNet201, are evaluated in the first step. Despite having somewhat simpler structures than the other designs, VGG16 and VGG19 showed lower training and validation accuracies shown in Table 6, with VGG19 showing a modest improvement over VGG16. This implies that these models may have trouble capturing the complexity of the underlying data or that they may be prone to overfitting.

The ResNet50 architecture, on the other hand, showed a notable improvement in training accuracy, scoring 95.33%, demonstrating its competence in fitting the training data. The fact that there was still a glaring discrepancy between training and validation accuracy suggests possible overfitting.

Regarding training and validation accuracy, the ResNet101 and DenseNet121 architectures fared well, with ResNet101 above the 90% in both measures in Table 6 threshold. These models appear to have achieved a better equilibrium between training data fitting and generalization to validation data.

The usefulness of DenseNet169 and DenseNet201 in identifying intricate patterns in the data was shown by their strong validation accuracies, which exceeded 83%, and good training accuracies.

High training and validation accuracy was demonstrated by the ensemble model, ResNet50+DenseNet201, demonstrating that mixing different architectures can enhance model performance.

However, the suggested hybrid model stood out in this baseline examination, earning a fantastic 91.02% validation accuracy and an excellent 99.54% training accuracy. This shows that not only was the hybrid model able to fit the training data incredibly well, but it could also generalize well to fresh, untried data.

**TABLE 6.** Comparison results of models with different architectures based on Tran accuracy and valid accuracy.

| Model | Train_accuracy | Valid_accuracy |
|---|---|---|
| VGG16 | 72.73% | 65.69% |
| VGG19 | 77.27% | 68.25% |
| ResNet50 | 95.33% | 75.67% |
| ResNet101 | 92.25% | 79.50% |
| DenseNet121 | 92.87% | 76.12% |
| DenseNet169 | 94.13% | 76.00% |
| DenseNet201 | 92.67% | 83.83% |
| ResNet50+DenseNet201 | 97.70% | 86.83% |
| **Proposed hybrid model** | **99.54%** | **91.02%** |

### D. MODEL PERFORMANCE ANALYSIS

In Table 7, a model's precision, recall, and F1 score for several classes are displayed. The model's accuracy is at its maximum (99%) for the healthy class. This demonstrates how adept the model is at spotting instances of healthy honey.

Additionally, the model has the greatest bug class recall (96%). This demonstrates how adept the model is at spotting cases of bug honey. The average accuracy, recall, and F1 score may be used to assess the model's overall performance. The average accuracy, recall, and F1-score are all 0.88 in this illustration. This indicates that the model is operating effectively as a whole. To assess the efficiency and dependability of a hybrid diagnosis system, it is critical to look at the performance metrics for the classification of palm leaves according to Dabus bug infestations. Such a method is probably intended to automatically identify and classify palm leaves according to whether or not they contain Dabus insects.

In Table 8 Among all K-folds, Fold-9 has the highest Tran accuracy (99.48%) and valid accuracy (98.67%), as seen by the image you gave. In other words, the model trained on data from all folds except for Fold-9 performed the best on the data from Fold-9. Hybrid models were manually used, but K-Fold cross-validation improved accuracy, reducing overfitting. Overfitting occurs when a model learns the training data too well and cannot generalize to new data. By training the model on numerous alternative data splits, K-Fold cross-validation aids in mitigating this.

The red and blue lines in Figure 7 show training and validation accuracy, respectively. Validation accuracy assesses how well the model performed using validation data, whereas training accuracy analyzes its performance using the training dataset.

Indicating the model's early lack of accuracy, the red line begins at a low value, about 0.75. The red line increases consistently throughout the epochs, showing how the model's accuracy increases as it gains knowledge from the training set. The red line's peak at epoch 20 suggests that this is the ideal position at which the model isn't overfitting the training set at that time, which is about 0.95. However, the red line's fall after epoch 20 indicates overfitting. In the exact Figure, the blue line shows the validation loss, and the red line shows the training loss. The red line starts at a high value of about 1.0, illustrating the model's poor accuracy. The red line, however, continuously drops as the number of epochs rises, suggesting increased accuracy as the model gains knowledge from the training set. The red line's lowest point, at around epoch 20, indicates the ideal number of epochs for the model to achieve decent generalization without overfitting. The red line begins to increase after epoch 20, indicating the commencement of overfitting as the model gets too closely matched to the training data. Following a similar pattern, the blue line maintains a slightly higher position than the red line, primarily due to disparities between the validation and training data. Unseen by the model, validation data are used to assess how well generalizable the model is. The apparent space between these lines indicates overfitting when the model overly latches onto patterns in training data, hurting its capacity to generalize. Reducing the number of training epochs or using regularization techniques

**TABLE 7.** The performance parameters for the hybrid model, including precision, recall, and F1 score, are shown in the table below.

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| Bug | 0.88 | 0.96 | 0.91 |
| Dubas | 0.83 | 0.80 | 0.81 |
| Healthy | 0.99 | 0.99 | 0.99 |
| Honey | 0.85 | 0.82 | 0.83 |

**TABLE 8.** Our model is based on Tran accuracy and valid accuracy performance using K-Fold cross-validation.

| Fold | Train_accuracy | Valid_accuracy |
|---|---|---|
| Fold 1 | 98.92% | 89.66% |
| Fold 2 | 99.29% | 94.66% |
| Fold 3 | 99.44% | 95.67% |
| Fold 4 | 99.29% | 97.33% |
| Fold 5 | 99.40% | 98.33% |
| Fold 6 | 97.59% | 94.99% |
| Fold 7 | 99.14% | 98.66% |
| Fold 8 | 98.62% | 98.66% |
| **Fold 9** | **99.48%** | **98.67%** |
| Fold 10 | 98.89% | 98.33% |

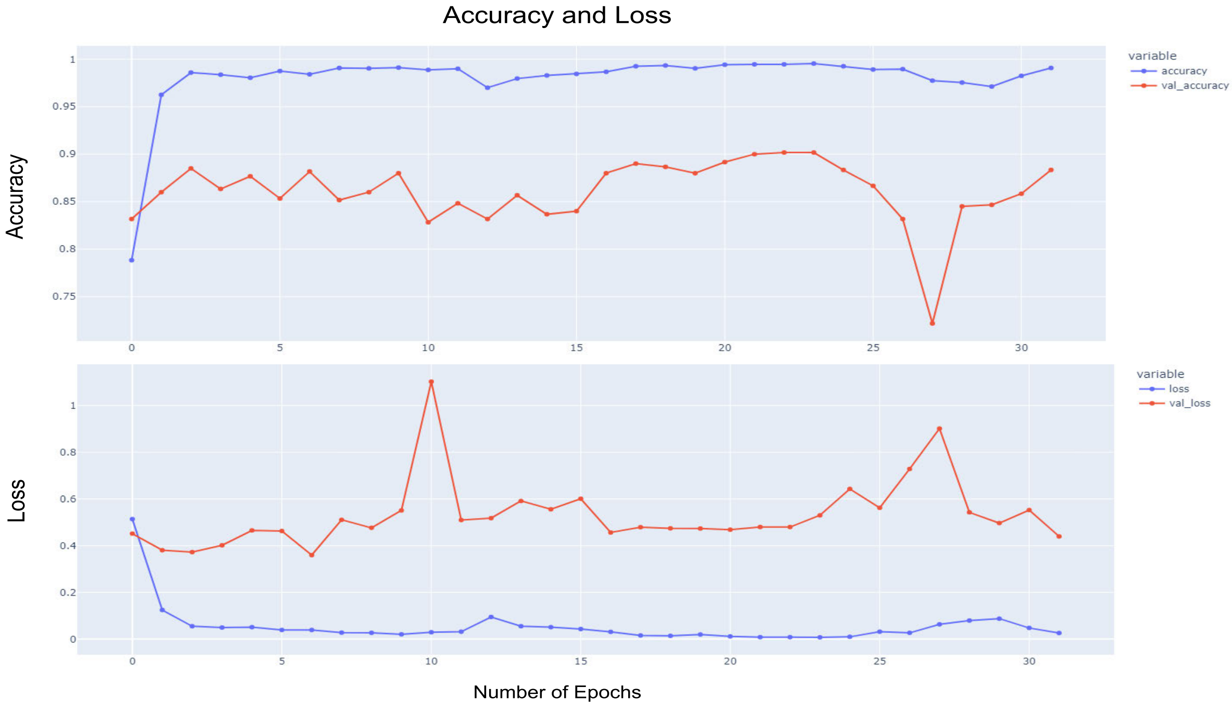



**FIGURE 7.** To evaluate and optimize the performance of hybrid models, plots show how the model's accuracy and loss changes due to training, dataset variation, or other factors.

**TABLE 9.** Comparison results of models with different architectures.

| Ref. | Dataset | Model | Accuracy |
|---|---|---|---|
| [3] | Date Palm White Scale Disease | SVM, KNN, RF and LightGBM | 98.29% |
| [22] | Plant Disease Symptom | AlexNet-based DCNN | 80% |
| [23] | nutrient disease of oil palm leaves | SVM | 95% |
| [24] | Date Palm White Scale Disease | K-Nearest Neighbors classifier | 96.90% |
| [25] | Palm Disease Image Datase | AlexNet CNN and AlexNet-SVM | 96% |
| [26] | Palm Ganoderma Disease | KNN, NaïveBayes | 97% |
| [27] | palm leaf image | machine learning | 87.75% |
| [28] | Date Palm Leaves | CNN,GoogleNet, and AlexNet | 98% |
| **Proposed model** | **Infected date palm leaves by dubas insects** | **Proposed hybrid model** | **99.54%** |

to prevent overfitting during training are two tactics that promote enhanced generalization.

In Figure 8, the image's confusion matrix is a table used to assess how well a classification model is performing. Although the actual class differed, the matrix's cell values show how many occurrences were given a certain classification. The confusion matrix used in this illustration is for a model that divides insects into groups of bugs and dubas. The actual classes are in the left column, while the anticipated classes are in the top row. The matrix's cell values correspond to the number of instances in each actual and anticipated classes, respectively.

## E. COMPARATIVE ANALYSIS

The Table 9 we provided shows the different models for detecting infected date palm leaves by dubas insects. Our suggested hybrid model gets a score of 99.54%, much higher than the competing models. The advantages of our hybrid model are numerous. We use a brand-new data collection

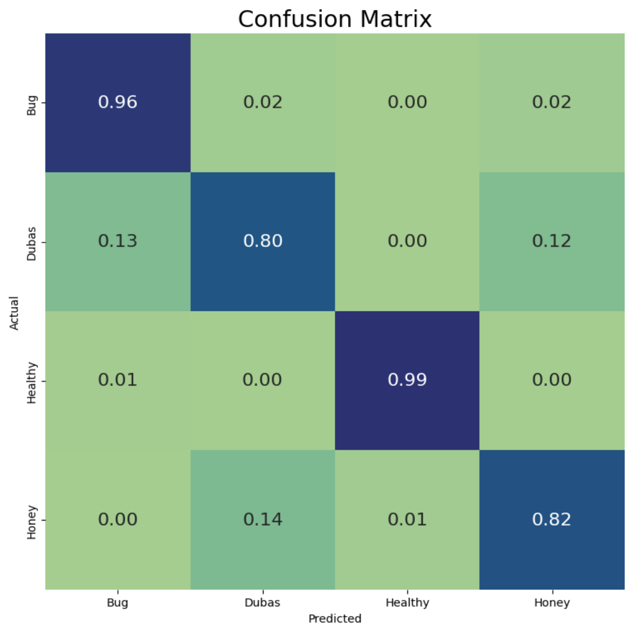



**FIGURE 8. Confusion matrix showing the effectiveness of the hybrid model.**

titled "Infected date palm leaves by dubas insects." This one is more focused than the previous data sets used to identify different kinds of plant illnesses. This indicates that the hybrid model is more adept at identifying the distinctive traits of diseased date palm leaves. The hybrid model uses the advantages of two machine learning techniques: CNN and SVM. SVMs are superior at classification tasks, whereas CNNs are better in feature extraction from pictures [29]. The hybrid model outperforms the separate applications of these methods by combining them. The model also underwent training via transfer learning, using prior information from a similar activity [30]. In this case, the hybrid model was trained on a dataset including images of healthy and sick date palm leaves, giving it a strong comprehension of visual attributes and improving its precision in identifying infected leaves [31]. Consequently, the model developed to identify sick date palm leaves is more precise and effective. Therefore, our model is superior to previous models.

## V. DISCUSSION

Detecting insect infestation on palm leaves is a significant concern that requires a detailed framework for accurate diagnosis at all stages of palm growth. This study established a reliable process for early detection of parasitic infestations, often resulting from bug, dubas, healthy, and honey palm leaves, which can cause extensive damage to palm leaves and other plant parts.

Our methodology collects high-resolution images of bug, dubas, healthy, and honey palm leaves, annotated and divided into training and validation sets. To ensure consistent data, we use augmentation techniques to correct data imbalances and implement preprocessing procedures to improve data quality and model practices, such as downsampling, data normalization, and color transformation. Our study evaluates various deep learning models, including VGG16, VGG19, ResNet-50, ResNet-101, DenseNet121, DenseNet169, and DenseNet201, which accurately diagnose diseases with unprecedented precision. We carefully evaluate model dependability and efficiency using K-Fold cross-validation and class-specific measures to ensure thorough accuracy evaluation.

Our diagnostic procedure using ResNet50, ECA-Net, and DenseNet201 on $224 \times 224$ palm leaf images has proven highly effective, achieving an outstanding 99.54% training accuracy and 91.02% validation accuracy. To address overfitting issues, we introduce the K-fold cross-validation method, which successfully generalizes the model to unknown data in Fold 9, with a notable 99.48% training accuracy and 98.67% validation accuracy.

However, we acknowledge that the hybrid model for diagnosing leaf disease has significant challenges, such as the impact of labeled datasets' quantity and caliber on model performance. Additionally, the mix of methodologies could make it challenging to comprehend models, and the complexity of hybrid models could constrain real-time applications due to longer training durations and higher computational demands.

Despite these constraints, hybrid models for leaf disease detection hold great promise for real-world agricultural contexts. To this end, we recommend more resource-constrained deployment, restricted generalization to new illnesses or plant kinds, ongoing model maintenance, and consideration of privacy-related ethical issues.

## VI. CONCLUSION AND FUTURE WORK

In this study, the problem of classifying objectionable language in the context of palm leaf disease, we presented two independent machine learning assessment methodologies: baseline and ensemble analysis. The identification and categorization of palm leaf diseases have been addressed in this research using a unique technique. We created a hybrid model using a fresh dataset with an excellent accuracy rate of 99.54%. Our findings highlight the superior performance of our suggested model compared to state-of-the-art field projects, highlighting how well it can handle the difficulties associated with detecting palm leaf disease. We see various ways to develop and broaden this study in the future. To make our model even more reliable and capable of recognizing a wider variety of palm leaf disease symptoms, we first intend to improve it by adding new characteristics and investigating more advanced algorithms. Additionally, we aim to investigate the viability of using drone cameras for symptom identification, providing a real-time, airborne viewpoint for monitoring palm farms and spotting infections early on. Additionally, to ensure the applicability and scalability of our strategy, we will continue to develop our research in conjunction with agriculture and machine learning specialists. Our efforts can help reduce the adverse effects of palm leaf diseases on crop output and food security while promoting sustainable agricultural development. This study offers a cutting-edge method for detecting palm leaf disease and creates a pathway for innovative future developments in disease control and precision agriculture. We look forward to further refining our model, exploring new technologies, and making a meaningful impact in the agricultural sector.

## ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University for funding this research through Researchers Supporting Project Number (RSPD2024R1027), King Saud University, Riyadh, Saudi Arabia.

## REFERENCES

[1] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, "Machine learning applications for precision agriculture: A comprehensive review," *IEEE Access*, vol. 9, pp. 4843–4873, 2021.

[2] Y. Hu, G. Liu, Z. Chen, J. Liu, and J. Guo, "Lightweight one-stage maize leaf disease detection model with knowledge distillation," *Agriculture*, vol. 13, no. 9, p. 1664, Aug. 2023.

[3] A. Hessane, A. El Youssefi, Y. Farhaoui, B. Aghoutane, and F. Amounas, "A machine learning based framework for a stage-wise classification of date palm white scale disease," *Big Data Mining Anal.*, vol. 6, no. 3, pp. 263–272, Sep. 2023.

[4] T. Ticktin, "The ecological implications of harvesting non-timber forest products," *J. Appl. Ecol.*, vol. 41, no. 1, pp. 11–21, 2004.

[5] T. A. Kumar, R. Rajmohan, S. A. Ajagbe, T. Gaber, X.-J. Zeng, and F. Masmoudi, "A novel CNN gap layer for growth prediction of palm tree plantlings," *PLoS ONE*, vol. 18, no. 8, Aug. 2023, Art. no. e0289963.

[6] J. Kong, H. Wang, C. Yang, X. Jin, M. Zuo, and X. Zhang, "A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition," *Agriculture*, vol. 12, no. 4, p. 500, Mar. 2022.

[7] X. Li, M. Li, P. Yan, G. Li, Y. Jiang, H. Luo, and S. Yin, "Deep learning attention mechanism in medical image analysis: Basics and beyonds," *Int. J. Netw. Dyn. Intell.*, vol. 2023, pp. 93–116, Mar. 2023.

[8] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic," in *Proc. 2nd Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2015, pp. 500–505.

[9] A. A. Elngar, M. Arafa, A. Fathy, B. Moustafa, O. Mahmoud, M. Shaban, and N. Fawzy, "Image classification based on CNN: A survey," *J. Cybersecur. Inf. Manage.*, vol. 6, no. 1, pp. 18–50, 2021.

[10] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–11, Jan. 2016.

[11] K. Roy, S. S. Chaudhuri, J. Frnda, S. Bandopadhyay, I. J. Ray, S. Banerjee, and J. Nedoma, "Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet," *IEEE Access*, vol. 11, pp. 14983–15001, 2023.

[12] A. N. I. Masazhar and M. M. Kamal, "Digital image processing technique for palm oil leaf disease detection using multiclass SVM classifier," in *Proc. IEEE 4th Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Nov. 2017, pp. 1–6.

[13] B. A. M. Ashqar and S. S. Abu-Naser, "Image-based tomato leaves diseases detection using deep learning," *Int. J. Acad. Eng. Res.*, vol. 2, no. 12, pp. 10–16, Dec. 2018.

[14] U. Shruthi, V. Nagaveni, and B. K. Raghavendra, "A review on machine learning classification techniques for plant disease detection," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 281–284.

[15] A. Septiarini, H. Hamdani, T. Hardianti, E. Winarno, S. Suyanto, and E. Irwansyah, "Pixel quantification and color feature extraction on leaf images for oil palm disease identification," in *Proc. 7th Int. Conf. Electr., Electron. Inf. Eng. (ICEEIE)*, Oct. 2021, pp. 1–5.

[16] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transition Proc.*, vol. 3, no. 1, pp. 305–310, Jun. 2022.

[17] J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, "Deep learning-based leaf disease detection in crops using images for agricultural applications," *Agronomy*, vol. 12, no. 10, p. 2395, Oct. 2022.

[18] J. J. Mondal, M. F. Islam, S. Zabeen, A. B. M. A. A. Islam, and J. Noor, "Note: Plant leaf disease network (PLeaD-Net): Identifying plant leaf diseases through leveraging limited-resource deep convolutional neural network," in *Proc. ACM SIGCAS/SIGCHI Conf. Comput. Sustain. Societies (COMPASS)*, Jun. 2022, pp. 668–673.

[19] A. M. Al-Mahmood, H. I. Shahadi, and A. R. H. Khayeat, "Image dataset of infected date palm leaves by dubas insects," *Data Brief*, vol. 49, Aug. 2023, Art. no. 109371.

[20] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. de Geus, "Malicious software classification using transfer learning of ResNet-50 deep neural network," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1011–1014.

[21] A. Sagar and J. Dheeba, "On using transfer learning for plant disease detection," *BioRxiv*, vol. 2020, pp. 1–13, May 2020.

[22] M. Nababan, Y. Laia, D. Sitanggang, O. Sihombing, E. Indra, S. Siregar, W. Purba, and R. Mancur, "The diagnose of oil palm disease using naive Bayes method based on expert system technology," *J. Phys., Conf. Ser.*, vol. 1007, Apr. 2018, Art. no. 012015.

[23] H. M. Asraf, M. T. Nooritawati, and M. S. B. S. Rizam, "A comparative study in kernel-based support vector machine of oil palm leaves nutrient disease," *Proc. Eng.*, vol. 41, pp. 1353–1359, Jan. 2012.

[24] A. Hessane, A. E. Youssefi, Y. Farhaoui, and B. Aghoutane, "Toward a stage-wise classification of date palm white scale disease using features extraction and machine learning techniques," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, May 2022, pp. 1–6.

[25] J. H. Ong, P. Ong, and K. L. Woon, "Image-based oil palm leaves disease detection using convolutional neural network," *J. Inf. Commun. Technol.*, vol. 21, pp. 383–410, Jul. 2022.

[26] S. Liaghat, R. Ehsani, S. Mansor, H. Z. M. Shafri, S. Meon, S. Sankaran, and S. H. M. N. Azam, "Early detection of basal stem rot disease (Ganoderma) in oil palms based on hyperspectral reflectance data using pattern recognition algorithms," *Int. J. Remote Sens.*, vol. 35, no. 10, pp. 3427–3439, May 2014.

[27] A. F. Aji, Q. Munajat, A. P. Pratama, H. Kalamullah, J. Setiyawan, and A. M. Arymurthy, "Detection of palm oil leaf disease with image processing and neural network classification on mobile device," *Int. J. Comput. Theory Eng.*, vol. 5, no. 3, pp. 528–532, 2013.

[28] S. Alshehhi, S. Almannaee, and M. Shatnawi, "Date palm leaves discoloration detection system using deep transfer learning," in *Proc. Int. Conf. Emerg. Technol. Intell. Syst* Cham, Switzerland: Springer, 2022, pp. 150–161.

[29] X. Sun, J. Park, K. Kang, and J. Hur, "Novel hybrid CNN-SVM model for recognition of functional magnetic resonance images," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1001–1006.

[30] K. Aderghal, K. Afdel, J. Benois-Pineau, and G. Catheline, "Improving Alzheimer's stage categorization with convolutional neural network using transfer learning and different magnetic resonance imaging modalities," *Heliyon*, vol. 6, no. 12, Dec. 2020, Art. no. e05652.

[31] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocessors Microsyst.*, vol. 80, Feb. 2021, Art. no. 103615.

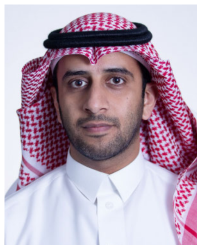
**S. M. NURUZZAMAN NOBEL** is currently pursuing the Graduate degree in computer science and engineering with the Bangladesh University of Business and Technology (BUBT), Bangladesh. He has a major in artificial intelligence (AI). He has published some articles using ML and DL. His current research interests include artificial intelligence, machine learning, deep learning, computer vision, object detection, NLP, and federated learning.

**MD. ASIF IMRAN** received the Graduate degree in computer science and engineering from the Bangladesh University of Business and Technology (BUBT). He is currently an Artificial Intelligence Specialist. He is quite good at machine learning and deep learning and has a solid academic background. He investigated the complexities of artificial intelligence during his time with BUBT, concentrating on deep learning and machine learning techniques. He is a unique student in the area because of his commitment to and enthusiasm for these technologies. He is still making important advances in computer science today, especially in the field of artificial intelligence development. His work involves putting forth novel ideas and methods to improve sentiment analysis and recommender system efficacy and accuracy.

**NAHIDA ZAMAN BINA** is currently pursuing the bachelor's degree in computer science and engineering with the Bangladesh University of Business and Technology (BUBT), Bangladesh. She has a major in artificial intelligence (AI). She has recently conducted research on artificial intelligence, machine learning, and deep learning.

**MD. MOHSIN KABIR** (Member, IEEE) received the Bachelor of Science degree in CSE from the Bangladesh University of Business and Technology (BUBT), Bangladesh, in 2021. He is currently pursuing the joint master's degree in intelligent field robotics systems (IFRoS) with the University of Girona, Spain, and Eötvös Loránd University, Hungary, funded by Erasmus Mundus Scholarship (2022–2024). In addition to his studies, he holds a position as a Lecturer with the Department of Computer Science and Engineering, BUBT (study-leave), where he was a Research Assistant and a Researcher with the Advanced Machine Learning Laboratory. Also, he has had the privilege of collaborating with several prominent research laboratories around the globe, including the Computer Vision & Pattern Recognition Laboratory, University of Asia Pacific, Bangladesh, and the Database System Laboratory, University of Aizu, Japan. With an extensive research background, he has authored over ten articles in high-impact journals, such as IEEE Access, *Sensors*, *Computer Systems Science and Engineering*, *Biology*, and *Mathematics*. In addition, he has contributed to the scientific community by publishing over ten conference papers and actively participating in well-established conferences, including IEEE HONET, ICCIT, ICIEV, icIVPR, DASA, BIM, and ICSCT. Moreover, some of his research work has been published as a chapter in a few esteemed books related to machine learning and AI. His research interests include artificial intelligence, machine learning, deep learning, computer vision, the IoT, and robotics.

**MEJDL SAFRAN** received the M.Sc. and Ph.D. degrees in computer science from Southern Illinois University Carbondale, in 2013 and 2018, respectively. He is currently an Assistant Professor in computer science with King Saud University. He has been a Faculty Member, since 2007. His research interests include computational intelligence, artificial intelligence, deep learning, pattern recognition, and predictive analytics. Since 2018, he has been providing part-time consulting services in the field of artificial intelligence to private and public organizations and firms. He has published articles in refereed journals and conference proceedings, such as *ACM Transactions on Information Systems*, *Applied Computing and Informatics*, *Biomedicine* (MDPI), *Sensors* (MDPI), IEEE International Conference on Cluster, IEEE International Conference on Computer and Information Science, International Conference on Database Systems for Advanced Applications, and International Conference on Computational Science and Computational Intelligence. His current research interests include developing efficient recommendation algorithms for large-scale systems, predictive models for online human activities, machine learning algorithms for performance management, and modeling and analyzing user behavior.



**SULTAN ALFARHOOD** received the Ph.D. degree in computer science from the University of Arkansas. He is currently an Assistant Professor with the Department of Computer Science, King Saud University (KSU). Since joining KSU, in 2007, he has made several contributions to the field of computer science through his research and publications. His research interests include machine learning, recommender systems, linked open data, text mining, and the ML-based IoT systems. His work includes proposing innovative approaches and techniques to enhance the accuracy and effectiveness of these systems. His recent publications have focused on using deep learning and machine learning techniques to address challenges in these domains. Sultans research continues to make significant contributions to the field of computer science and machine learning. His work has been published in several high-impact journals and conferences.

**M. F. MRIDHA** (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He is currently working as an Associate Professor with the Department of Computer Science, American International University-Bangladesh (AIUB). He is the Founder and Director of Advanced Machine Intelligence Research (AMIR) Laboratory. He also worked as an Associate Professor and the Chairman with the Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT) form 2019 to 2022, and as a CSE Department Faculty Member at the University of Asia Pacific and as a graduate head from 2012 to 2019. His research experience, within both academia and industry, results in over 160 journal and conference publications. His research work contributed to the reputed *Journal of Scientific Reports Nature, Knowledge-Based Systems, Artificial Intelligence Review*, Engineering Applications of Artificial Intelligence IEEE Access, Sensors, Cancers, Biology and Applied Sciences etc. His research interests include artificial intelligence (AI), machine learning, deep learning, and natural language processing (NLP). He is a Professional Member of ACM. He served as an Editorial Board Member of several journals including *PLOS ONE Journal*. He has served as a reviewer of reputed journals like IEEE Transactions on Neural Networks, Artificial Intelligence Review, IEEE Access, Knowledge base System, Expert System, Bioinformatics, *Springer Nature*, MDPI etc.