IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2024, accepted 14 May 2024, date of publication 30 May 2024, date of current version 20 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3407153

RESEARCH ARTICLE

Mulberry Leaf Disease Detection Using CNN-Based Smart Android Application

ABDUS SALAM[1,2], MANSURA NAZNINE[3], NUSRAT JAHAN[1], EMAMA NAHID[1], MD NAHIDUZZAMAN[1,2], AND MUHAMMAD E. H. CHOWDHURY[2], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh
[2]Department of Electrical Engineering, Qatar University, Doha, Qatar
[3]Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh

Corresponding author: Muhammad E. H. Chowdhury (mchowdhury@qu.edu.qa)

This work was supported by Qatar National Library (QNL).

ABSTRACT Mulberry leaves serve as the primary food source for Bombyx mori silkworms, crucial for silk thread production. However, mulberry trees are highly susceptible to diseases, spreading rapidly and causing significant losses. Manual disease identification across large farms is arduous and time-consuming. Leveraging computer vision for early disease detection and classification can mitigate up to 90% of production losses. This study collected leaves from two regions of Bangladesh, categorized as healthy, leaf rust-affected, and leaf spot-affected. With a total of 1091 images, split into training (764), testing (218), and validation (109) sets for 5-fold cross-validation, preprocessing and augmentation yielded 6,000 images, including synthetics. This study compares ResNet50, VGG19, and MobileNetV3Small on a specific task following architecture modifications. Four convolutional layers with different output channels (512, 128, 64, and 32) were added to baseline models. We assessed how these architectural changes affected model correctness, computing efficiency, and convergence rates. Comparing three pretrained convolutional neural networks (CNNs) - MobileNetV3Small, ResNet50, and VGG19 - augmented with four additional layers, the modified MobileNetV3Small excelled in precision, recall, F1-score, and accuracy, achieving notable results of 97.0%, 96.4%, 96.4%, and 96.4%, respectively, across cross-validation folds. An efficient smartphone application employing the proposed model for mulberry leaf disease recognition was developed. Overall, the model outperformed existing State of the Art (SOTA) approaches, showcasing its effectiveness in disease identification. The interpretative Grad-CAM visualization images match sericulture specialists' assessments, validating the model's predictions. These results imply that, this eXplainable AI (XAI) approach with a modified deep learning architecture can appropriately classify mulberry leaves.

INDEX TERMS Mulberry leaf disease, plant disease detection, transfer learning, modified MobileNetV3Small, AI-enabled mobile application.

I. INTRODUCTION

The mulberry tree (Morus alba) is a deciduous tree local to northern China that has been cultivated for centuries across Asia, Europe, and North America [1]. Various elements of the mulberry plant were utilized in historic times for food, textiles, and medication. Today, the mulberry tree remains an economically and culturally essential species because of its many uses. Mulberry fruits, known as mulberries, are suitable for eating and can be fed on clean or processed into products like juices, jams, wines, and teas [2]. The fruit is nutritious, containing vitamins, minerals, and bioactive phenolic compounds with antioxidant residences [3]. Mulberry leaves are fit for human consumption and farm animals. These are also rich in protein, calories, and minerals. In the sericulture industry, the leaves are the sole food supply for silkworms used in silk manufacturing. Beyond meal uses, extracts from mulberry leaves and culmination have proven

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

medicinal fee. Mulberry extracts exhibit antibacterial, antidiabetic, anti-inflammatory, and antitumor activities, according to pharmacological research. All this helps the traditional use of mulberry flora in natural medication throughout diverse cultures. Cotton, also referred to as ''White Gold'' and the ''King of Fibers,'' is the principal raw material for the booming textile industry and has superior rank to other cash crops. It helps about 60 million individuals make a living. It is a vital agricultural commodity that also generates income for millions of farmers [4]. The mulberry tree, especially the white mulberry (Morus alba), is vital for silk production as it is the primary food source for silkworms (Bombyx mori) [5].

The leaves of mulberry trees contain essential nutrients and bioactive compounds to support the growth and development of silkworm larvae. The S-36 and BR-2 varieties of mulberry leaves were found to have the highest nutritional composition among the other types. Consistently high quantities of protein, chlorophyll, and carotenoids make it suitable for silkworm growth. Silkworms fed on a diet of mulberry leaves produce higher-quality silk compared to other plant sources [6]. According to comparative proteomics investigations, mulberry leaves protect silkworms from feeding harm in addition to providing them with vital amino acids. Despite being heavily consumed by silkworms, mulberry leaves continue to flourish, offering fresh perspectives on the relationships between hosts, plants, and insects. Mulberry leaves are rich in carbohydrates, proteins, vitamins, and minerals, which are converted into silk fibers during the metamorphosis stage of silkworms. The quality and quantity of silk are highly dependent on the abundant supply of mulberry leaves [7].

Fungal diseases such as leaf spot, powdery mildew and rust can affect the nutritional value of mulberry leaves. Studies have shown a reduction in protein, amino acids, chlorophyll, and carotenoids in diseased leaves. Bacterial diseases such as viruses decrease the protein content along with sugar and ascorbic acid in mulberry leaves [8]. Since there was no appropriate mulberry picture dataset available online, the mulberry images utilized in this research were collected from a mulberry garden in Rajshahi, Bangladesh. It was hand-annotated by a sericulture specialist into three classes: ''healthy,'' ''leaf spot,'' and ''leaf rust''. Due to the seasonal nature of mulberry disease, only two diseases were taken into consideration in this study, both of which are prevalent in the winter season in Bangladesh. The work of this study includes:

1) To prevent overfitting and improve the performance of the Deep Learning model (DL), synthetic images were generated through standard data augmentation techniques.
2) A Transfer Learning (TL) based CNN model was used to create a lightweight model for implementation as a mobile application.
3) The interpretability of the mulberry disease classification model was demonstrated using Gradient-weighted Class Activation Mapping (Grad-CAM), an eXplainable AI (XAI) technique. This ensured that the model correctly focused on the relevant areas of interest in the leaf images.

Section II delineates recent research in this domain. The proposed methodology is expounded upon in Section III. Section IV presents the results obtained from the ML model, which are compared with findings from other recent studies. Discussions and prospects for future work are outlined in Section V. Lastly, Section VI encapsulates the primary conclusions

II. LITERATURE REVIEW

Researchers have been leading the way in developing advanced techniques for automatically diagnosing and categorizing plant diseases in the field of plant pathology [9], [10], [11]. Recent studies have shown the feasibility of using CNN to accurately detect plant diseases from leaf images [12], [13], [14] but very few studies have investigated the detection of mulberry leaf diseases. The development of machine learning models to detect major mulberry leaf diseases will be invaluable for agriculture. Therefore, many approaches have been developed to identify, categorize, and examine features connected to different plant diseases.

Zhou et al. used a restructured residual dense network (RRDN) to diagnose 8 different tomato leaf diseases [15]. By using convolution in Res-Dense Block (RDB), they reduced the number of computations. A total of 13,185 images within 9 classes were used from AI Challenger dataset. The dataset was split into 60% for training, 20% for validation, and 20% for testing. After 200 training epochs, the approach achieved a 95% accuracy rate on both the validation and test sets. Similarly, Wu et al. applied a novel data augmentation method called deep convolutional generative adversarial networks (DCGAN) on tomato leaf images collected the Plant Village dataset [16]. The network architecture enhances GAN quality by using strided and fractional-strided convolutions instead of pooling layers. Four pre-trained models—AlexNet, GoogLeNet, ResNet, and VGG16—were used for classification. By adjusting parameters, the LeNet model of Google achieved 94.33% accuracy. Chen et al. also used 18,835 images from the same dataset to indentify tomato leaf diseases using a lightweight CNN (LBFNet) [17]. They enhanced the original dataset with data augmentation and created LBFtomato dataset. Only 0.21 seconds of testing time were required to achieve 99.06% accuracy.

Gómez-Flores et al. used TL to identify Huanglongbing (HLB) in orange leaves [18]. The proposed approach consists of three steps: Image acquisition, Region of Interest (ROI) segmentation, and CNN-based leaf classification. Leaf samples were obtained from orange trees in Mexico to create the dataset which contains 953 images from 12 different classes. Six pre-trained CNN architectures were adjusted for leaf classification while an automatic ROI segmentation method was created. The study discovered that VGG19, due to its

TABLE 1. Summary of the state-of-art models.

Reference	Dataset	Best Model	Testing Accuracy (%)	Best Model's Parameters (Million)	XAI
[16]	PlantVillage (Tomato Leaves)	DCGAN with GoogLeNet	94.33	-	No
[17]	PlantVillage and LBFtomato (Tomato Leaves)	LBFNet	99.0	0.897	No
[18]	Custom (Orange Leaves)	VGG19	99	144	No
[19]	Citrus Dataset and Plant Village (Citrus Fruits and Leaves)	Deep Learning Based CNN	94.55	-	No
[20]	Corn Disease and Severity (CD&S) Database (Maize Leaves)	MaizeNet	97.89	23.91	No
[21]	challenger.ai (Corn Leaves)	K-means with an Improved Deep Learning	93	16.8	No
[22]	PlantVillage, AI challenge, Google websites (Maize Leaves)	DMS-Robust Alexnet	98.62	-	No
[23]	PlantVillage (Apple Leaves)	Deep CNN	98	0.12	No
[24]	FGVC7 and Baidu AI Studio (Apple Leaves)	Improved ResNet	94.23	25.12	No
[25]	Custom (Paddy Leaves)	ResNet-50	92.61	-	No
[26]	Custom (Rice Leaves)	FCM-KM + Faster R-CNN	97.2	-	No
[27]	Custom (Chinese Rose Leaves)	MixNet-CA	98.82	2.37	No
Proposed	**Custom (Mulberry Leaves)**	**Modified MobilenetV3Small**	**96.4**	**3.9**	**Yes (Grad-CAM)**

greater depth and number of characteristics, had the highest accuracy in identifying 12 separate classes, with an overall accuracy of 99%. Khattak et al., on the other hand suggested an integrated system for citrus to detect and classify fruit and leaf diseases that uses CNN [19]. For evaluation, 2293 images from the citrus and Plant Village datasets were utilized. The proposed approach had two CNN layers to extract features and a max-pooling layer to reduce feature map dimensions. For further analysis, the feature matrix was transformed into a feature vector by adding a flattening layer and a classification layer was added to sort images into different groups based on the types of diseases. 20% of the images were used to calculate the model's performance and it achieved the accuracy of 94.65%.

Masood et al. described a DL approach to correctly localize and classify several leaf diseases in maize crops [20]. The Faster Region-based CNN (R-CNN) framework was used in this study. The proposed framework utilizes the ResNet-50 model as the fundamental network for computing deep key points, incorporating spatial-channel attention. The dataset was obtained from the Corn Disease and Severity (CD&S) database, which contains 2112 sample images. With an accuracy rate of 97.89%, MaizeNet model quickly and accurately classifies leaf diseases in maize crops in 0.26 seconds. Additionally, Yu et al. [28] suggested an approach based on K-means clustering and an enhanced DL model for correctly diagnosing the three common diseases of corn leaves. They applied an improved CNN model based on the VGG-19 architecture. The dataset was obtained from the 2018 AI Challenger Competition (challenger.ai) which consists of 900 images. Using a 5-fold cross-validation procedure, the possibility of model overfitting was reduced. An average accuracy of 93% was attained. FLOPs (floating-point operations), which represent the model's time complexity, were used to measure the computational complexity of this model. The number of operations was greatly reduced by reducing the network layout and settings. Amin et al. presented an end-to-end DL model using EfficientNetB0 and DenseNet121 to identify unhealthy corn plant leaves [21]. A subset of the PlantVillage [29] dataset, with 217,000 images, was used in this study. To avoid overfitting, various data augmentation techniques were employed. The model demonstrated an excellent accuracy rate of 98.56% after training on all images for a duration of 6183 seconds. Lv et al. presented a technique for identifying diseases of maize leaves [22]. They suggested a DMS-Robust Alexnet neural network. 12,227 maize leaf images from multiple sources (Plant Village, the AI challenge dataset, and Google websites) were used in this research. The images covered diverse time periods and seven disease classes. The proposed model achieved an accuracy of 98.62%.

Vishnoi et al. detected three distinct apple leaf diseases from images using a deep CNN (Conv-3 DCNN) model with fewer layers [23]. The model included two fully con-

nected layers, three convolutional layers, and employed Rectified Linear Unit (ReLU) activation. Softmax was used for multiclass classification. A dropout layer was added after the third max-pooling unit that randomly deactivated neurons to reduce overfitting. From the Plant Village dataset, 3171 images were collected. Augmentation techniques were utilized to expand the training dataset. The dataset was split with a ratio of 70:10:20 into three subsets for training, validation, and testing. After 8 hours of training and 11 seconds of testing, 98% classification accuracy was achieved. Luo et al., on the other hand, used a multi-scale feature fusion ResNet model on data from FGVC7 [30] and Baidu AI Studio to find different kinds of apple leaf diseases. First, the authors improved the information flow inside a ResNet architecture by applying batch normalization and ReLU. Then, to reduce information loss during ResNet down sampling, they separated channel and spatial projection. Lastly, the 3×3 convolution in ResBlocks was substituted. On the raw dataset, which had 3120 apple leaf images, the optimized model obtained an accuracy of 94.23%, while on the preprocessed dataset, it obtained 94.99%.

Wang et al. [31] suggested a mobile compatible ADSNN-BO model for identifying rice disease which achieved 94.65% testing accuracy. MobileNet served as the foundation for the architecture of the model, which also included layers for attention augmentation and Bayesian Optimization. The dataset had 2370 images of rice leaf diseases. The performance was assessed using 5-fold cross-validation of five different DL models, including VGG16, DenseNet121, MobileNetV1, ResNet50, Inception, and InceptionV3 [24]. Then, a custom AlexNet based deep CNN model was generated. The overall accuracy of the experiments was 97.62%. Moreover, the model reduced the number of parameters by 51,206,928 and the training procedure took only 34.72 minutes. A dataset consisting of 3355 images of paddy plant leaves was collected by Malvade et al. [25] which were divided into four categories. To reduce the possibility of overfitting, they used a variety of popular image augmentation techniques. They also carried out a comparative evaluation of five TL models and the ResNet-50 model showed effective accuracy of 92.61%, which was obtained in just 1626 seconds. Zhou et al. proposed a novel technique for the rapid detection of rice diseases [26] which included a noise reduction filter, a background interference reduction algorithm, and a dynamic clustering optimization strategy (FCM-KM). These were incorporated with the Faster R-CNN algorithm and this approach was tested on 3010 images, which demonstrated a high degree of accuracy. 96.71% accuracy for rice blast, 97.53% for bacterial blight, and 98.26% for blight detection were achieved, while the training time was 69.1 minutes, 78.7 minutes, and 67.3 minutes, respectively.

Liu et al. proposed a lightweight CNN incorporating inception and depth-wise separable convolution with Coordinate Attention (CA) to identify Chinese rose diseases from leaf images [27]. The dataset comprised 1972 training images, 677 validation images, and 677 testing images, including both healthy and diseased leaves. The model achieved an accuracy of 98.82%.

Table 1 shows the summary of all state of art models. This literature review indicates that most studies in this field relied primarily on publicly accessible datasets or occasionally collected real-time data related to plants. Most of these studies used DL and TL models, which are characterized by many parameters, like 144 million for VGG19, 51.21 million for Deep CNN, 25.12 million for enhanced ResNet, and 23.91 million for MaizeNet. Despite being efficient, these models required a lot of Graphics Processing Units (GPUs) training time. Moreover, some researchers trained the model for an extended period of time, such as 8 hours [23] and 1 hour and 18 minutes [32]. Applying these algorithms on low-power embedded devices, such as low-configuration Android cell phones, poses significant challenges. Hence, in order to be compatible with low-configuration devices, it is necessary to develop a lightweight model with reduced parameters and layers, which will take less training time compared to the current state-of-the-art models. Beyond that, no studies have shown the use of explainable Artificial Intelligence (AI) techniques, such as Grad-CAM or SHapley Additive exPlanations (SHAP).

III. MATERIALS AND METHODS

Figure 1 demonstrates a flowchart illustrating the entire methodology of this research. The following subsections provide a step-by-step description of the conducted approach. The primary objective of the proposed methodology is to broaden and enhance the capabilities of the pre-trained MobileNetV3Small model to perform the classification task. The architecture of the MobileNetV3Small has been modified by putting on extra convolutional and fully connected layers.

A. IMAGE ACQUISITION

Collaborating with seasoned experts from the Bangladesh Sericulture Development Board, the dataset contains two primary mulberry leaf diseases for the study—leaf spot and leaf rust. High-resolution images were taken of mulberry leaves at various locations, such as Mirganj, Bagha, and Vodra in Rajshahi, Bangladesh. The dataset collected for this study consists of 1091 images. These images have been categorized by sericulture specialists into three distinct classes. There are 440 images of healthy mulberry leaves. 489 images of leaves that have been affected by leaf rust, and an additional 162 images with symptoms of leaf spots. Every image in the dataset is high resolution ($4{,}000 \times 6{,}000$ pixels).

The expert who categorized these images is from Bangladesh's primary center for sericulture research and has experience with at least ten years of specialized knowledge, thereby adding credibility and accuracy to the categorization and subsequent analysis of the collected leaf images. **Figure 2** illustrates several instances of the obtained images and **Table 2** shows the description of the dataset.

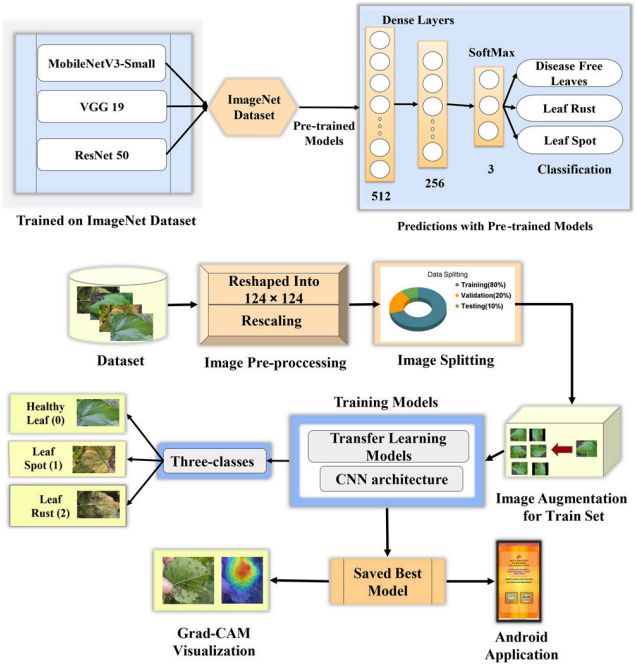

FIGURE 1. Proposed methodology.

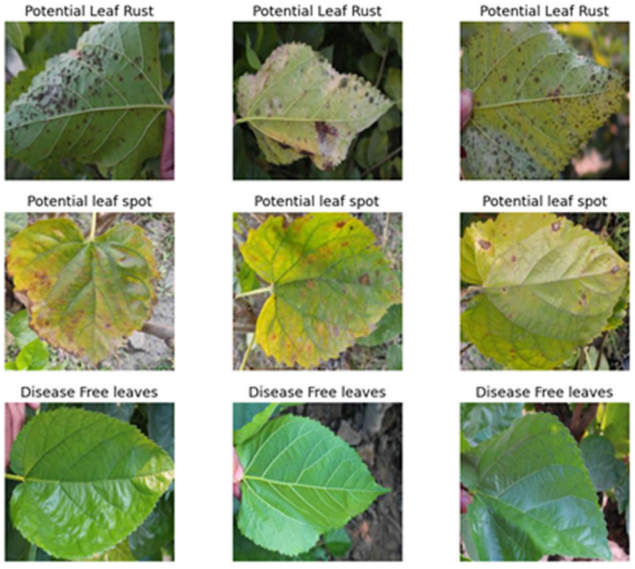

FIGURE 2. Obtained images.

B. DATA PRE-PROCESSING

Preprocessing the images of mulberry leaves is an essential step before classification, as the quality of the dataset significantly affects the accuracy of the classification results. The process is completed through the following sequential actions:

1) DATASET SCALING / RESIZING

The quality of the data preprocessing directly affects the classification's accuracy. The objective of this study is to streamline image processing algorithms in order to make them easier to use on embedded devices. In the preprocessing stage, the images within the dataset undergo a reduction in both width and height dimensions to 124 pixels. This reduction serves the purpose of minimizing storage requirements and optimizing processing resources. To reduce the intricacy of the images, a process known as normalizing is performed. This process entails rescaling the pixel values from their initial range of 0-255 to a redefined range of 0-1. To accomplish this, the pixel value is divided by 255 [33].

TABLE 2. Dataset description.

Sch.	Types	Training Set		Validation Set
		Before Augmentation	After Augmentation	
Three Class	Disease Free Leaf	308	2,000	44
	Leaf Rust	342	2,000	49
	Leaf Spot	114	2,000	16
Total		764	6,000	109

2) IMAGE AUGMENTATION

Due to the initial dataset's class imbalance, training a robust deep learning model posed challenges, with 764 training images, 218 test images, and 109 validation images. The data quantity was insufficient for deep learning training. To mitigate this, various image augmentation techniques were employed to expand the training set and balance class distributions.

One technique involved randomly rotating images by up to 30 degrees, increasing dataset complexity and model resilience to object orientation changes. Additionally, random flipping between horizontal and vertical orientations was applied to half of the training images, enhancing dataset diversity and model generalization. Arbitrary affine transformations, such as angle changes of 5–15 degrees, translation shifts of 0.1–0.2, and size changes of 0.7–0.8, were also employed to further augment the dataset [34].

Figure 3 showcases some augmented images, illustrating the dataset's expanded flexibility. These augmentation methods generated 6,000 synthetic images from the initial 764 training images, while test and validation datasets were derived from the original and synthetic images.

Expanding the dataset to include 6,000 images for a three-class system and 4,000 images for a two-class scheme ensured each class had 2,000 images. This significantly enhanced the training dataset, improving the model's accuracy and resilience with realistic data. The dataset was carefully curated to be vast, balanced, and representative.

C. MODIFIED PRETRAINED MODEL ARCHITECTURE

For model building, three pretrained models (VGG19 [35], ResNet50 [36], and MobileNets [37]) were utilized,

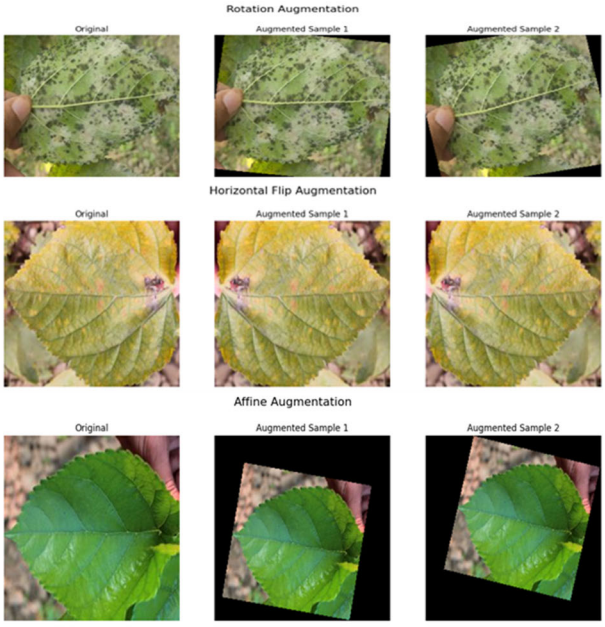

FIGURE 3. Sample augmented images from the original input mulberry leaf image.

with additional convolutional layers added to enhance performance. The modified MobileNetV3Small, designed for mobile and embedded vision applications, was chosen for its adaptability to specific classification tasks through fine-tuning with additional convolutional layers. This approach leveraged the model's powerful representation learning capabilities while customizing its top layers to the dataset, facilitating efficient deployment with minimal computational resources and training data. **Figure 4** demonstrates the CNN architecture of the modified MobileNetV3Small model. The adapted model exhibited strong performance on the classification benchmark, showcasing its versatility for practical deployment.

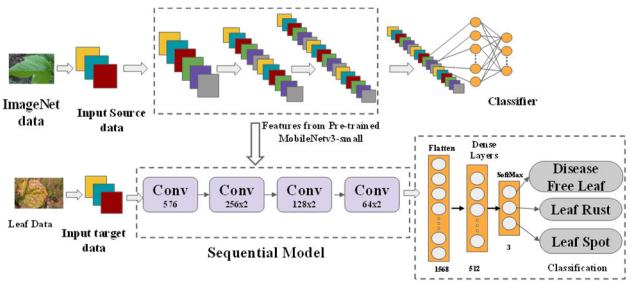

FIGURE 4. CNN architecture.

1) BASE MODEL

As the foundation of our model, we use pre-trained version of the MobileNetV3Small [37], VGG19 [35], and ResNet50 [36] architectures. These architectures are well-known for their efficiency. This study specially focused on MobileNetV3Small for its efficacy in mobile and embedded vision applications. This is mostly because it requires less computational power and takes up less space than other alternatives.

a: MobileNetV3Small

MobileNetV3 Small is designed for efficient computation and includes depth-wise separable convolutions and inverted residual blocks. The operations in MobileNetV3Small can be expressed as:

$$x_{l+1} = H(x_l) + x_l$$

where, $H(x_l)$ is a series of operations including Batch Normalization, ReLU, and depth-wise separable convolutions.

b: VGG19

VGG19 consists of 19 layers with a very deep convolutional network architecture:

$$x_{l+1} = \sigma(W_l * x_l + b_l)$$

where, $(*)$ represents convolution, W_l and b_l are the weights and biases, and σ is the ReLU activation function.

c: ResNet50

ResNet50 employs residual connections to allow gradients to flow through the network effectively:

$$x_{l+1} = x_l + F(x_l, W_l)$$

where F is a residual function consisting of convolutions, batch normalizations, and ReLU activations.

2) ADDITIONAL CONVOLUTIONAL LAYER

This study modifies the architectures of three well-known TL models, ResNet50, VGG19, and MobileNetV3Small, and examines how well they perform on a particular task. Four further convolutional layers, with (512, 128, 64, and 32) number of output channels, were added to the baseline models. The desire to capture more complex and task-specific features in order to perhaps improve the models' representational abilities drove this development. We assessed how these architectural modifications affected the convergence rates, computational efficiency, and accuracy of the models. The proposed adjustments resulted in trade-offs between model complexity and performance increases, as demonstrated by the results of experiments conducted on a benchmark dataset. When compared to the baseline models, the expanded models showed better accuracy, but at the cost of higher processing demands. These results imply that, if the computing overhead is controlled, deliberate design changes may be useful in enhancing transfer learning performance for particular tasks.

We added several convolutional layers on top of the pre-trained model's output. The purpose of these layers is to learn more task-specific features. The operations in these layers can be mathematically described as follows:

$$y_1 = \sigma(W_1 * x_d + b_1)$$

$$y_2 = \sigma\ (W_2 * y_1 + b_2)$$
$$y_3 = \sigma\ (W_3 * y_2 + b_3)$$
$$y_4 = \sigma(W_4 * y_3 + b_4)$$

Here:

- x_d is the output from the batch normalization layer of pretrained models.
- w_i and b_i are the weights and biases of the i^{th} convolutional layer.
- σ is the activation function (ReLU in our case).
- $*$ denotes the convolution operation.

By adding these convolutional layers, the network can capture additional hierarchical features specific to the classification of healthy and scarred mulberry leaves.

A series of four additional convolutional layers have been attached to the model to boost its capacity for representational accuracy. Rectified Linear Unit (ReLU) activation was used after each conv layer. After the ReLU activation, the training process has been given a boost in terms of both consistency and speed by having a Batch Normalization layer. These layers bring the total number of channels down from 576 to 256, then 128 channels, and then finally to 64 and 32 channels. To prevent overfitting, dropout layers with a rate of 0.30 are specifically positioned throughout the model.

3) FULLY CONNECTED LAYERS

Following the extra convolutional blocks, the model consists of a fully connected layer sequence. This sequence's objective is to turn the high-level features that have been learned into a format that is acceptable for classification. It then applies a further ReLU and Dropout layer with a rate of 0.40 for additional regularization, taking the feature maps and flattening them into a 512-dimensional vector. After that, a layer that is completely connected creates a mapping between these features and the final output classes.

4) FORWARD PROPAGATION

The initial step in the forward propagation of the model is to run the input data through the fundamental MobileNetV3 capabilities. After this step, the extra convolutional layers are performed, and finally, the fully connected layers are reshaped to produce the class probabilities or scores.

5) OPTIMIZATION AND TRAINING

The model is trained using an appropriate loss function that has been adapted specifically for the classification job, such as Cross-Entropy Loss. ADAM [38] optimizer has utilized because of its efficacy and versatility. We have high hopes that by using this technique, we will be able to take use of the compact and efficient properties of MobileNetV3Small while simultaneously enhancing its capabilities with our extra layers, so transforming it into an approach that is well suited for the classification task. **Table 3** represents the layers and parameters of the model, while **Table 4** represents the layers and model hypermeters.

TABLE 3. Summary of the proposed model without using the pre-trained model.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(224, 224, 3)	0
MobilenetV3small (Functional)	(7, 7, 576)	939120
conv2d (Conv2D)	(7, 7, 256)	1327360
re_lu_32 (ReLU)	(7, 7, 256)	0
batch_normalization (BatchNormalization)	(7, 7, 256)	1024
dropout (Dropout)	(7, 7, 256)	0
conv2d_1 (Conv2D)	(7, 7, 128)	295040
re_lu_33 (ReLU)	(7, 7, 128)	0
batch_normalization_1 (BatchNormalization)	(7, 7, 128)	512
conv2d_2 (Conv2D)	(7, 7, 64)	73792
re_lu_34 (ReLU)	(7, 7, 64)	0
batch_normalization_2 (BatchNormalization)	(7, 7, 64)	256
dropout_1 (Dropout)	(7, 7, 64)	0
conv2d_3 (Conv2D)	(7, 7, 32)	18464
re_lu_35 (ReLU)	(7, 7, 32)	0
flatten (Flatten)	(1568)	0
dense (Dense)	(512)	803328
dropout_2 (Dropout)	(512)	0
dense_1 (Dense)	(3)	513000
Total params: 3,971,896 Trainable params: 3,031,880 Non-trainable params: 940,016		

TABLE 4. Proposed model's parameters.

Parameter Name	Attribute
Base model	MobileNetV3_Small
Additional Conv Layers	4 (Channels reducing from 576 to 32)
Dropout Rate	Two layers: 0.30 and 0.40
Activation Function	ReLU (Rectified Linear Unit)
Number of Epochs	60
Batch Size	32
Loss Function	Cross-Entropy Loss (Proposed)

D. SOFTWARE APPLICATION DEVELOPMENT

1) ANDROID STUDIO

Android Studio, developed by JetBrains' IntelliJ IDEA, is the authorized integrated development environment (IDE) for creating Android applications. It provides a wide range of advanced features, such as a powerful code editor, a user interface (UI) designer, and a build system based on Gradle. The software enables seamless integration of machine learning (ML) with frameworks such as TensorFlow Lite and includes the ML Kit for performing tasks such as text recognition and face identification. The Android Neural Networks API utilizes the computational capabilities of the device's hardware to do computations at a faster rate. Android Studio is a crucial platform for designing intelligent mobile applications because it simplifies the development, optimization, and deployment of ML models with its broad APIs and libraries for data manipulation.

2) TENSORFLOW LITE

TensorFlow Lite facilitates effective ML on Android devices by enabling the direct deployment of lightweight models.

TABLE 5. Modified model's performance.

Model Name	Metrices	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average μ±σ (%)
Modified MobileNetV3-Small	Precision	**0.97**	**0.98**	**0.9**	**1**	**1**	**97±4.12**
	Recall	**0.97**	**0.97**	**0.88**	**1**	**1**	**96.4±4.93**
	F1-Score	**0.97**	**0.97**	**0.88**	**1**	**1**	**96.4±4.93**
	Accuracy	**0.97**	**0.97**	**0.88**	**1**	**1**	**96.4±4.93**
Modified VGG19	Precision	0.96	0.91	0.91	0.96	0.98	94.4±3.21
	Recall	0.96	0.9	0.91	0.96	0.98	94.2±3.49
	F1-Score	0.96	0.9	0.91	0.96	0.98	94.2±3.49
	Accuracy	0.96	0.9	0.91	0.96	0.98	94.2±3.49
Modified ResNet50	Precision	0.93	0.93	0.97	0.97	0.96	95.2±2.05
	Recall	0.92	0.93	0.96	0.96	0.94	94.2±1.79
	F1-Score	0.92	0.93	0.96	0.96	0.95	94.4±1.82
	Accuracy	0.93	0.93	0.96	0.96	0.94	94.4±1.52

TABLE 6. Accuracy comparison with existing work.

Model	Precision	Recall	F1 Score	Accuracy
PDSCNN [46]	95.8	96.2	**96.8**	96.06
Proposed Model	**97**	**96.4**	96.4	**96.4**

It is equipped with a user-friendly API in Android Studio, enabling real-time inference for applications like as image classification, natural language processing (NLP), and recommendation systems. The framework is designed to maximize performance and accuracy on devices with limited resources. The system utilizes hardware acceleration through the Android Neural Networks API and use the TensorFlow Lite Converter to decrease the size of the model without compromising its performance. TensorFlow Lite provides tools for model quantization and data management, offering a comprehensive solution for integrating machine learning into mobile apps to create smarter user experiences.

E. GRADIENT-WEIGHTED CLASS ACTIVATION MAPPING (Grad-CAM)

Model visualization is crucial for analyzing and debugging model performance. Tools like CAM [39], Grad-CAM [40], and Grad-CAM++ [41] offer insights into DL model predictions by highlighting important image areas.

Grad-CAM, for example, generates heatmap saliency maps, showcasing areas of focus for model predictions. These Visualizations enhance model transparency and user trust by revealing model decision-making processes. Additionally, frameworks like Score-CAM can provide category-specific arguments, and it's important to interpret visual explanations rigorously, as gradients alone may not provide accurate explanations. Wang et al. demonstrated Score-CAM's capability to offer specific arguments for both categories and locations [42], while Adebayo et al. cautioned against relying solely on gradients for model interpretations, emphasizing the need for thorough analysis [43].

Despite potential drawbacks of gradient-based methods, the importance of model visualization persists for debugging, feedback, and user trust in artificial intelligence systems. These techniques rely on gradients and weights from convolutional layers to generate "visual explanations" for CNN models, using the gradient of a target concept [44].

IV. EXPERIMENTAL RESULTS

A. PERFORMACE METRICS

The experiment was carried out on the Kaggle platform (www.Kaggle.com) to take advantage of its available Nvidia P100GPU. Specification of the GPU is: GPU Memory: 16 GB, GPU Memory Clock: 1.32 GHz, Performance: 9.3 trillion floating-point operations per second (TFLOPS). To quantify the performance of the created model, many assessment measures were used, including accuracy (ACC), precision (P), recall (R), F1-score, and area under the curve (AUC). The percentage of correctly identified images among all images are referred to as accuracy (ACC). It showed how effective the classification approach was at identifying leaf diseases. True positive (TP) implies a disease-infected leaf was detected as non-infected. False positive (FP) signifies that a non-diseased leaf was mistakenly identified as an infected leaf. disease-infected, whereas false negative (FN) indicates a disease-free state. From **Equation 1** to **Equation 4**, numerous performance measure metrics are defined.

$$\text{Accuracy} = \text{TP} + \text{TN} / \text{TP} + \text{TN} + \text{FP} + \text{FN} \quad (1)$$

$$\text{Precision} = \text{TP} / \text{TP} + \text{FP} \quad (2)$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (3)$$

$$\text{F1Score} = 2 \times \text{Recall} \times \text{Precision} / \text{Recall} + \text{Precision} \quad (4)$$

B. MULTICLASS CLASSIFICATION RESULTS

1) MODEL PERFORMANCE

Table 5 presents a comprehensive performance analysis of three popular deep learning models: MobileNetV3Small,

VGG19, and ResNet50. The evaluation includes various metrics such as precision, recall, F1-score, AUC, and accuracy, employing a five-fold cross-validation methodology.

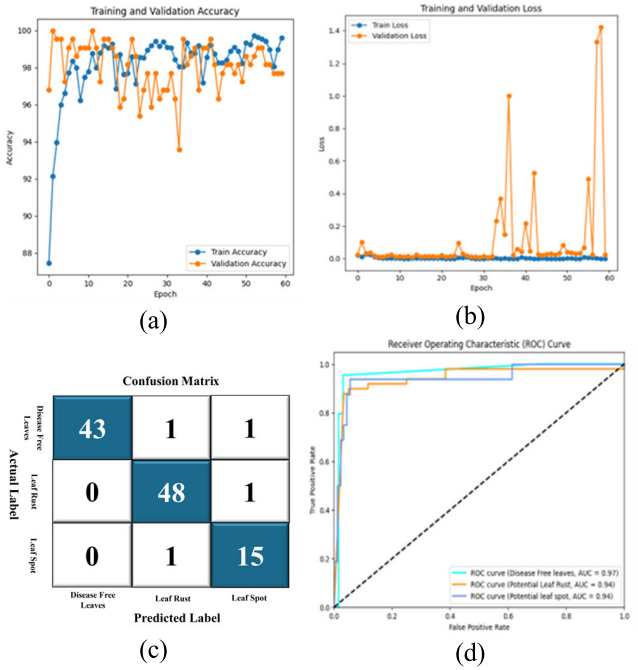

FIGURE 5. (a) Training and validation accuracy curves (b) Loss curves, (c) Confusion matrix and (d) ROC curve for modified MobileNetV3_small model on fold 1.

MobileNetV3Small emerges as the top performer with an impressive average accuracy of 96.4%. While accuracy may be misleading in imbalanced datasets, its high precision (97%) and recall (96.4%) demonstrate effective performance across majority and minority classes. However, the model exhibits the highest standard deviation (4.93%), indicating potential performance variations across different datasets.

VGG19 follows closely with an overall accuracy of 94.2% and a lower standard deviation (3.49%), suggesting more consistent results across diverse data folds. Its balanced AUC reflects efficient management of sensitivity and specificity, making it suitable for balanced classification tasks.

Despite not achieving the highest accuracy, ResNet50 demonstrates exceptional consistency with an average accuracy of 94.4% and a remarkably low standard deviation (1.52%). Its balanced precision and recall rates around 95.2% and 94.2%, respectively, make it a reliable choice for various applications without bias towards positive or negative classes.

In summary, each of the three models possesses distinct advantages. MobileNetV3Small excels in raw performance metrics but may exhibit unpredictability across different datasets. VGG19 lags behind in raw numbers but offers a blend of high performance and improved consistency. Despite not being the most accurate, ResNet50 distinguishes itself with exceptional consistency and reliability. **Table 6** presents a comparison with existing work [45], demonstrating that the proposed modified MobileNetV3Small model outperforms

TABLE 7. The results of five TL models with augmentation in multiclass classification.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	AUC (%)
DenseNet121	94.32	90.17	92.77	90.56	97.48
MobileNet	91.68	90.83	91.13	91.48	98.35
MobileNetV3	90.75	90.99	88.77	90.01	96.71
Xception	89.12	86.52	85.45	86.84	94.74
VGG19	89.17	87.53	85.41	85.48	96.45

existing models in all measures except for the F-1 measure by a marginal 0.2%. The maximum values in each field are highlighted in bold. Additionally, accuracy and loss curves per epoch, as well as confusion matrix and ROC curves, are provided for the best fold among the five folds.

Figure 5 illustrates the training and validation accuracy curves, loss curves, confusion matrix, and ROC curve for the modified MobileNetV3Small model on Fold 1. Fluctuations are observed in the training and validation accuracy/loss curves, indicating potential overfitting as the model becomes overly focused on patterns within the training data, rather than acquiring adaptable representations. This phenomenon can be attributed to the use of pretrained weights trained on the ImageNet dataset, where the model may attempt to relate new features from the dataset to existing features from the pretrained weights or learning rates. Additionally, dataset imbalance contributes to inconsistent learning phases.

The confusion matrix reveals specific performance metrics for each class. For the disease-free class, there are 43 true positives, 2 false negatives, 0 false positives, and 65 true negatives. Similarly, for the leaf rust class, there are 48 true positives, 1 false negative, 2 false positives, and 59 true negatives. For the leaf spot class, 15 true positives, 1 false negative, 2 false positives, and 92 true negatives are observed.

The ROC curve indicates the AUC scores for the disease-free, potential leaf rust, and leaf spot classes as 0.97, 0.94, and 0.94, respectively. These scores provide an overall measure of the model's ability to differentiate across classes. Classification mistakes between the leaf spot and leaf rust classes may be attributed to acceptable but imperfect AUC-ROC values, potentially due to visual resemblance, dataset imbalance, or an absence of typical training samples [46].

In this work, a number of cutting-edge deep learning models were used to classify images of mulberry leaves into three class. Overall, the DenseNet121 model was the most accurate (94.32%), with high precision (90.17%) and memory (92.77%), showing that it could reliably identify the difference. Its strong discriminatory power is further shown by its high AUC of 97.48%. With a 91.68% accuracy rate, an AUC of 98.35%, and balanced precision (90.83%) and recall (91.13%), MobileNet proved to be very efficient and effective at this classification job.

The accuracy of MobileNetV3 was 90.75%, and its precision and recall were 90.99% and 88.77%, respectively. The

TABLE 8. Pretrained model's performance.

Model Name	Metrices	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average μ±σ (%)
MobileNetV3-Small	Precision	**0.96**	0.94	**0.91**	0.91	0.95	93.5 ± 0.70
	Recall	**0.96**	**0.96**	0.88	0.90	0.94	93 ± 1.41
	F1-Score	**0.96**	0.95	0.88	0.90	0.95	92 ± 0.70
	Accuracy	**0.96**	0.94	0.88	0.90	0.94	92.5 ± 1.41
VGG19	Precision	0.82	0.84	0.90	0.88	**0.96**	88.00 ± 4.89
	Recall	0.84	0.82	0.90	0.88	0.95	87.8 ± 4.58
	F1-Score	0.84	0.82	0.90	0.88	0.95	87.8 ± 4.58
	Accuracy	0.82	0.83	0.90	0.88	**0.96**	90.89 ± 4.93
ResNet50	Precision	0.93	0.85	0.84	**0.94**	0.90	91.2 ± 2.05
	Recall	0.93	0.89	0.83	**0.94**	0.90	90.2 ± 1.79
	F1-Score	0.93	0.89	0.83	**0.94**	0.90	91.4 ± 1.82
	Accuracy	0.93	0.89	0.83	**0.94**	0.90	**94.4 ± 1.52**

AUC for this model stayed high at 96.71%, showing good results. With an accuracy of 89.12% and an AUC of 94.74%, Xception did a good job, but its precision (86.52%) and memory (85.45%) was a little lower, which suggests that it could do better at telling the difference between scarred leaves. VGG19 got an impressive AUC of 96.45%, which shows that it can do the classification job even though it isn't as advanced as some newer models. Its accuracy was 89.17%, its precision was 87.53%, and its recall was 85.41%. **Table 7** presents the result of five transfer learning (TL) models.

2) ABLATION STUDY

The results of the ablation study presented in **Table 8** demonstrate that fine-tuning pretrained models through targeted modifications and adjustments tailored to the specific dataset leads to enhanced performance compared to using unmodified pretrained models. Evaluation results reveal that the modified versions of the three base architectures—MobileNetV3Small, VGG19, and ResNet50—consistently demonstrate superior performance across various metrics including average precision, recall, F1 scores, and accuracy across all five folds of the evaluation process. Notably, the modified MobileNetV3Small model achieves exceptionally high metric scores of 1.0 on specific folds, a significant improvement from its original score of 0.96. Similarly, the modified versions of VGG19 and ResNet50 show a notable improvement of approximately 3-4% in accuracy compared to their pretrained counterparts.

The presence of lower standard deviations in the adapted models' performance across folds suggests a higher level of reliability. Through fine-tuning, pretrained model weights are adjusted to align with the specific dataset and task, enhancing their alignment with the underlying data distribution and subsequently improving performance and accuracy. Fine-tuning enables models to significantly enhance precision and recall rates in identifying positive cases. Overall, pretrained models demonstrate notable performance enhancements compared to off-the-shelf models for image classification tasks. The accuracy, loss curves by epoch, and confusion matrix and ROC curves are provided for the best fold among the five.

C. ANDROID APPLICATION

To showcase the established model's real-world application, a functional prototype of an Android app was developed, highlighting its capabilities. The CNN model, after being trained, was converted into the TFLite model format to create an Android app with a user interface (UI) for classifying diseased leaf images on an Android smartphone. We employed model quantization techniques to convert the model from 32-bit floating-point to 8-bit integer representations. This process significantly reduces the model size and the number of computations required, resulting in lower power consumption and faster inference times. Quantized models are well-supported on mobile devices through frameworks like TensorFlow Lite.

Initially, the Android app loads the model into memory before creating a TFLite Interpreter. This Interpreter imports the input leaf picture into the input tensor, which is then scaled to 224 × 224 pixels. The TFLite Image Processor API is utilized to normalize the data. Subsequently, the Interpreter executes the inference, providing the classification results in the output tensor. The TFLite model can classify a total of two disease classes in addition to a healthy leaf image.

Figure 6 describes the working procedure of the android application through a flowchart. **Figure 7** showcases the various interfaces and functionalities of the Android application. The "Opening activity" depicted in **Figure 7(a)** is followed by the "Landing activity," shown in **Figure 7(b)**, after a brief interval. This activity includes two buttons labeled "Scan" and "Load". The scan activity involves capturing an image using the smartphone's camera activity and then classifying it. On the other hand, the load activity first navigates to the check activity (**Figure 7(c)**), where it loads an image from the phone's gallery, as depicted in **Figure 7(d)**. Simply tapping "load image" allows the selection of an image from the gallery and pressing the "CHECK" button imports the leaf image into the smartphone for classification. Once the proposed model processes the imported image, the program displays the class of the image, as shown in **Figure 7**, even for images from the testing dataset that the proposed model has not encountered before.

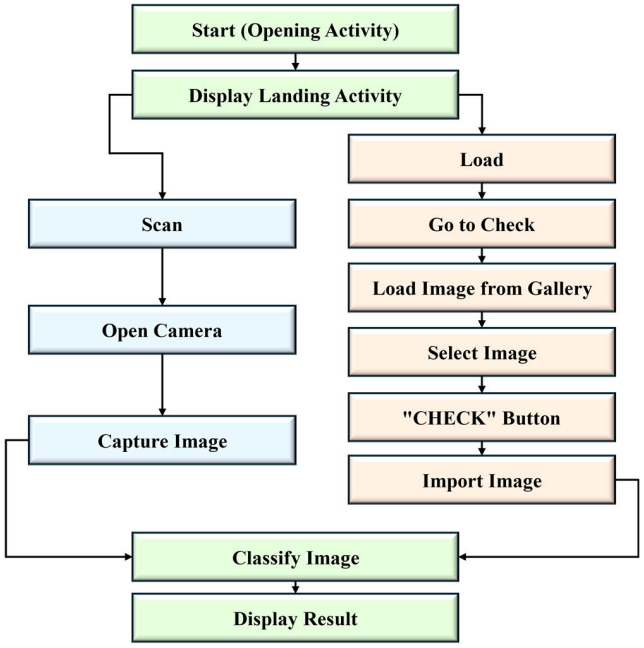

FIGURE 6. Flowchart of the android application functionality.

Figure 8 illustrates the test results for accuracy and inference runtime. It took between 4 and 7 milliseconds to perform inference on sample leaf photos and detect cotton and tomato leaf diseases with higher accuracy and speed than a manual assessment by a plant diseases specialist. All experiments were conducted using GPU acceleration, although it was not necessary to categorize an image. The Poco X2 smartphone with Android 11 and a Qualcomm SDM730 Snapdragon 730G chipset, featuring a CPU Octa-core (2 × 2.2 GHz Kryo 470 Gold & 6 × 1.8 GHz Kryo 470 Silver), and a GPU Adreno 618, was utilized for all tests. The Android app can identify two different diseases affecting plant leaves, classifying each image in significantly less time (up to 12ms). With careful model selection, quantization, TensorFlow Lite use, on-demand inference it greatly reduced the amount of battery life that deep learning algorithms on mobile devices use. To keep a good balance between performance and energy economy, these changes make sure that our mobile app can correctly and quickly classify images of mulberry leaves.

D. GRAD-CAM MODEL VISUALIZATION

Visualizations clarify model decision-making processes, hence increasing transparency and boosting user trust. **Figure 9** displays three randomly selected images from different categories of the original dataset. These images are utilized to objectively assess the model's performance. Grad-CAM showcases the model's capacity to concentrate on the most pertinent regions of an image, effectively classifying various types of leaves. The red and yellow sections represent pivotal portions in the example images that are crucial for the decision-making process. Grad-CAM provides a visual representation of the precise regions that the model directs its attention towards. Visualization approaches greatly improve

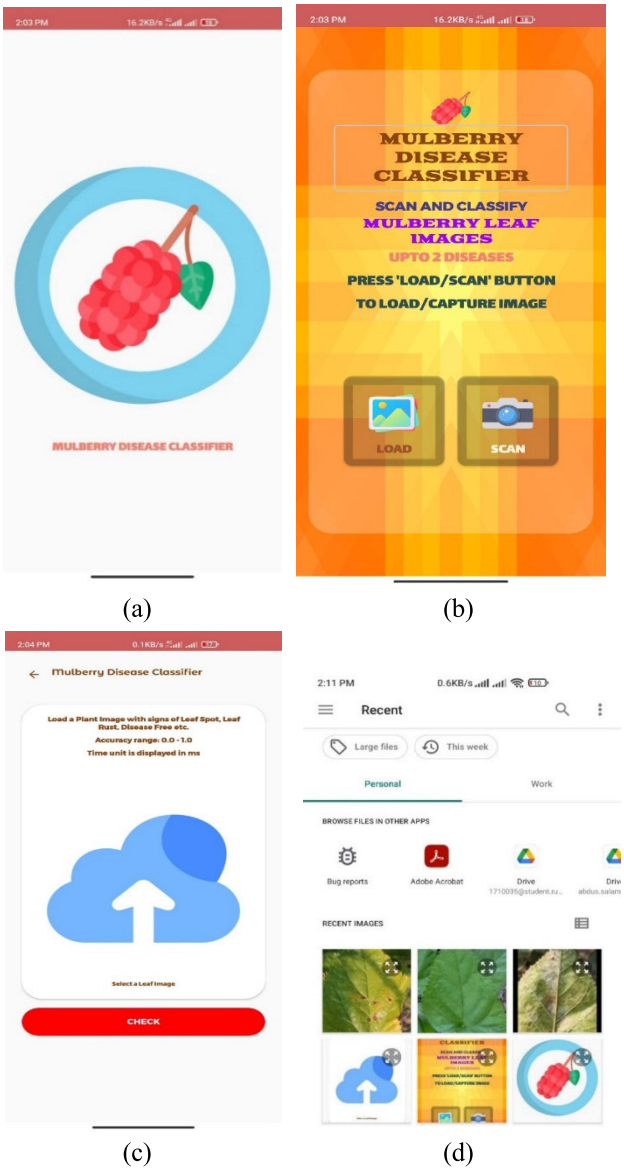

FIGURE 7. Images of the android application activities (a) Opening activity, (b) Landing page, (c) Checking activity, (d) Select image from gallery.

the interpretability of deep learning models by emphasizing important characteristics, and areas.

V. DISCUSSIONS AND FUTURE WORK

The study has demonstrated the significance of utilizing AI technology to tackle agricultural difficulties, mainly in the silk production sector where mulberry trees are vulnerable to diseases that can have an influence on productivity. Utilizing transfer learning, a lightweight model has been created by adapting MobileNetV3Small, resulting in a notable progress in disease identification in the field of agriculture. This research has significantly contributed to the successful development of a Convolutional Neural Network (CNN) model for classifying mulberry leaf diseases. Using explainable AI (XAI) approaches such as Grad-CAM to see and interpret models enhances transparency and trust in the

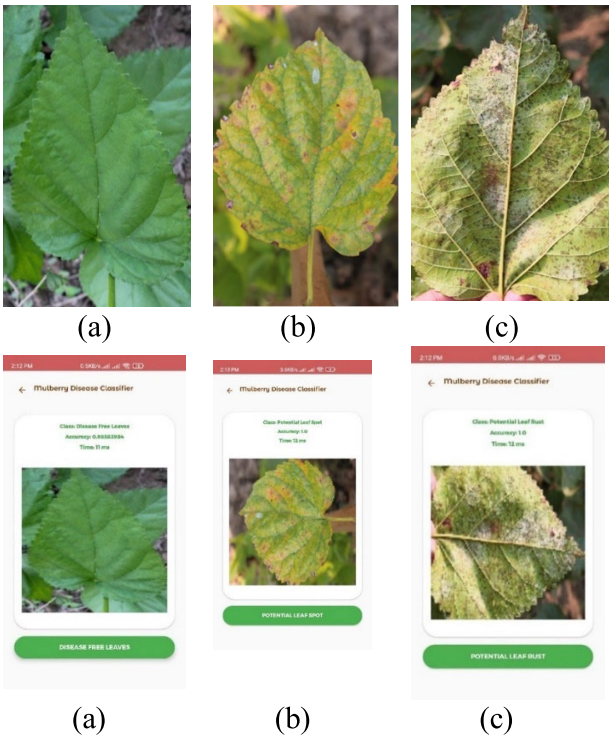

FIGURE 8. Examples of the android App TFLite's classification of images of (a)Disease free leaves, (b)Leaf spot, and (c)Leaf rust input images. (a'), (b'), and (c') represents the output from the App for (a)Disease free leaves, (b)Leaf spot, and (c)Leaf rust.

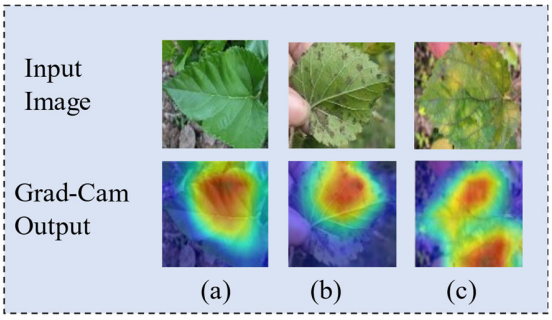

FIGURE 9. Original leaf images and Grad-CAM output for (a) Leaf rust, (b) Leaf spot, (c) Healthy leaves.

decision-making process of the proposed approach. The document demonstrates the experimental outcomes showcasing superior performance of the suggested model in plant pathology when compared to existing state-of-the-art methodologies. The dataset collection is comprehensive and includes healthy leaves, leaves afflicted by leaf rust, and leaves affected by leaf spot. This collection allows for a complete investigation of multiple disease categories to train and validate robust models.

Potential future developments in the automated classification of mulberry diseases may incorporate cutting-edge technologies like drones with strategically mounted cameras. These technologies will play a vital role in gathering comprehensive data on plant diseases, allowing for early identification, ongoing surveillance, and accurate measurement of the affected areas in crops. Drones have the capability to provide real-time aerial images across vast areas, whereas static cameras provide meticulous monitoring and examination of plant well-being. Timely identification and ongoing surveillance will facilitate well-informed decision-making, enhancing the efficient allocation of resources and enhancing silk production output and sustainability.

VI. CONCLUSION

Several transfer learning models, including modified MobileNetV3Small, were trained and tested for mulberry leaf disease detection. Modified MobileNetV3Small showed superior accuracy, albeit with higher performance variability. VGG19 proved to be a robust option with slightly lower accuracy, while ResNet50 offered a compelling balance of accuracy and consistency, characterized by its notably low standard deviation. The selection of the model was based on performance and model size, with MobileNetV3Small having the smallest size in megabytes, making it suitable for building Android applications. The app size remained below 100MB, which is optimal for mobile applications running machine learning models. Although the study results demonstrate the promising nature of the proposed framework, further research is warranted. Testing the model on additional datasets, such as PlantVillage and PlantDoc [47], will assess its generalizability and adaptability. Incorporating a broader range of plant diseases, disease development phases, and image quality metrics from these datasets will challenge the model to handle real-world variability. Rigorous testing on multiple datasets will help identify any biases or limitations in the model's generalizability.

ACKNOWLEDGMENT

The authors extend their appreciation to AI tools, such as ChatGPT, QuillBot, and Grammarly for their invaluable assistance in refining and enhancing the clarity of the writing.

REFERENCES

[1] E. M. Sánchez-Salcedo, P. Mena, C. García-Viguera, J. J. Martínez, and F. Hernández, "Phytochemical evaluation of white (*Morus alba* L.) and black (*Morus nigra* L.) mulberry fruits, a starting point for the assessment of their beneficial properties," *J. Funct. Foods*, vol. 12, pp. 399–408, Jan. 2015, doi: 10.1016/j.jff.2014.12.010.

[2] M. Gundogdu, F. Muradoglu, R. I. G. Sensoy, and H. Yilmaz, "Determination of fruit chemical properties of *Morus nigra* L., *Morus alba* L. and *Morus rubra* L. by HPLC," *Scientia Horticulturae*, vol. 132, pp. 37–41, Dec. 2011, doi: 10.1016/j.scienta.2011.09.035.

[3] S. Iqbal, U. Younas, Sirajuddin, K. W. Chan, R. A. Sarfraz, and M. K. Uddin, "Proximate composition and antioxidant potential of leaves from three varieties of mulberry (*Morus* sp.): A comparative study," *Int. J. Mol. Sci.*, vol. 13, no. 6, pp. 6651–6664, May 2012. [Online]. Available: https://sciprofiles.com/profile/358570?utm_source=mdpi.com&utm_medium=website&utm_campaign=avatar_name, doi: 10.3390/ijms13066651.

[4] K. Prashar, R. Talwar, and C. Kant, "CNN based on overlapping pooling method and multi-layered learning with SVM & KNN for American cotton leaf disease recognition," in *Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM)*, Apr. 2019, pp. 330–333.

[5] R. V. Kumar, S. Chauhan, D. Kumar, and N. More, "Nutritional composition in leaves of some mulberry varieties:- a comparative study," in *Proc. Int. Conf. Bioinf. Biomed. Technol.*, Apr. 2010, pp. 438–442.

[6] D. Wang, Z. Dong, Y. Zhang, K. Guo, P. Guo, P. Zhao, and Q. Xia, "Proteomics provides insight into the interaction between mulberry and silkworm," *J. Proteome Res.*, vol. 16, no. 7, pp. 2472–2480, Jun. 2017.

[7] L. Ghose, F. A. Neela, T. C. Chakravort, M. R. Ali, and M. S. Alam, "Incidence of leaf blight disease of mulberry plant and assessment of changes in amino acids and photosynthetic pigments of infected leaf," *Plant Pathol. J.*, vol. 9, no. 3, pp. 140–143, Jun. 2010.

[8] R. Zhang, Q. Zhang, S. Zhu, B. Liu, F. Liu, and Y. Xu, "Mulberry leaf (*Morus alba* L.): A review of its potential influences in mechanisms of action on metabolic diseases," *Pharmacol. Res.*, vol. 175, Jan. 2022, Art. no. 106029.

[9] Y. Kurmi, P. Saxena, B. S. Kirar, S. Gangwar, V. Chaurasia, and A. Goel, "Deep CNN model for crops' diseases detection using leaf images," *Multidimensional Syst. Signal Process.*, vol. 33, no. 3, pp. 981–1000, Sep. 2022, doi: 10.1007/s11045-022-00820-4.

[10] N. S. Shadin, S. Sanjana, S. Chakraborty, and N. Sharmin, "Performance analysis of glaucoma detection using deep learning models," in *Proc. Int. Conf. Innov. Sci., Eng. Technol. (ICISET)*, Feb. 2022, pp. 190–195, doi: 10.1109/ICISET54810.2022.9775828.

[11] Y. Kurmi, S. Gangwar, D. Agrawal, S. Kumar, and H. S. Srivastava, "Leaf image analysis-based crop diseases classification," *Signal, Image Video Process.*, vol. 15, no. 3, pp. 589–597, Apr. 2021, doi: 10.1007/s11760-020-01780-7.

[12] J. Andrew, 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, doi: 10.3390/agronomy12102395.

[13] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[14] T. T. Høye, J. Ärje, K. Bjerge, O. L. P. Hansen, A. Iosifidis, F. Leese, H. M. R. Mann, K. Meissner, C. Melvad, and J. Raitoharju, "Deep learning and computer vision will transform entomology," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 2, Jan. 2021, Art. no. e2002545117, doi: 10.1073/pnas.2002545117.

[15] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, and Y. Hu, "Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105730.

[16] Q. Wu, Y. Chen, and J. Meng, "DCGAN-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 98716–98728, 2020, doi: 10.1109/ACCESS.2020.2997001.

[17] H. Chen, Y. Wang, P. Jiang, R. Zhang, and J. Peng, "LBFNet: A tomato leaf disease identification model based on three-channel attention mechanism and quantitative pruning," *Appl. Sci.*, vol. 13, no. 9, p. 5589, Apr. 2023, doi: 10.3390/app13095589.

[18] W. Gómez-Flores, J. J. Garza-Saldaña, and S. E. Varela-Fuentes, "A huanglongbing detection method for orange trees based on deep neural networks and transfer learning," *IEEE Access*, vol. 10, pp. 116686–116696, 2022, doi: 10.1109/ACCESS.2022.3219481.

[19] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaei, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE Access*, vol. 9, pp. 112942–112954, 2021, doi: 10.1109/ACCESS.2021.3096895.

[20] M. Masood, M. Nawaz, T. Nazir, A. Javed, R. Alkanhel, H. Elmannai, S. Dhahbi, and S. Bourouis, "MaizeNet: A deep learning approach for effective recognition of maize plant leaf diseases," *IEEE Access*, vol. 11, pp. 52862–52876, 2023, doi: 10.1109/ACCESS.2023.3280260.

[21] H. Amin, A. Darwish, A. E. Hassanien, and M. Soliman, "End-to-end deep learning model for corn leaf disease classification," *IEEE Access*, vol. 10, pp. 31103–31115, 2022, doi: 10.1109/ACCESS.2022.3159678.

[22] M. Lv, G. Zhou, M. He, A. Chen, W. Zhang, and Y. Hu, "Maize leaf disease identification based on feature enhancement and DMS-robust AlexNet," *IEEE Access*, vol. 8, pp. 57952–57966, 2020, doi: 10.1109/ACCESS.2020.2982443.

[23] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, and A. A. Khan, "Detection of apple plant diseases using leaf images through convolutional neural network," *IEEE Access*, vol. 11, pp. 6594–6609, 2023, doi: 10.1109/ACCESS.2022.3232917.

[24] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, Dec. 2017, doi: 10.3390/sym10010011.

[25] N. N. Malvade, R. Yakkundimath, G. Saunshi, M. C. Elemmi, and P. Baraki, "A comparative analysis of paddy crop biotic stress classification using pre-trained deep neural networks," *Artif. Intell. Agricult.*, vol. 6, pp. 167–175, Jan. 2022, doi: 10.1016/j.aiia.2022.09.001.

[26] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019, doi: 10.1109/ACCESS.2019.2943454.

[27] J. Liu, C. Zhang, Q. Qi, H. Li, and L. Du, "MixNet-CA: A novel disease identification method for Chinese roses based on MixNet-s," *IEEE Access*, vol. 11, pp. 97538–97548, 2023, doi: 10.1109/ACCESS.2023.3313177.

[28] H. Yu, J. Liu, C. Chen, A. A. Heidari, Q. Zhang, H. Chen, M. Mafarja, and H. Turabieh, "Corn leaf diseases diagnosis based on K-means clustering and deep learning," *IEEE Access*, vol. 9, pp. 143824–143835, 2021, doi: 10.1109/ACCESS.2021.3120379.

[29] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, Jun. 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[30] Y. Luo, J. Sun, J. Shen, X. Wu, L. Wang, and W. Zhu, "Apple leaf disease recognition and sub-class categorization based on improved multi-scale feature fusion network," *IEEE Access*, vol. 9, pp. 95517–95527, 2021, doi: 10.1109/ACCESS.2021.3094802.

[31] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention based neural network and Bayesian optimization," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 114770, doi: 10.1016/j.eswa.2021.114770.

[32] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Comput. Electron. Agricult.*, vol. 142, pp. 369–379, Nov. 2017, doi: 10.1016/j.compag.2017.09.012.

[33] M. Nahiduzzaman, L. F. Abdulrazak, M. A. Ayari, A. Khandakar, and S. M. R. Islam, "A novel framework for lung cancer classification using lightweight convolutional neural networks and ridge extreme learning machine model with Shapley additive explanations (SHAP)," *Expert Syst. Appl.*, vol. 248, Aug. 2024, Art. no. 123392, doi: 10.1016/j.eswa.2024.123392.

[34] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[41] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847, doi: 10.1109/WACV.2018.00097.

[42] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 111–119, doi: 10.1109/CVPRW50498.2020.00020.

[43] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Montréal, QC, Canada, Canada, 2018, pp. 1–30.

[44] M. Nahiduzzaman, M. R. Islam, and R. Hassan, "ChestX-ray6: Prediction of multiple diseases including COVID-19 from chest X-ray images using convolutional neural network," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118576, doi: 10.1016/j.eswa.2022.118576.

[45] M. Nahiduzzaman, M. E. H. Chowdhury, A. Salam, E. Nahid, F. Ahmed, N. Al-Emadi, M. A. Ayari, A. Khandakar, and J. Haider, "Explainable deep learning model for automatic mulberry leaf disease classification," *Front. Plant Sci.*, vol. 14, 2023, Art. no. 1175515, doi: 10.3389/fpls.2023.1175515.

[46] *A Deep Dive Into Learning Curves in Machine Learning | Ml-Articles—Weights & Biases*. Accessed: Jan. 6, 2024. [Online]. Available: https://wandb.ai/mostafaibrahim17/ml-articles/reports/A-Deep-Dive-Into-Learning-Curves-in-Machine-Learning–Vmlldzo0NjA1ODY0?fbclid=IwAR3epf3cWqlhiQHBvuGENZTdt1H9dieEcABcELd45y3PpCIYzVVACytt0yQ

[47] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *Proc. 7th ACM IKDD CoDS 25th COMAD*, Jan. 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

ABDUS SALAM received the degree in electrical and computer engineering from the Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh.

Currently, he is a Research Assistant with the Machine Learning Group, Qatar University, where he conducts research in his areas of interest. These include machine learning applications in domains, such as disease detection, classification, segmentation, in medical images. His research is specifically centered toward creating advanced automated systems by image processing. His goal is to continue advancing his research skills and contribute innovative AI solutions to improve medical image analysis, food security, and agricultural sustainability on a global scale. With his demonstrated expertise and dedication to the field. He has coauthored a article published in *Frontiers in Plant Science*. He is motivated by a strong desire to apply artificial intelligence to improve society. Furthermore, he is committed to improving current deep learning models, making them more relevant to various real-life situations, while also lowering their complexity and increasing their interpretability. Research Gate Profile: https://www.researchgate.net/profile/Abdus-Salam-45.

MANSURA NAZNINE received the B.Sc. degree in computer science and engineering from RUET, in 2023. Fueled by a passion for cutting-edge technology, she envisions a future, where she will pursue advanced research in artificial intelligence and machine learning. Building on her strong foundation in image processing and data compression, she aims to explore novel applications in these fields. Her research interests encompass a diverse range of fields within computer science, with a primary focus on data compression, disease detection, and computer vision, particularly in the domain of classification. Her work in data compression reflects a commitment to developing efficient and innovative techniques to reduce the size of digital data while preserving its essential information. In the realm of disease detection, she explores cutting-edge technologies and methodologies to enhance early diagnosis and monitoring through computational approaches. Her future goals involve pursuing higher education to deepen her understanding and collaborating on interdisciplinary projects that align with her interest in the intersection of technology and environmental sustainability. Research Gate Profile: https://www.researchgate.net/profile/Mansura-Naznine.

NUSRAT JAHAN received the B.Sc. degree in electrical and computer engineering from the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh, in 2021. She has a year of experience as an AI Engineer with multinational software company in Bangladesh. She has published IEEE conference papers in relevant fields. Her research interests include machine learning, deep learning applications in disease detection and agriculture, object detection, and natural language processing. Research Gate Profile: https://www.researchgate.net/profile/Nusrat-Jahan-122.

EMAMA NAHID received the degree in electrical and computer engineering from the Rajshahi University of Engineering and Technology (RUET), Bangladesh. With a passion for cutting-edge technologies, she is eager to make a mark in the fields of machine learning, artificial intelligence, disease detection, and cybersecurity. During her academic journey with RUET, she has demonstrated a keen interest in understanding the intersection of technology and societal challenges. This interest led her to delve into the realm of machine learning, where she developed a profound understanding of its applications, particularly in the areas of disease detection and cybersecurity. Her commitment to academic excellence and her innovative mindset are underscored by her coauthorship of a notable article published in *Frontiers in Plant Science*. Research Gate Profile: https://www.researchgate.net/profile/Emama-Nahid.

MD NAHIDUZZAMAN received the Bachelor of Science degree in computer science and engineering from the Rajshahi University of Engineering and Technology (RUET), with an impressive CGPA of 3.85 out of 4.00 and ranking fifth among a competitive cohort of 120 students.

He is a highly accomplished and a passionate researcher from Bangladesh, has consistently demonstrated his commitment to academic excellence and innovation throughout his career. He has been taught and researched with the Electrical and Computer Engineering Department, RUET, since November 2019. His dedication to academics and community includes teaching future engineers. To improve society using AI, he carried out research on sophisticated automated disease detection. These innovative medical image analysis technologies use explainable AI and powerful machine and deep learning models. He enhances deep learning models' interpretability and practicality. Many IEEE conference papers and 14 high-impact Q1 journal publications belong to him. University Grants Commission grant helped him build an ensemble-based machine learning deep learning model for CT scan COVID-19 detection. To enhance his studies, he collaborated with prominent abroad universities, including Manchester Metropolitan University. As Qatar University Machine Learning Laboratory Research Assistant, he follows his interests include medical image analysis, power, and disease classification using machine learning. Research Gate Profile: https://www.researchgate.net/profile/Md-Nahiduzzaman-7.

MUHAMMAD E. H. CHOWDHURY (Senior Member, IEEE) received the Ph.D. degree from the University of Nottingham, U.K., in 2014. He was a Postdoctoral Research Fellow with the Sir Peter Mansfield Imaging Centre, University of Nottingham. He is currently an Assistant Professor with the Department of Electrical Engineering, Qatar University. He has filed several patents and published more than 200 peer-reviewed journal articles, more than 30 conference papers and several book chapters. He is currently running NPRP, UREP, and HSREP grants from Qatar National Research Fund (QNRF) and internal grants (IRCC and HIG) from Qatar University along with academic projects from HBKU and HMC. His current research interests include biomedical instrumentation, signal processing, wearable sensors, medical image analysis, machine learning and computer vision, embedded system design, and simultaneous EEG/fMRI. He is a member of British Radiology, ISMRM, and HBM. He has won the COVID-19 Dataset Award, the AHS Award from HMC, and the National AI Competition Awards for contributing to the fight against COVID-19. His team was the Gold Medalist at the 13th International Invention Fair in the Middle East (IIFME). He has been listed among the Top 2% of scientists in the World List, published by Stanford University. He serves as a Guest Editor for *Polymers*, an Associate Editor for IEEE Access, and a Topic Editor and a Review Editor for *Frontiers in Neuroscience*.