Detection and Classification of Mango Leaf Diseases Utilizing Convolutional Neural Network Models

by

Md.Rubaiat Ferdous
19101299
Abdulla Al Amin
19301083
Senjuti Sarkar Shaily
20301359
Fabliha Tarannum Ricky
20301048
Tahmeed Tabeeb
20101582

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
May 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md.Rubaiat Ferdous
19101299

Abdulla Al Amin
19301083

Senjuti Sarkar Shaily
20301359

Fabliha Tarannum Ricky
20301048

Tahmeed Tabeeb
20101582

Approval

The thesis/project titled "Detection and Classification of Mango Leaf Diseases Utilizing Convolutional Neural Network Models" submitted by

1. Md.Rubaiat Ferdous(19101299)
2. Abdulla Al Amin(19301083)
3. Senjuti Sarkar Shaily(20301359)
4. Fabliha Tarannum Ricky(20301048)
5. Tahmeed Tabeeb (20101582)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on May 27, 2024.

Examining Committee:

Supervisor:
(Member)

Md. Saiful Islam
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Ms. Anika Tasnim
Contractual Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Plant diseases, particularly those affecting crop plants, pose a significant danger to world food security by compromising the quality and yield of agricultural produce. Mango leaf disease is one such example. Mango leaf diseases are quite harmful since they can significantly lower crop yields of mangos, both in quantity and quality. Therefore, it is critical to identify leaf diseases in crops like mangos as soon as possible in order to take prompt preventive action. In large cultivated areas where mangoes are planted in significant quantities, the amount of manual inspection can be significantly reduced by mechanizing the process of disease recognition. With their exceptional ability to identify complex patterns in images, Convolutional Neural Networks (CNN) have great potential for automating the identification of illnesses affecting mango leaves. CNNs have been used in several studies with impressive accuracy rates, opening the door to improved crop management techniques and early disease diagnosis. In order to identify various illnesses, a research study using CNN—one of the most advanced deep learning algorithms—is presented in this work. CNN is used to segment and classify pictures of mango leaves. A number of CNN models, including Xception, MobileNetV2, ResNet50, DenseNet201, and ResNet50, have been used to accurately identify and categorize diseases, which will ultimately improve the production and health of mango crops. MangoLeafBD, a dataset that was acquired from Kaggle, has been used, and XAI (Explainable Artificial Intelligence), specifically Grad-CAM (Gradient-weighted Class Activation Mapping) and LIME (Local Interpretable Model-agnostic Explanations), were used to understand the rationale behind the decisions made by the applied models. Using these CNN models approach, it was observed that ResNet50 performs better than other models with 98% F1 score in identification and classification of mango leaf diseases.

Keywords: Mango Leaf Disease, Mango, Convolutional Neural Network(CNN), MobileNetV2, ResNet50, DenseNet201, Xception, Deep learning (DL), Explainable Artificial Intelligence (XAI), Machine learning (ML)

Table of Contents

Chapter 1

Introduction

Plant diseases cast a long shadow over agricultural production, causing quantitative losses and reducing the quality of crops. These losses ripple through the entire agricultural ecosystem, inflating production costs and decreasing profits. More alarmingly, they jeopardize the livelihoods of farmers and disrupt the food supply chain, posing a direct threat to a nation's nutritional security. Recognized as one of the most appetizing and important fruit crops in horticulture, mangoes are found on tables throughout the world in a variety of forms, including ripe, fresh mangoes and processed foods like sweet raw mango pickles or slices. However, when the plant is harmed by illnesses that impair photosynthesis, it can become challenging to produce this tropical fruit, which ultimately affects the crops. Farmers must take proactive measures to reduce the threat of these diseases, particularly those who cultivate on smaller, easier-to-manage plots. Early disease detection and pest control can be achieved through organic pesticides or minimal chemical intervention for these individuals, but for larger-scale agricultural operations, the magnitude of constant monitoring and early identification becomes a formidable challenge. Delayed detection often leads to severe outbreaks of diseases and pest infestations that can hardly be controlled using organic means. Consequently, farmers are forced to resort to toxic chemical solutions to protect their yields.

Addressing this dilemma involves automating the monitoring process through advanced image processing methods. Remarkably, scant research has delved into the realm of mango leaf disease recognition, and the majority of existing approaches still need to lean on traditional machine learning algorithms. Historically, plant disease detection has relied on the visual acumen of farmers and plant pathologists, who make decisions based on their accumulated experiences. This approach, however, is tainted by inaccuracies, biases, and the limitation that early-stage diseases often display similarity. The agriculture sector urgently requires accurate disease detection tools bolstered by comprehensive databases to address these challenges, especially to assist young and less experienced farmers. New avenues for disease detection have been made possible by the development of computer vision and the introduction of Deep Learning (DL) and Machine Learning (ML) algorithms. In addition, an early disease detection strategy is now necessary to protect crops.

The research landscape has seen numerous studies in this sector, with many utilising datasets like "Plant Village" and CNNs as the preferred models [1]. Convolutional neural network (CNN) models have seen a surge in attention recently in a number of image recognition fields, including the identification of plant diseases [2], [3]. Al-

though crop disease detection has advanced significantly with this technology, it is noteworthy that mango leaf disease detection has received relatively less attention and has primarily relied on traditional machine learning techniques. Additionally, a number of fields have shown how effective CNN is at classifying images. The InceptionV3 model, for example, has been successfully applied to identify and categorize bird species within soundscape recordings [4]. People also experimented with facial expression recognition, a challenging job that involves identifying seven main emotions, using a potent hybrid model that combines VGG-16 and ResNet50 [5]. Moreover, deep learning models such as Inception and ResNet have demonstrated inventive performance in tasks like traffic sign recognition and chromosomal categorization, significantly lowering the amount of training data needed [6], [7]. The creation of an image classification model for breast cancer histology is another example of how deep learning is being used in medical imaging to great effect [8]. There is a term in machine learning, "black box", which refers to complex machine learning models, where even the designers of the model may struggle to explain the model's decision. XAI can help us to tackle this problem and enrich our comprehension of the model. If there's any biases, it can detect that and make it easier for debugging and improvements of the models. In this paper, XAI have been implemented to enhance the transparency and understanding of the deep learning model's methods for the humans. Techniques such as Grad-CAM and LIME have been used in achieving this goal. Grad-CAM and LIME both provides visualization and evaluation of the images that our models particularly focused on to make theirs decision. It helps us troubleshooting the problems and as a result improve the models to get a better result. By incorporating these XAI techniques, this research expands the area of plant disease detection and guarantees that the solutions are reliable, practical, and comprehensible, meeting the demands of both seasoned and novice farmers.
In this context, the proposed research attempts to address the lack of studies in mango leaf disease recognition by harnessing the capabilities of renowned CNN models such as MobileNetV2, ResNet50, DenseNet201, and Xception. The implementation of transfer learning among these models aims to create an automated system with enhanced accuracy, making it competitive in the field of mango leaf disease detection [9]–[11]. These approaches are tailored to tackle the multi-class classification problem, focusing on seven mango diseases—Anthracnose, Gall Midge, Powdery Mildew, Sooty Mould, Bacterial Canker, and Die Back. This paper proceeds with an exploration of related works using CNN, followed by the presentation of our models, implementing XAI techniques, results, and discussions, reaching in a conclusion that sheds light on the future of disease detection in agriculture.

1.1 Research Problem

In the southern region of Asia, mango is a highly beloved fruit. Mango cultivation is distributed among 85 countries all over the world[12]. Moreover, the ninth-highest producer of mangoes worldwide is Bangladesh[13]. In fiscal year 2021–2022, almost 23.5 lakh tonnes of mangoes were produced in Bangladesh, according to statistics from the Department of Agricultural Extension(DAE)[14]. Plant diseases are now being a thought of concern for agricultural countries. This alarming issue hampers the year end crops, which causes massive loss in the national economy of the suffering countries. If we study deeply about mango we can observe mango leaves are affected by different diseases. Some diseases are- Anthracnose, Powdery Mildew, Sooty Mould, Bacterial Canker and Die Back. These diseases have created problems for the farmers to find an optimum amount of yield. According to studies, Anthracnose affects roughly 39% of mango trees, while Powdery Mildew can kill up to 23% of untreated plants and Bacterial Canker can reduce mango harvests by 10% to 100% [15]. According to an article published on Jun 18, 2015, the Dieback disease has devastated hundreds of mango trees in the district, Chapainawabganj in Bangladesh. It was a great concern for producers in the region known for delivering a range of premium mangoes. Bordeaux paste may rescue the trees if the disease could be identified in its early stages, according to Dr. Shorof Uddin[16]. A number of mango leaf diseases are caused by environmental factors such as temperature, moisture, soil characteristics, rainfall.

To address these problems in the agricultural sector, we are to find ways using technologies to save the mango yield from these diseases. To diagnose mango leaf diseases, there is a critical need for effective and more accurate methods. Machine learning(ML), a section of Artificial Intelligence (AI) is capable of being applied to identify diseases and deeply analyse the patterns of them. Real world data and objects are used in ML techniques. We have 1800 images in our dataset and after augmentation there are 4000 images. Convolutional Neural Networks (CNNs) are effective deep learning tools frequently employed for tasks involving large datasets and a high number of classes to categorise or identify. We will use the following models of CNN for our purposes - ResNet50, DenseNet201, MobilenNetV2, and Xception. This study presents our proposed methods, discussion about the findings, and conclusions about the mango leaf disease detection at early stage in agriculture.

1.2 Research Objectives

According to a research, there are more than 83 diseases which affect the mango leaf and fruit [12]. If the crop could be protected from several illnesses, worldwide mango production may increase by at least 28% [17]. Our research aims to detect the different kinds of mango leaf diseases and achieve better accuracy with less disadvantages. In Bangladesh, mango seedlings are afflicted with a number of illnesses, but there is no accurate information available on their distribution, occurrence, severity, or epidemiology nationwide[18]. We choose this research field because the farmers face challenges in recognising and controlling mango leaf diseases in the field. However, CNN model determines the mango leaf diseases at an early stage with a high accuracy level which leads to more productive techniques in agriculture. This study will aid in the early detection of mango leaf diseases, enhancing crop management and improving food security in the agricultural sector. We will try to focus on achieving better accuracy than other proposed methods with less computational cost for early and accurate disease detection of mango leaves.

Chapter 2

Literature Review

2.1 Related Works

In modern times, CNN (Convolutional Neural Network) is frequently used to detect mango leaf diseases. Rajbongshi et al. [19] proposed the classification and differentiation of mango leaf diseases using various CNN models - ResNet50, InceptionV3, InceptionResNetV2, ResNet152V2, DenseNet201, and Xception with transfer learning techniques. A raw dataset of around 1000 photos of 5 different classes of mango leaves—Healthy, Powdery Mildew, Anthracnose, Red Rust, and Gall Machi—was acquired as part of the procedure. In order to get better results and more photos, data augmentation was done on the original dataset. Each target class had around 500 images after the augmentation, and it was split into 80% and 20% for the train and test datasets, respectively. The above-mentioned models were trained with the train dataset, and the test dataset was used to evaluate the accuracy of the models. DenseNet201 achieved 98.00% accuracy, the highest among the used models. Inception-ResNetV2, InceptionV3, ResNet50, ResNet152V2, Xception achieved an accuracy of 96.67%, 96.67%, 97.00%, 94.67% and 97.67% respectively.

This paper[20] also used the CNN model to identify and detect mango leaf disease on time. According to the paper, CNN utilizes multilayer perceptions, requiring the least amount of preprocessing, and is the most common model to use for predictions. A dataset of around 980 images was gathered from the Sher-e-Kashmir University of Agriculture Sciences and Technology, Jammu (SKUAST-J). It had four classes - Normal, Anthracnose, Red Rust, and Powdery Mildew. On the dataset, augmented methods such as rotation, reflection, translation, and scaling were performed, and 40 epochs and 35 batch sizes were utilised to train the model after splitting it into 80%-20% for the train and test datasets. After the performance evaluation, 90.36% accuracy was achieved, which can be further improved using the transfer learning method.

Arya et al. [21] compiled potato and mango leaf images from the plantvillage website and Govind Ballabh Pant University of Agriculture and Technology (GBPUAT), respectively. As the paper was trying to classify the images, they stated that CNN and its other pre-trained architecture are the best to use in this case. After resizing the photos, they added augmentation to expand the total amount of images. The dataset was randomly divided between training and test datasets in order to train

CNN and AlexNet, respectively. Although it takes longer to train AlexNet since it has more layers than CNN, AlexNet nevertheless outperformed CNN in terms of accuracy (98.33%) and training efficiency (90.85%) despite having fewer layers. The goal of the study is to further the research and develop a mobile application that can quickly identify sick leaves in order to enhance agricultural productivity.

In this research paper[22] 2286 images were collected, and 66 of them were distorted. After augmentation, the dataset expanded to 8880 photos, of which 7104 were utilised to train the model and the remainder to evaluate it. Local-Contrast-Haze-Reduction (LCHR) was used to reduce background haze and eliminate noise. The texture, geometric, LBP, and colour data that were extracted are put together into a single matrix using the Canonical Correlation Analysis (CCA). Following the implementation of CCA, feature reduction based on Neighborhood-Correlation-Analysis (NCA) was used to prevent any unnecessary information. Since it can build a full-resolution feature of the given photos, the CNN-based Fully-Convolutional-Network (FrCNnet) model was built to remove the concatenation approach from the architecture and for enhanced segmentation of the diseased portion of the mango leaf. In segmentation tasks, it performs better (98.9% accurate) than CNN, Multi-layer Convolutional Neural Network (MCNN), or Mangonet approaches. The Cubic-Support Vector Machine's (SVM) accuracy in the classification issue was 96.2% on CCA-based features, and it rose to 98.8% after subjecting the model to NCA-based feature reduction.

To identify mango leaf disease, Anthracnose, Kumar et al. [23] used Adam Optimizer and CNN architecture motivated by VGG16. The Adam Optimizer incorporates the finest attributes of Root Mean Square Propagation (RMSProp) and Adaptive Gradient Descent (AdaGrad) gradient-based optimization algorithm. Rectified Linear Activation Function (ReLU) serves as the activation function for the model's four convolutional layers. A discreet layer, flatten, is also used to enhance the performance. The dataset was gathered from New Dehli, Karnataka, and Maharashtra and categorized into healthy and diseased leaves. Histogram equalization was applied to the dataset to improve the contrast, and images were rescaled and resized. The best weights were saved after the model had been trained to create a GUI allowing the user to recognize sick leaves. The proposed model attained 96.16% accuracy on the mentioned dataset, and it can also be utilized in Internet of Things (IoT) applications, which can aid us in observing the mango leaves.

The research paper[24] explores deep learning techniques for detecting early-stage diseases on plant leaves using Artificial Neural Network (ANN). The research aims to detect small disease blobs that are only discernible with higher-resolution images. The model was used on the 450 mango leaves dataset, and the recommended model performed better than the popular CNN models (AlexNet, VGG16, ResNet50) strengthened with the use of transfer learning. In order to improve the image's quality, various contrast enhancement methods have been utilized. The paper extracted the following features - Statistics-based, Geometric-based, Textural features, and Compressed HSV density. After performing the model, ANN attained 89.41% accuracy, and AlexNet, VGG16, and ResNet50 achieved 78.64%, 79.92%, and 84.88% accuracy, respectively.

This paper [25] used CNN to differentiate five diseased mango leaves - Leaf Webber, Anthracnose, Leaf Gall, Alternaria Leaf Spots, and Leaf Burn from healthy leaves. 1200 images were included in the dataset, and the images were resized to decrease the computational time. The model was trained with 600 images. Various layers of CNN were used to achieve the optimal result, such as the input layer, Convolutional layers 1, 2, 3, and Maxpooling layers 1, 2, 3, and the Output layer. After providing the dataset into the input layer, data augmentation was executed. The proposed model was evaluated with the test dataset, achieving an accuracy of 96.67%. Furthermore, the model had 100% accuracy while classifying Leaf Gall, Leaf Burn, Alternia Leaf Spot, and Leaf Webber because they differed from other classes in terms of their traits.

Saleem et al.[26] proposed segmenting the diseased portion of the leaf using the vein pattern. According to the paper, the vein pattern is the most important part of the leaves. Images were collected in RGB form, and after resizing and augmentation, 135 images of three classes - Powdery Mildew, Sooty Mold, and Healthy- were available. Leaf Vein-Seg Architecture was used to extract the leaf's veins, and Canonical Correlation Analysis (CCA)-based feature reduction was used on the extracted Shape, Color, and Texture features. Cubic support vector machine (SVM) was performed after fusing and extracting, resulting in 96.6% accuracy while classifying Sooty Mold disease and 95.5% accuracy for the Powdery Mildew disease.

In this paper [27], EfficientNetV2, MobileNetV2, and ResNet152V2 were used in 14 different plants to detect 38 types of leaf diseases. The dataset was collected from kaggle, and 43,429 images were used to train the models, while the validation and test set included 5,417, and 5,459 images respectively. EfficientNetV2L achieved the highest accuracy of 99.63%. It attained the same number for F1 score, recall and precision. Later in the paper an XAI technique - LIME was used to understand the decision-making process of the used models, which helped us to comprehend the machine-learning models at a human level. We can visualize which sections of the images influenced the model to predict the outcome that it initially predicted with the help of LIME.
In another paper [28], deep learning classifiers such as Stochastic Gradient Descent(SGD), Support Vector Machine(SVM), and the Hybrid method (SVGD) were used to execute early mango leaf identification since they are essential to maintaining the quality and yield of mango fruit. The models were used to classify mango leaves into three groups: Healthy, Anthracnose, and Black Sooty Mold. The dataset was obtained from Kaggle, and 34% of the photos were used to test the model, while the remaining 66% were utilized to train it. For 5 fold cross validation, SVGD's accuracy was 97.2%; for 20 and 10 fold cross validation, it was 97.8%.

Chapter 3

Methodology

3.1 Background Study

We are classifying and segmenting data from photos of mango leaves in order to identify various diseases. Convolutional Neural Network (CNN), and various models of CNN - MobileNetV2, ResNet50, DenseNet201, and Xception are some of the most well-known picture segmentation and classification systems.

3.1.1 Convolutional Neural Network (CNN)

CNN is a network architecture in deep learning applications that enables trained machines to learn directly from the data. Because it searches for patterns in the pixel data of the images, CNN is highly helpful when attempting to identify items, classes, and categories. It uses matrix multiplication and other linear algebra basics to find image patterns. It can analyze both time series and visual input, which is beneficial in circumstances where object detection is important. CNN was developed on the idea of the human brain. A method similar to the hundreds of billions of brain cells the human brain utilizes to process information is how CNN completes its work. The study used 1.3 billion high-resolution images divided into 1000 different groups [29]. Input layer is the first layer of a typical neural network, where we supply input for our models to prepare the data for processing. The input layer transfers the data from the hidden layer to the output layer. There may be more or fewer hidden layers, depending on the model we are using and the size of the data. The output layer represents the outcome by transforming the final result into a probability score using a logistic function, such as softmax.

In convolution neural networks there are multiple layers such as convolutional layer, polling layer, fully connected layer. Figure 3.1 illustrating how a neural network processes data. This image is a four layer connected CNN model. If we consider this image $n_0 = 3$ input units, $n_1 = 4$ units in the first hidden layer, $n_2 = 2$ units in the second input layer and there are $n_3 = 2$ output units.

If we identify the network's input as x_i where i=1....n_0 and the network's output as y_1 where i=1..., n_3. Then we can write-

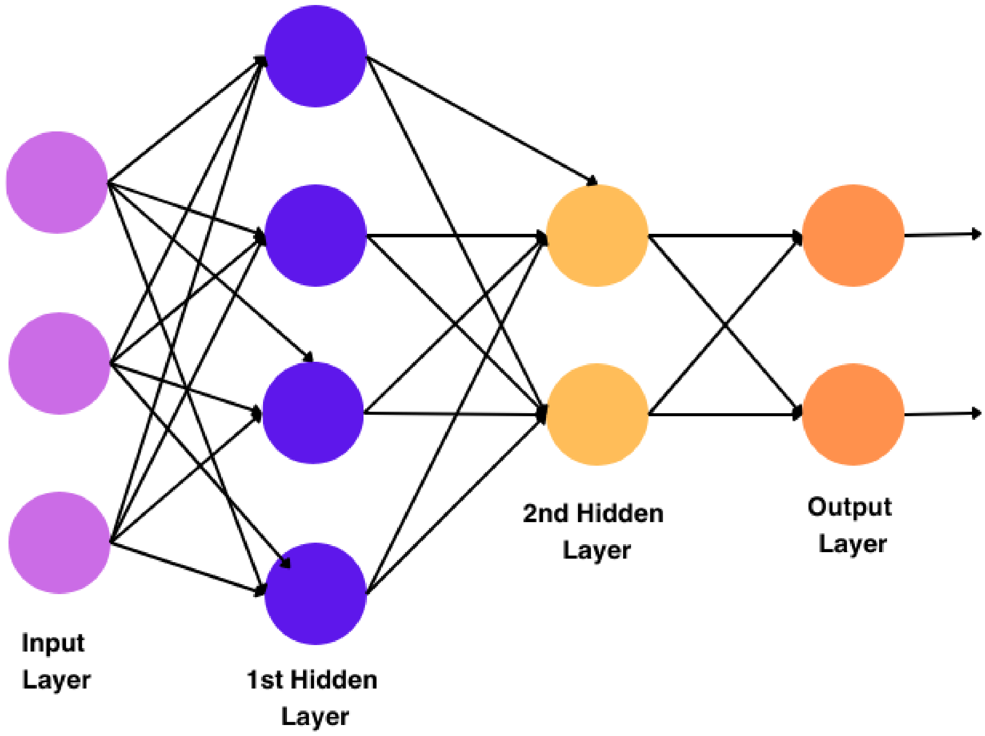


Figure 3.1: Data processing technique of Neural Network

$$h_{1i} = f_1(\sum_{j=1}^{n_0} w^1_{(i)j}\ x_j\ + b_i^{(1)} \quad \text{for i} = 1,...n_1$$

$$h_{2i} = f_2(\sum_{j=1}^{n_1} w^2_{(i)j}\ h_{1j}\ + b_i^{(2)} \quad \text{i} = 1,...n_2$$

$$\hat{y} = f_3(\sum_{j=1}^{n_2} w^3_{(i)j}\ h_{2j}\ + b_i^{(3)} \quad \text{i} = 1,...n_2$$

Here, layer i's activation function is indicated by f_i . The first hidden layer's results are indicated as h_1 for the i-th unit. Likewise, the second hidden layer's outputs are designated as h_{2i}. The specified values for the weights and bias of i in layer k are w_{ij}^k and b_i^k respectively.

These are algebraic equations which construct a neural network.For minimizing the training loop we need a loss function.Depending on various applications we generally use cross entropy or mean squared error. If we consider target variable as y_i, the mean square error loss function will be-

$$L = \sum_{i=1}^{n_3} (y_i - \hat{y})^2$$

3.1.2 Resnet50

One of the most well-known convolutional network topologies is called Resnet50. This innovative neural network was first covered by Xiangyu Zhang, Shaoqing Ren, Kaiming He, and Jian Sun in their computer vision research publication titled "Deep Residual Learning for Image Recognition" from 2015 [30]. It is an architecture where a 50-layer deep neural network network serves as the foundation for several neural networks. ResNet50's capability to operate with more than 150 deep neural networks is its key innovation [31]. ResNet50 uses residual blocks or skip connections to solve the vanishing gradient issue.

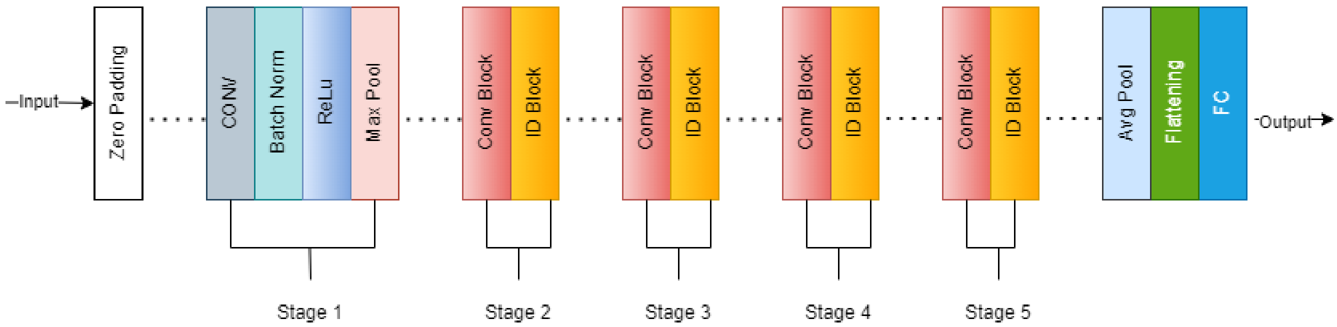


Figure 3.2: ResNet50 architecture

The input of one layer is added to the output of the layer after it, in a residual block. This is represented as F(x)+x, where F(x) is the transformation applied by the layer. The goal is to make the residual F(x) close to zero, enabling the network to concentrate on understanding how the current layer differs from the layers which came before it. ResNet50 trains on the residuals F(x) rather than the final output (Y). The objective is to make F(x) close to zero, implying that Y should be close to X. To equalize input sizes in the skip connection, input volume padding and 1x1 convolutions are utilized.

$$[\frac{n+2p-f}{s+1}]^2$$

where n = size of the input image, p = padding, s = stride, and f = number of filters.
ResNet50's last layer is modified to correspond with the number of classes in the multiclass classification task. For probability distribution across classes, the output layer commonly employs a softmax activation function. Categorical Crossentropy loss is commonly used for multiclass classification [32]. The model is trained using back propagation and optimization algorithms, adjusting weights to minimize the defined loss function. By facilitating effective training of deep networks, skip connections allow the model to identify complex patterns in the data. The model is validated on a separate dataset to ensure generalization. Testing involves evaluating the model on unseen data to assess its performance. ResNet50's skip connections and residual learning enable the training of very deep neural networks, making it effective for complex tasks such as multiclass image classification. The model adapts to the specifics of the classification problem through adjustments in the output layer and loss function.

3.1.3 Densenet201

Densenet201 is one of the newest additions to the invasions for visual object recognition. Full form of Densenet is Densely Connected Convolutional Networks [33]. It was suggested by the third iteration of densely linked convolutional networks (DenseNet), a CNN architecture created in collaboration with Facebook AI Research, Tsinghua University, and Cornwell University and published in 2017 CVPR [34]. DenseNet201is a type of convolutional neural network and it has 201 deep

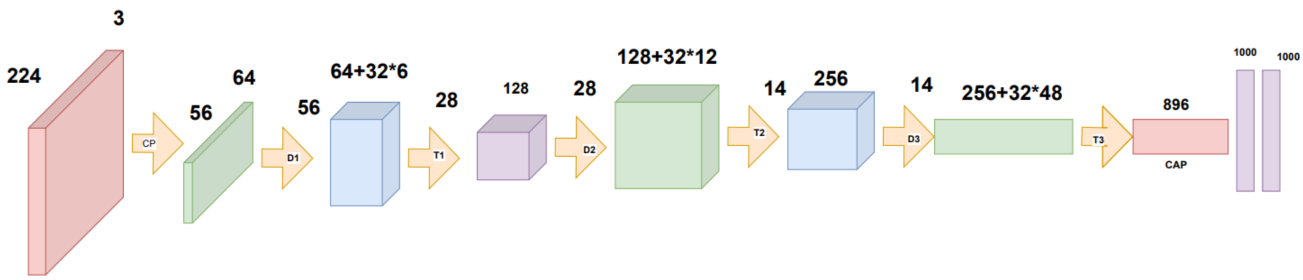


Figure 3.3: DenseNet201 architecture

neural layers the parameter range is 20,242,984. Every conventional convolutional network has n layers and n connections. However, with DenseNet, every layer is linked to every other layer. DenseNet has n(n+1)/2 total connections. Every layer in DenseNet201 obtains more information from the other levels [33]. Different sorts of DenseNet201 blocks apply to different datasets. A few of them are:

- Basic DenseNet Composition Layer: A 3*3 convolutional layer, a ReLU activation function, and a preactivated batch normalization layer connect each of the n layers in this layer.
- BottleNeck DenseNet (DenseNet-B): Every level's calculation time will grow since each layer generates K output. A bottleneck technique is used to tackle this issue, using a 1*1 convolution layer before every 3*3 convolution layer.

DenseNet uses a fictitious patient set to calculate the value for each decision criterion. The discounted value in case of a one-time expense v produced by an event that occurs at time t is,

$$v_{disc}(v, t, \gamma) = \frac{1}{(1+\gamma)^t} v,$$

The discount rate is represented by γ. where γ is the discount rate. The resulting value from applying the corresponding criterion to a cumulative payout of value v along the interval (t_1, t_2) is,

$$accuredValue_u(v, t_1, t_2, \gamma) = \int_{t_1}^{t_2} v_{disc}(v, t, \gamma)dt = v * \frac{1}{ln(1+y)} * [\frac{1}{(1+\gamma)^{t_1}} - \frac{1}{(1+\gamma)^{t_2}}],$$

If we clarify,

$$\delta = ln(1+\gamma),$$

These formulas can be represented as,

$$v_{disc}(v, t, \gamma) = e^{-\delta t} v,$$

and

$$accuredValue_u(v, t_1, t_2, \gamma) = \int_{t_1}^{-t_2} v_{disc}(v, t, \gamma) dt = \int_{t_1}^{t_2} ve^{-\delta t} dt = v\frac{e^{-\delta t_1} - e^{-\delta t_2}}{\delta}$$

which are sometimes used in the literature.
DenseNet201 can train the tiny connections placed close to input and output and operates on the foundation of convolutional networks with deeper, more accurate layers. DenseNet201 has the benefit of using dense block and transition layer to reduce shift overfitting and improve image processing accuracy. Every layer is linked to DenseNet201 using a forward feed topology. It reduces the number of parameters, strengthens feature propagation, and resolves the vanishing gradient issue.

3.1.4 MobileNetV2

MobilenetV2 is a refined version of mobilenetv1 because it relies on the depth-wise separable convolutional block used in the first version. There is a new block in this version called inverted residuals with linear bottlenecks [35]. These blocks consist of lightweight convolutions followed by a linear pointwise convolution. This version decreases computational time and complexity while maintaining its expressive power.

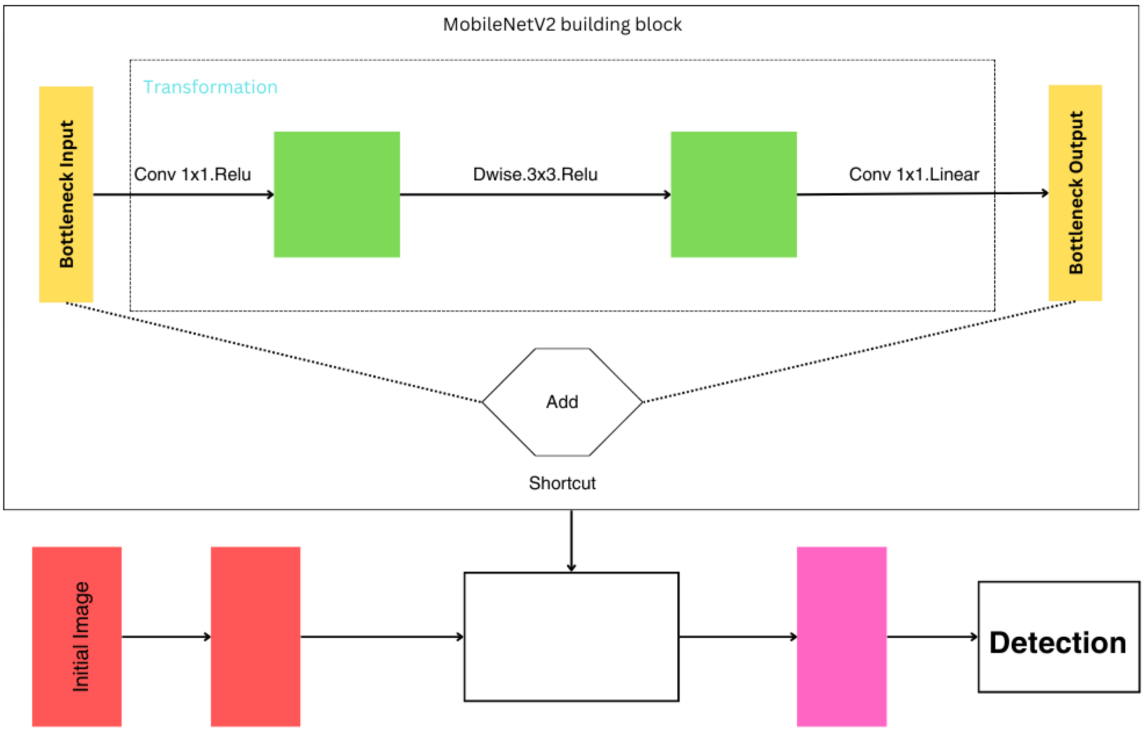


Figure 3.4: MobileNetV2 architecture

This architecture consists of a lightweight depth-wise convolution, a linear pointwise convolution (1 x 1 convolution) and an element-wise addition of the input to the output, also known as residual connection [35]. The depth-wise convolution is represented as-

$$y = DConv(x, W_{depthwise})$$

Here, y refers to the output feature map, x refers to the feature map and W(depthwise) refers to the depth-wise convolutional filters. As for pointwise convolutional, the formula is the same except W(pointwise) refers to the pointwise convolutional filters. Also, this architecture uses skip connections, which helps gradients to propagate better during training. The skip connection formula is-

$$y = x + F(x)$$

In this case, x is the layer's input, y is the output, and the transformation performed to x is F(x).
Mobilenetv2 is used for various image classification problems. Though it is efficient by reducing the model size and computational complexity, sometimes it does not give us top-tier accuracies like ResNet, Inception and such.

3.1.5 Xception

Xception stands for "extreme inception", which depends on two key factors -
-Depthwise Separable Convolution
-Shortcuts between Convolution blocks
If we examine convolution, we find that for a single kernel the equation is,

$$K^2.d^2.C$$

Here, C is the channel size, d is the filter size, and K is the resulting dimension after the convolution. For N kernels,

$$K^2.d^2.C.N$$

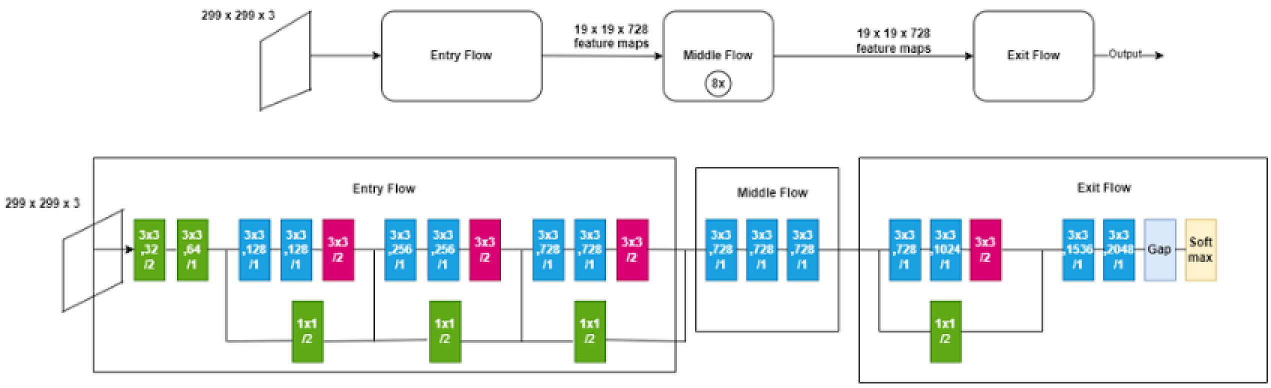


Figure 3.5: Xception architecture

Overall it is an expensive process. To overcome this, Xception architecture uses depthwise separable convolution, which relies on Depthwise Convolution and Pointwise Convolution. In depthwise convolution, instead of computing over the whole channel, we calculate one channel at a time[36]. If the kernel size is d*d, it becomes d*d*1 instead of d*d*C. By doing this, we create a K*K*C volume and then we apply the Pointwise Convolution. We perform a conventional convolution of size 1*1*N across the K*K*C volume, yielding a K*K*N shaped output. We were able to decrease the number of operations by a factor proportionate to $\frac{1}{N}$ by doing this. The overall number of convolutional layers in the Xception model is 36. The data enters the exit flow after going through the entry flow and then the middle flow, which is repeated eight times. [37]. Compared to InceptionV3, Resnet, and VGG-16, Xception performs better overall.

3.1.6 Grad-CAM

Grad-CAM is a deep learning technology that predicts and visualizes the most crucial areas of an image that are essential to the prediction of a neural network. It computes the target class score gradients by using feature maps from the final convolutional layers. It creates a heat-map that shows the areas that the model considers important when making its decision. In order to construct a Grad-CAM of height v and width u for any class c, we first compute $\frac{\delta yc}{\delta Ak}$, which is the gradient of the class c score, yc (before to the softmax), with respect to feature mappings Ak of a convolutional layer. Once the gradients are obtained, the following equation is used to emphasize the significance of each feature map k for certain classes, with the use of the global average pooling approach.

$$a_c^k = 1/z \sum_i \sum_j \frac{\delta Y^c}{\delta A_{ij}^k} \tag{3.1}$$

In this case, gradients over back-propagation are referred to by partial differentials, and the global average pooling is referred to by the summing over i and j. Afterwards, ReLU is executed after a weighted mixture of forward activation maps.

$$L_{GRAD-CAM}^c = RELU \sum_k a_k^c A^k \tag{3.2}$$

ReLU is the best option in this instance since it draws attention to the characteristics that are positively affecting the class of interest. In the absence of it, it has been observed that the localization performance may be impacted, meaning that occasionally the localization maps contain more information than the intended class and it may belong to other categories in the image. For a given class c, the class score is calculated as

$$S^c = RELU \sum_i \sum_j \sum_k w_k^c A_{ij}^k \tag{3.3}$$

Grad-CAM can be useful in localized explanations, model prediction, and other areas, but it can also lead to incorrect heat-map localization, the inability to localize numerous instances of an item in an image, and other issues.

3.1.7 LIME

Lime(local interpretable model-agnostic explanations), is an approach that provides a local interpretable model to approximate any black box machine learning model

and explain each unique prediction. It's a method that provides a roughly interpretable model for the classifier and regressor.
It is a technique created by Marco Ribeiro in 2016 to explain machine learning models' predictions. It works on the sample data features and values by tweaking the points. This model basically explains each sample data points. The model output represents each sample points in a form of set which is locally interpretable. LIME is model-agnostic it means we can apply it in any ML projects. LIME works with sample data inputs which is an interpretable representation that humans can easily Understand. The general formula that provides the explanation generated by LIME at a local position x is as follows:

$$\xi(x) = \underset{g \in G}{\arg\min} \left[L(f, g, \pi_x) + \Omega(g) \right] \tag{3.4}$$

When x denotes the locality, f is our true function (also known as ground truth), and g is a dummy function we employ to approximate f in the vicinity of x. We get a list of reflections and clarification for each aspect of the sample data points from LIME's outputs. A set of explanations that show how each attribute affected the prediction of a sample of data is the result of LIME. Lime gives us the chance to ascertain which element, out of all the ones that would affect the forecast the most. In order to create synthetic data that is assessed by the black box system and utilized as a training set for the glassbox model, LIME perturbs each particular datapoint. [38]. Lime is a modern machine learning classifier which is a model-agnostic easy to understand and model can implement with less effort.For humans to use the model it's important that the model is easily explainable to its users.It in case Lime help us by uncover the ideas behind black box and all the necessary answers behind AI-models predictions and classifies.

3.2 Dataset

In our research,the MangoLeafBD Dataset [39] presents a comprehensive collection of 4000 mango leaf images, each with a resolution of 240x320 pixels and stored in JPG format. Within this dataset, approximately 1800 images showcase distinct mango leaves, while the remaining images are skillfully prepared by applying zooming and rotation techniques where necessary. The dataset encompasses a diverse set of seven diseases affecting mango leaves, including Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, and Sooty Mould.
It is organized into eight classes, incorporating a category for healthy leaves. Each class comprises 500 images as shown in fig. 3.7, providing a balanced distribution for robust model training and evaluation. The data were captured using mobile phone cameras in four mango orchards across Bangladesh, specifically Sher-e-Bangla Agricultural University orchard, Jahangir Nagar University orchard, Udaypur village mango orchard, and Itakhola village mango orchard. One sample images from each classes of our collected datasets is shown in fig. 3.6.
In the preprocessing phase for the mango leaf image dataset, a multi-step approach was employed to ensure the data is appropriately formatted for training a convolutional neural network (CNN) based on Xception,MobileNetV2,ResNet50,DenseNet201 architectures. Initially, the OpenCV library was utilized to load and resize each image to a standardized dimension of 224x224 pixels and we re-scaled the pixel range

Figure 3.6: Dataset Sample

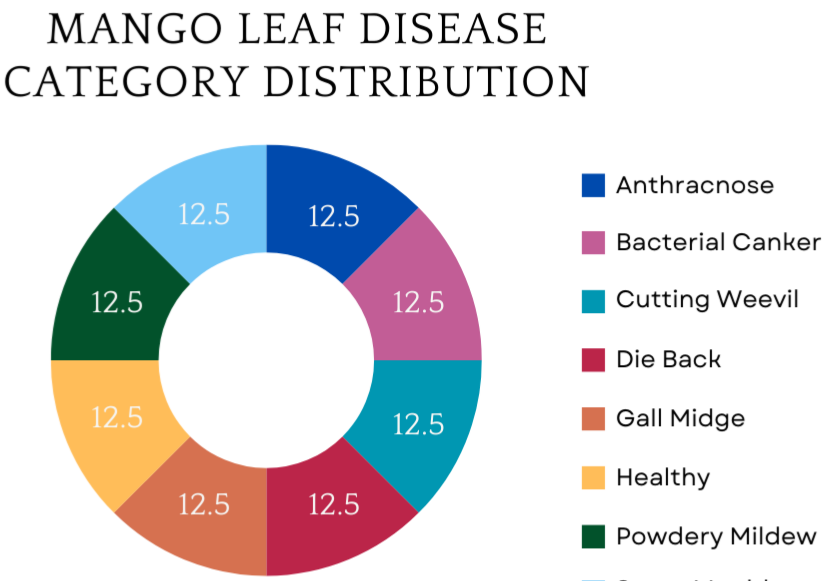


Figure 3.7: Dataset Distribution

from 0 to 1 and set the zoom_range to 0.5, which means the images may be arbitrarily zoomed in or out by up to 50% during data augmentation. Simultaneously, labels corresponding to the folder names of the images were encoded into numerical format using scikit-learn's LabelEncoder. The encoded labels were further converted into categorical format through Keras' to_categorical function, facilitating their compatibility with the neural network model. Subsequently, the dataset underwent a train-test split using scikit-learn's train_test_split method, where 80% of the data was allocated for training the model, and the remaining 20% was reserved for evaluating its performance. This split was stratified based on the original labels to ensure a balanced distribution of classes in both the training and testing sets. The resulting preprocessed dataset was then fed into Xception,MobileNet,ResNet50,DenseNet201 models for training, forming a crucial foundation for the subsequent classification tasks.

3.3 Research Workflow

The research workflow for this paper is presented in fig. 3.8. We will be using several CNN machine-learning algorithm models in this study with which we can detect and classify mango leaf diseases with a high accuracy level. To quickly extract features from the picture of a mango leaf, we will utilise each of the four models (ResNet50, DenseNet201, MobileNetV2, Xception) individually. Additionally, we'll retrain them using our dataset as necessary to ensure that it achieves our goal of a high degree of accuracy. The results of these models will be compared, and we will check for the accuracy and disadvantages which come with each of these models. To comprehend the mentioned models deeply and in a way that is understandable to human we will apply XAI, specifically LIME and Grad-CAM. It will help us to recognize how each model behaves in order to make the prediction.

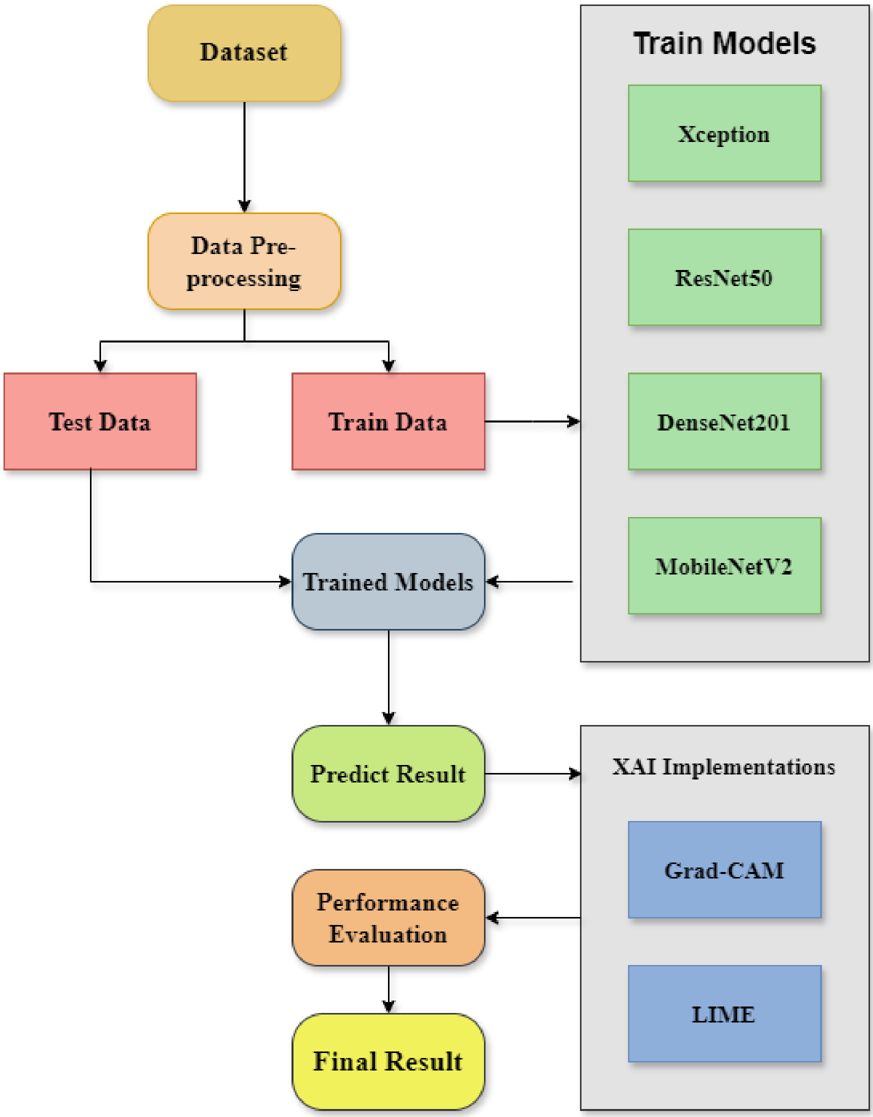


Figure 3.8: Workflow Diagram

Chapter 4

Result & Discussion

4.1 Model Implementation

The implementation of the proposed model for the dataset is defined in this section. By using Google Colaboratory, we introduced our models for training and testing. The model implementation consists of a few steps such as - preprocessing the input data, fitting our models, classification and checking the output result. In the preprocessing phrase we take necessary steps to qualify our mango leaf diseases dataset which is used as an input of the models.
Xception, MobileNetV2, ResNet50, and DenseNet201 are the four CNN models that we implement to categorize the input photos and determine the accuracy rate of the output.
The outcome of using the suggested method to identify mango leaf disease is also included in this chapter. To give the output result, we used the test data from our dataset as input and gave the output result.

4.2 Workflow overview

To get the best outcome by using these models on our proposed model Mango leaf diseases detection we followed some necessary steps. The overview of the research paper is shown below-

- Initially, we resized the Mango leaf images to 224x224 amd re-scaled the pixel range from 0 to 1.
- After that, we split the dataset in a propotation of 80:20 to train and test data.
- We trained our propose models - Xception,MobileNetV2,ResNet50,DenseNet201 and validate them.
- At the end, we calculated the accuracy score of our models using the test data and applied XAI to solve the "black box" issue.

4.3 Performance Evaluation

When assessing the predictions- performance, precision, recall and confusion matrix are crucial metrics. These metrics measure how well the result fit and from the given positive results we can deduct how many of them are actually false positive and how many true positive our models have missed. Confusion matrix also evaluates the performance of machine learning models. It represents the relationship between predicted values and actual values.

4.3.1 Precision

Precision is a critical indicator of positive predictive accuracy. It is the fraction of relevant results among the retrieved results The formula used to calculate precision:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

4.3.2 Recall

The accuracy of true positives among the relevant elements. It is calculated using the equation below:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

4.3.3 F1 Score

We measure the model's accuracy using F1 Score. It takes into account both precision and recall and calculates the performance metric of the proposed performs on the given dataset. The equation is:

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

4.3.4 Confusion Matrix

Confusion matrix is a NxN matrix. For binary classification, a 2x2 confusion matrix looks like -

$$\begin{bmatrix} TP & TN \\ FP & FN \end{bmatrix}$$

Since our problem is a classification problem and we have 8 classes, we can understand the accuracy of the class detected by our models compared to the actual class the image belong to from confusion matrix.

4.3.5 Accuracy

Accuracy indicates the general correctness of our model. To calculate the accuracy we need to divide the correct prediction by the total prediction made by the models.

$$Accuracy = \frac{CorrectPrediction}{TotalPrediction}$$

4.4 Result & Discussion

Table 4.1: Precision, Recall, F1 Score, Accuracy of the models

Models	Precision	Recall	F1 Score	Accuracy
Xception	78%	76%	75%	76%
MobileNetV2	91%	90%	90%	90%
DenseNet201	95%	95%	95%	95%
ResNet50	98%	98%	98%	98%

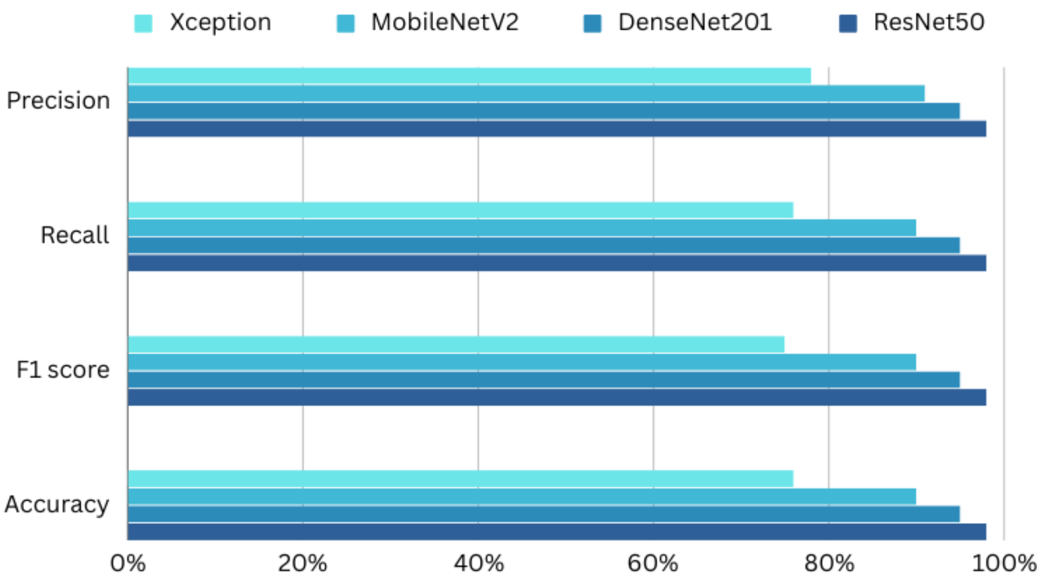


Figure 4.1: Model Comparison

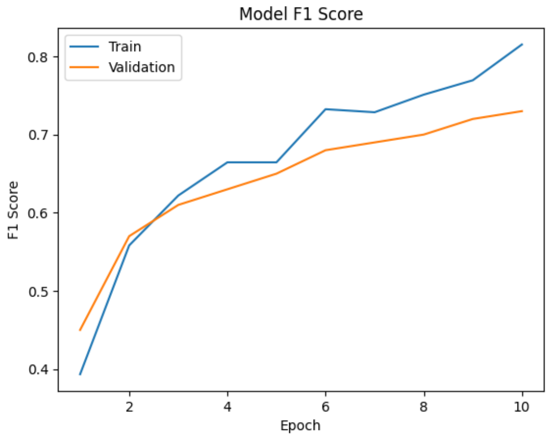
In this research paper we used four CNN models - Xception, MobileNetV2, DenseNet201, ResNet50 on our dataset. For all four models, we used the pre-trained models on imagenet. Our models' overall precision, recall, F1 score and accuracy are shown in table 4.1. We trained the models on our train dataset for 10 epochs and we've taken the weighted average values of these models after claculating them for each class. These values give us an overall idea about our model's performance and from the result, we can see, ResNet50 peformed better than the other models with 98% accuracy and 98% F1 score. Fig. 4.1 represents the different score for each models.

4.4.1 Xception

Table 4.2: Precision, Recall, F1 Score, Accuracy of Xception

Precision	Recall	F1 Score	Accuracy
78%	76%	75%	76%

(a) Accuracy of the Xception model



(b) F1 Score of the Xception model

Figure 4.2: Accuracy and F1 Score for the Xception Model

The model achieved 76% accuracy and 75% F1 score with a score of 78% and 76% for precision and recall respectively as shown in table 4.2. While fig. 4.2a and fig. 4.2b shows the model accuracy and F1 score respectively and fig. 4.3 displays the confusion matrix. We used the pre-trained Xception model for our mango leaf disease detection with a learning rate of 0.1 and froze all the layers. GlobalAveragePooling2D and Adam optimizer were employed in this model. Additionally, a Dense layer with eight units and softmax activation to the final layer were added to achieve the mentioned result

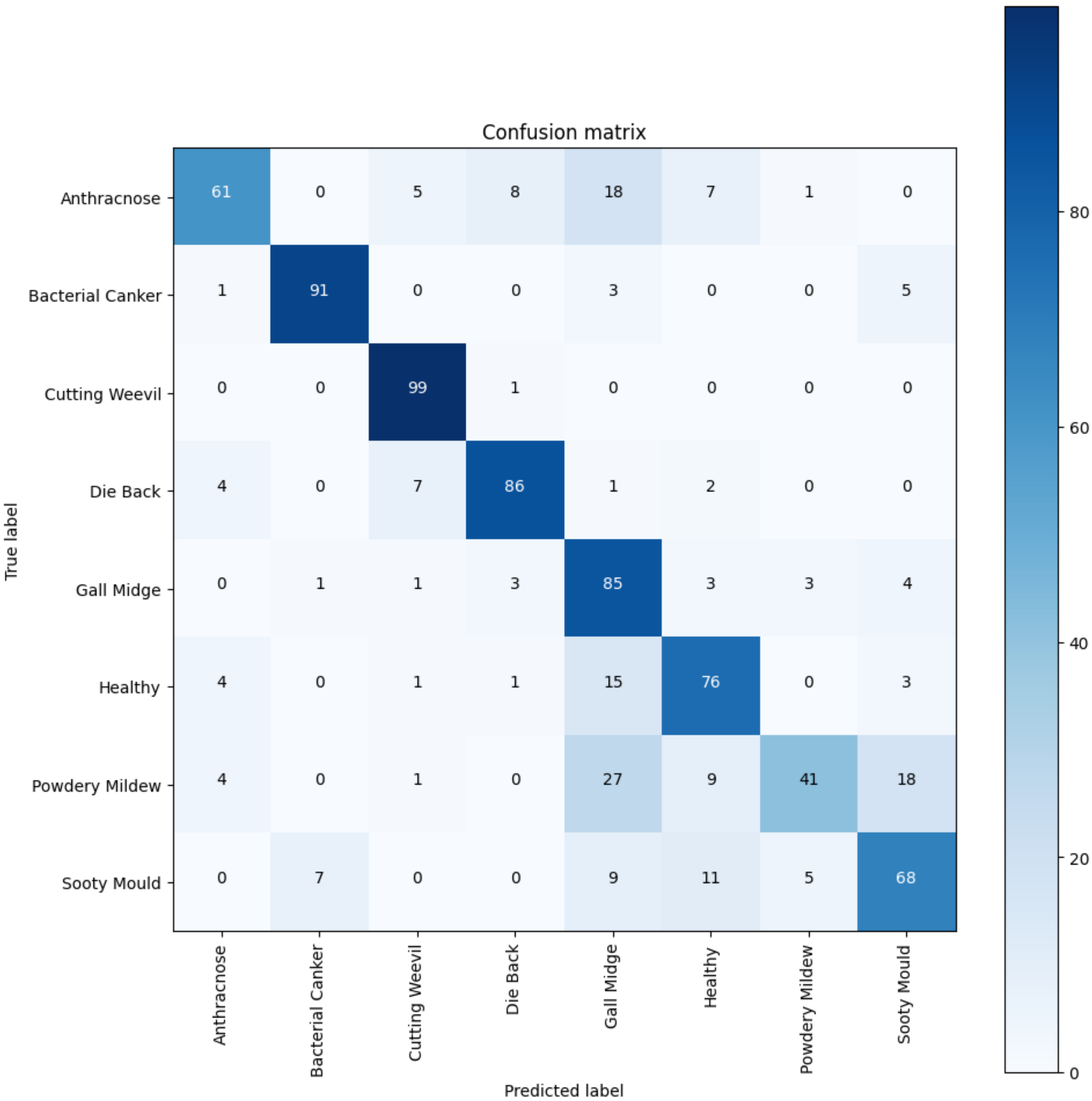
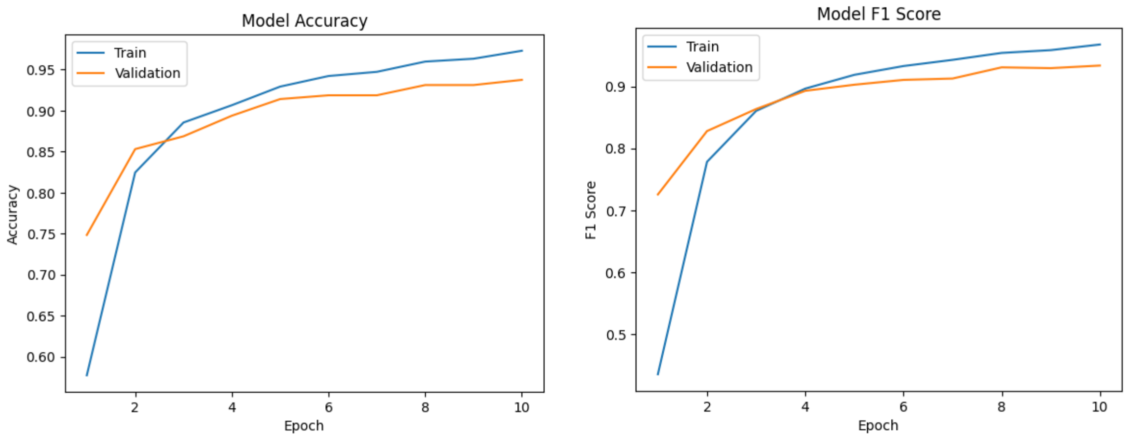


Figure 4.3: Xception Confusion Matrix

4.4.2 MobileNetV2

Table 4.3: Precision, Recall, F1 Score, Accuracy of MobileNetV2

Precision	Recall	F1 Score	Accuracy
91%	90%	90%	90%



(a) Accuracy of the MobileNetV2 model (b) F1 Score of the MobileNetV2 model

Figure 4.4: Accuracy and F1 Score for the MobileNetV2 Model

As the table 4.3 illustrates, the model produced a satisfactory performance with 90% precision, recall, accuracy, and F1 score. A model accuracy graph(fig. 4.4a), a F1 score graph(fig. 4.4b) and confusion matrix(fig. 4.5) are also provided. Similar to Xception, we froze every layer in the pre-trained model with a learning rate of 0.1. GlobalAveragePooling2D and the Adam optimizer were additionally utilized in this instance. A Dense layer of eight units and softmax activation were added to the final layer.

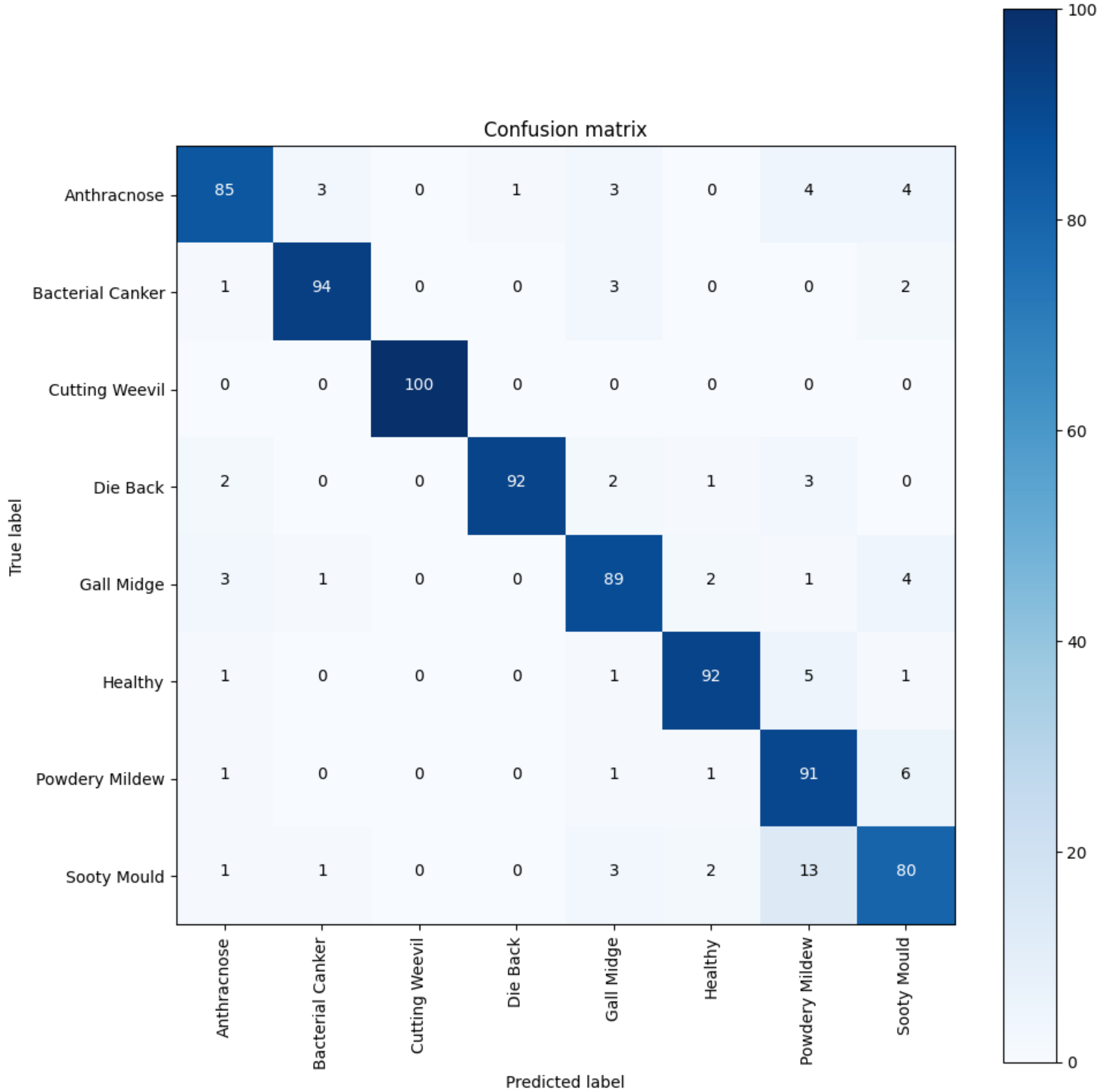
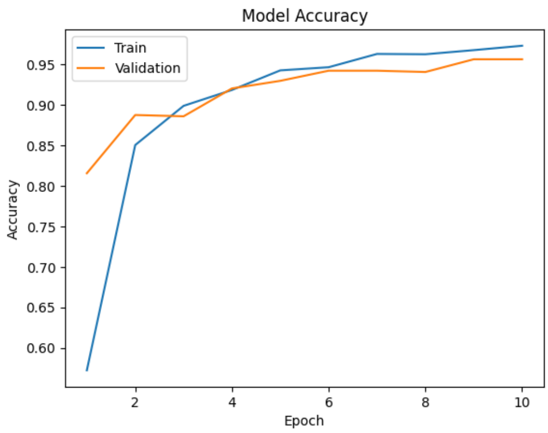


Figure 4.5: MobileNetV2 Confusion Matrix

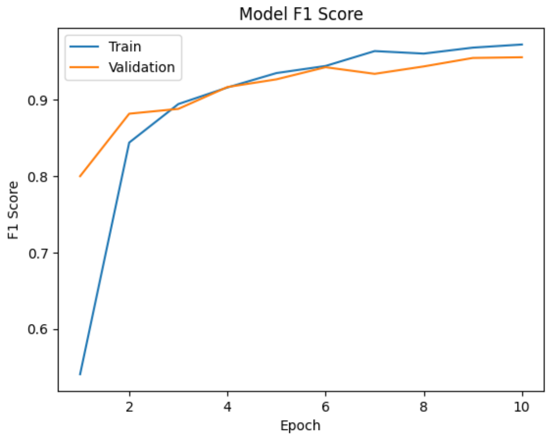
4.4.3 DenseNet201

Table 4.4: Precision, Recall, F1 Score, Accuracy of DenseNet201

Precision	Recall	F1 Score	Accuracy
95%	95%	95%	95%



(a) Accuracy of the DenseNet201 model



(b) F1 Score of the DenseNet201 model

Figure 4.6: Accuracy and F1 Score for the DenseNet201 Model

Precision, recall, F1 score and accuracy for this model are given in table 4.4. It obtained 95% precision, recall, accuracy and F1 score. Model accuracy graph(fig. 4.6a), F1 score graph(fig.4.6b) and confusion matrix(fig. 4.7) are included for better understanding. We used 0.1 learning rate while implementing the pre-trained DenseNet201, with all layers frozen. Adam optimizer and GlobalAveragePooling2D were also utilized for improved performance. To utilize the model for our uses the last layer was modified to a Dense layer of 8 units.

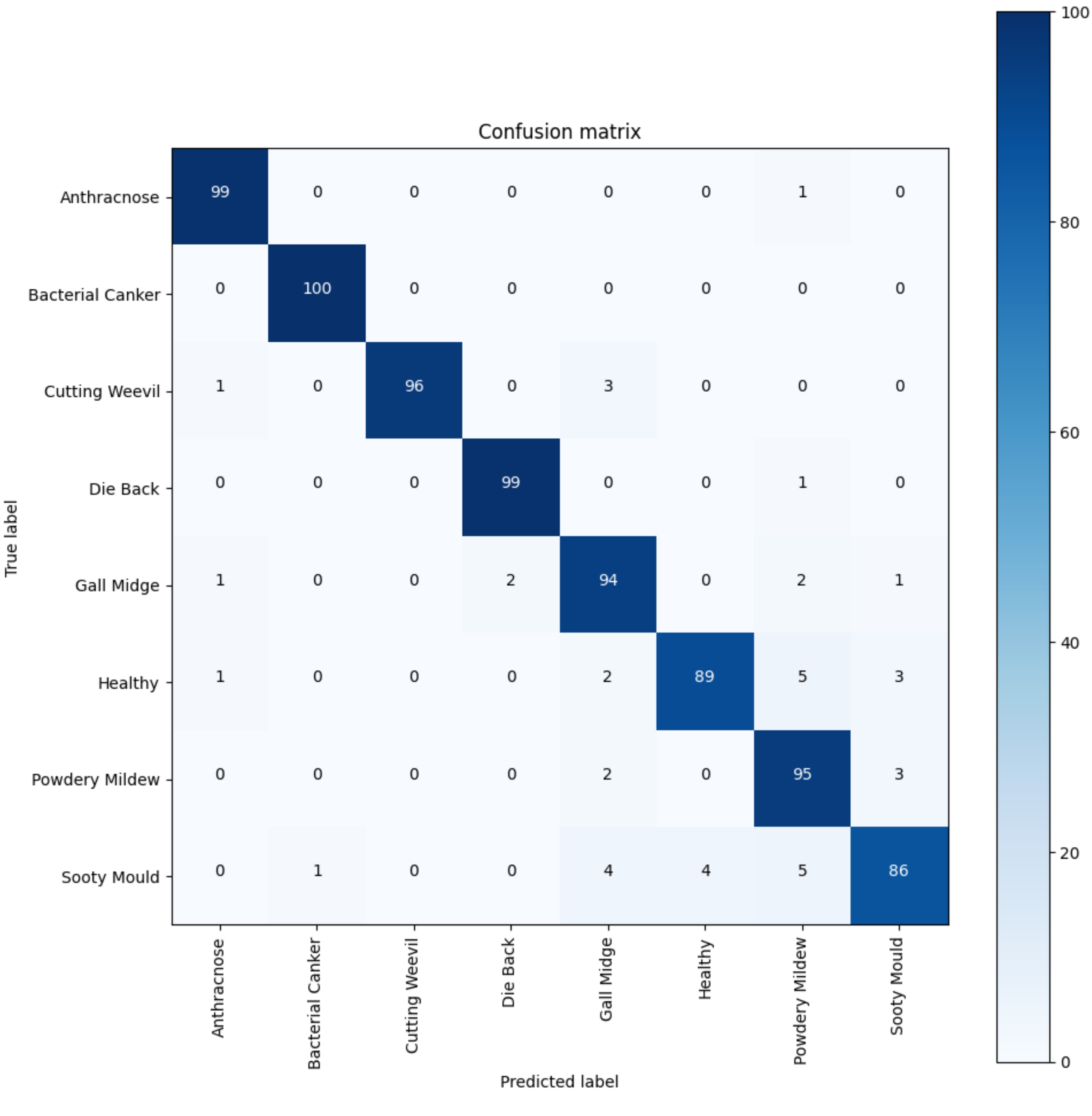
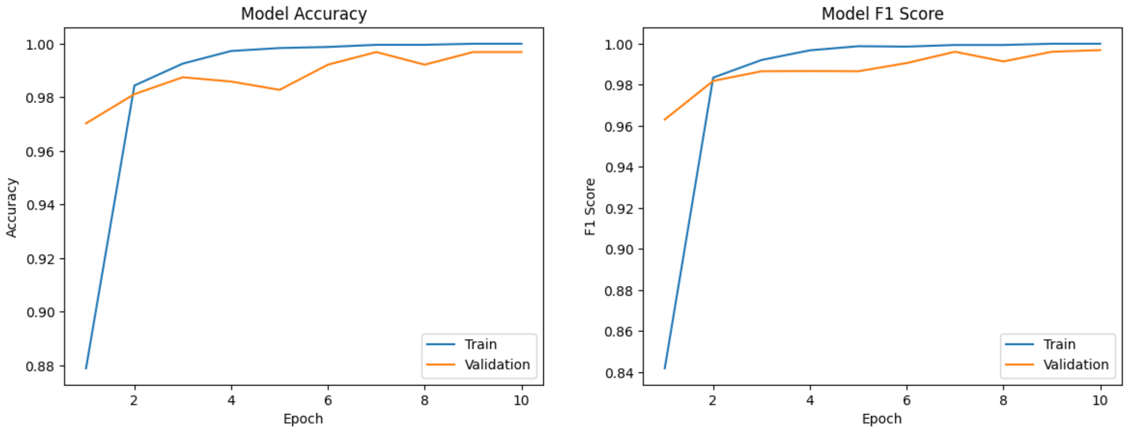


Figure 4.7: DenseNet201 Confusion Matrix

4.4.4 ResNet50

Table 4.5: Precision, Recall, F1 Score, Accuracy of ResNet50

Precision	Recall	F1 Score	Accuracy
98%	98%	98%	98%



(a) Accuracy of the ResNet50 model (b) F1 Score of the ResNet50 model

Figure 4.8: Accuracy and F1 Score for the ResNet50 Model

ResNet50 acquire 98% Precision, recall, F1 score and accuracy which are provided in table 4.5. Fig. 4.8a displays Model accuracy graph, fig. 4.8b displays F1 score graph and fig. 4.9 displays confusion matrix, which gives us an overview of our model. Pre-trained ResNet50 was used with a learning rate of 0.1 and all the layers were frozen for our classification problem. For improved result, Adam optimizer and GlobalAveragePooling2D were also deployed. The last layer was altered to a Dense layer of 8 units to utilize the model for our uses.

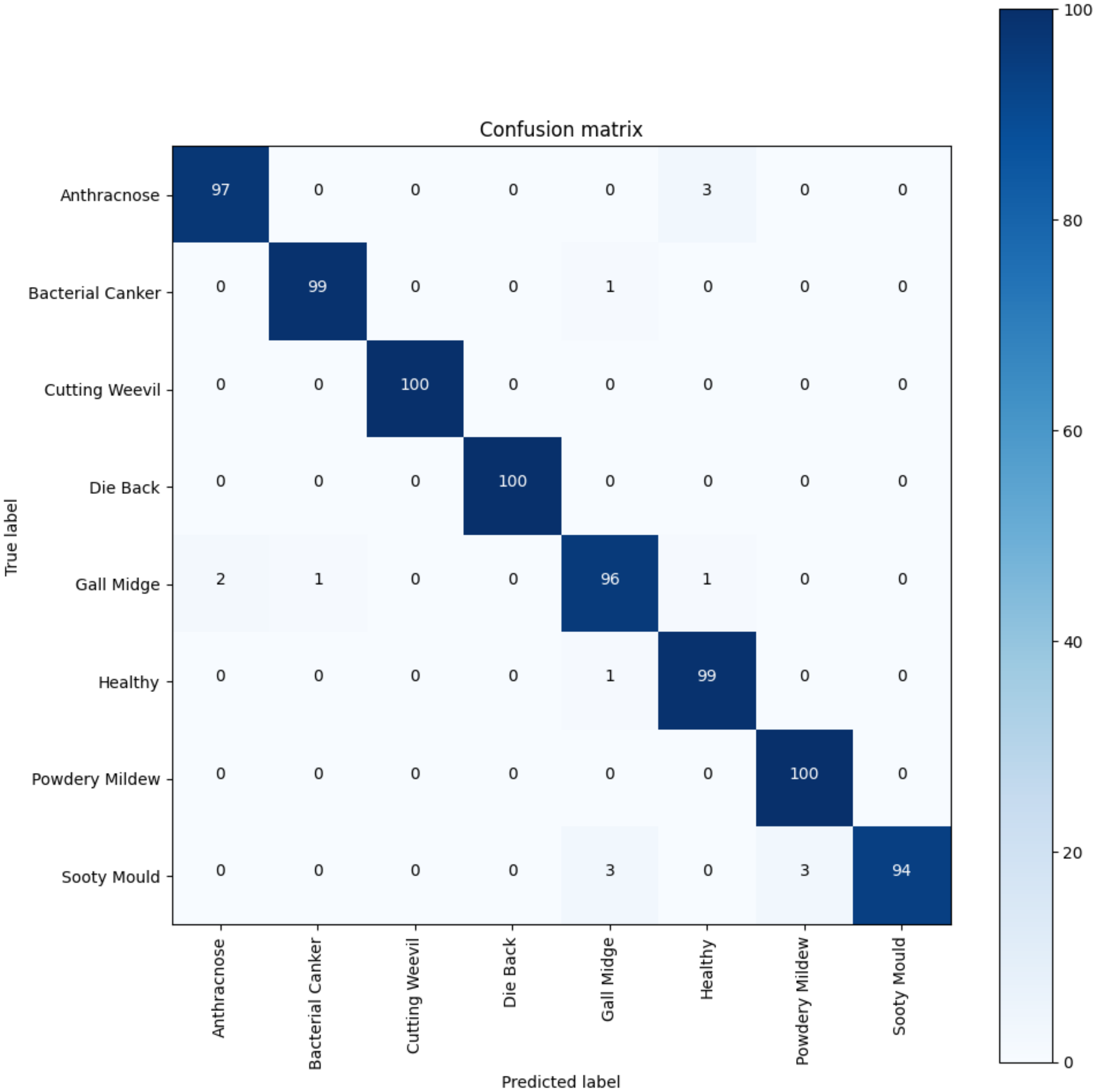


Figure 4.9: ResNet50 Confusion Matrix

4.4.5 Grad-Cam

Xception

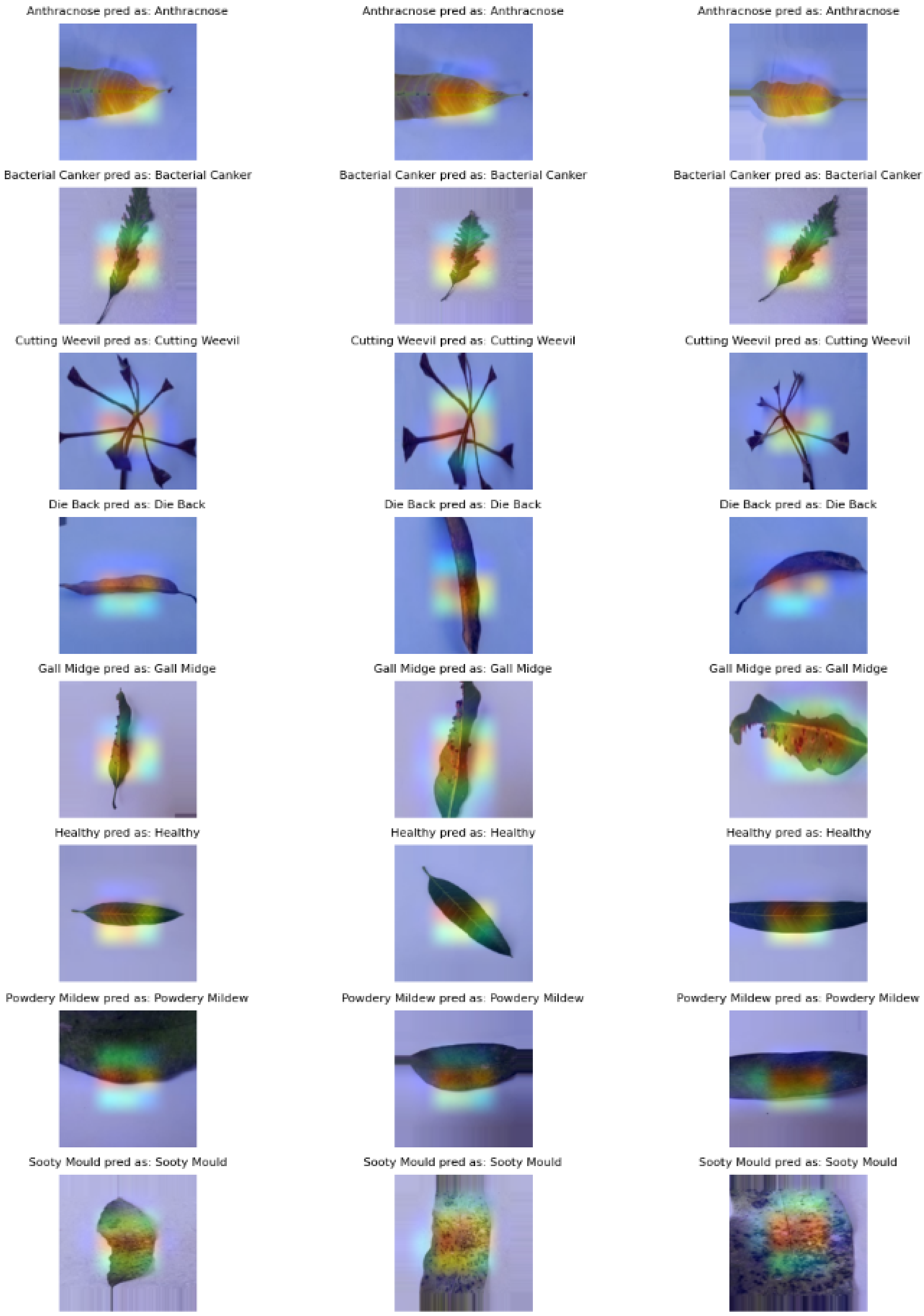


Figure 4.10: Grad-Cam applied on Xception model

MobileNetV2

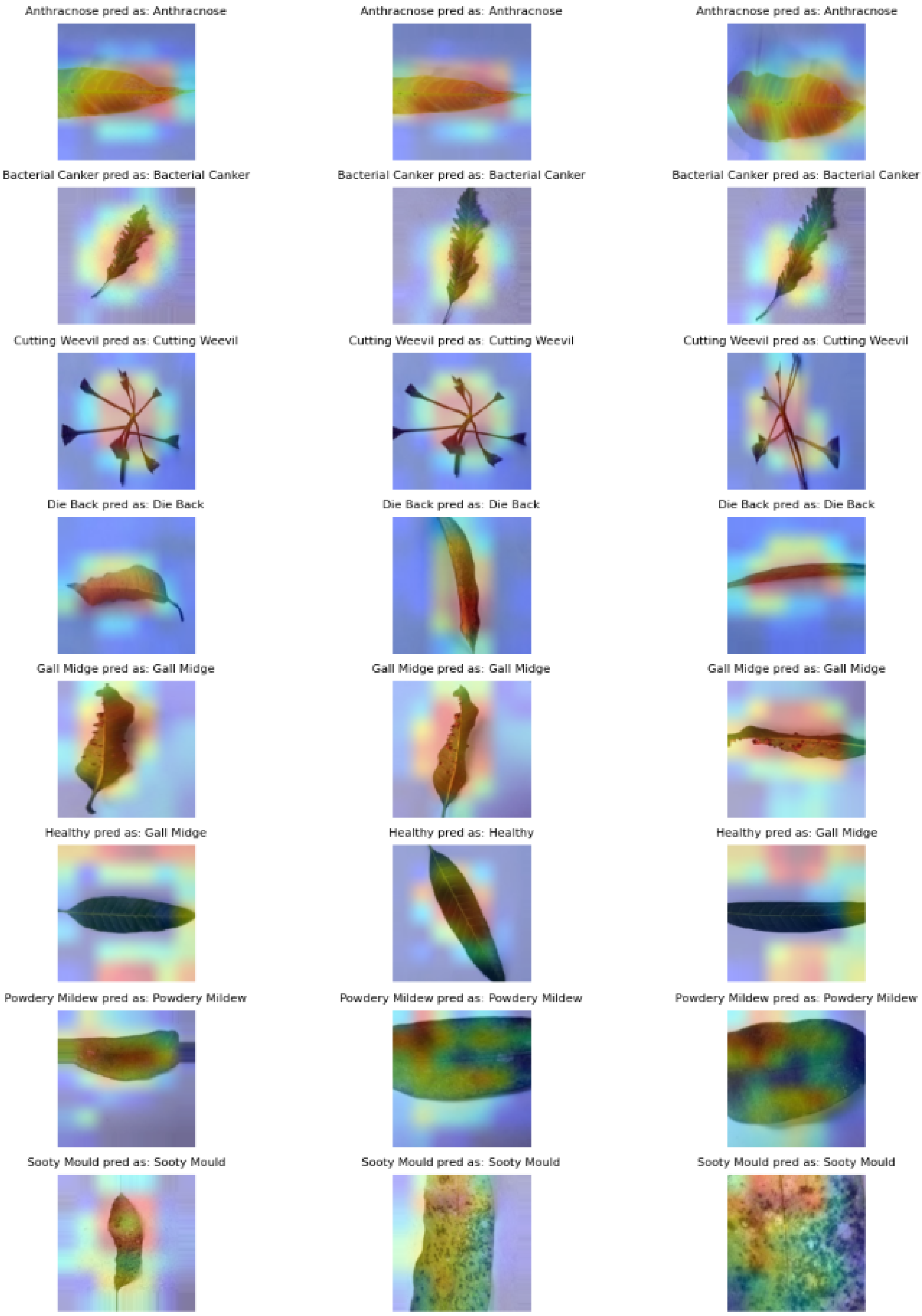


Figure 4.11: Grad-Cam applied on MobileNetV2 model

DenseNet201

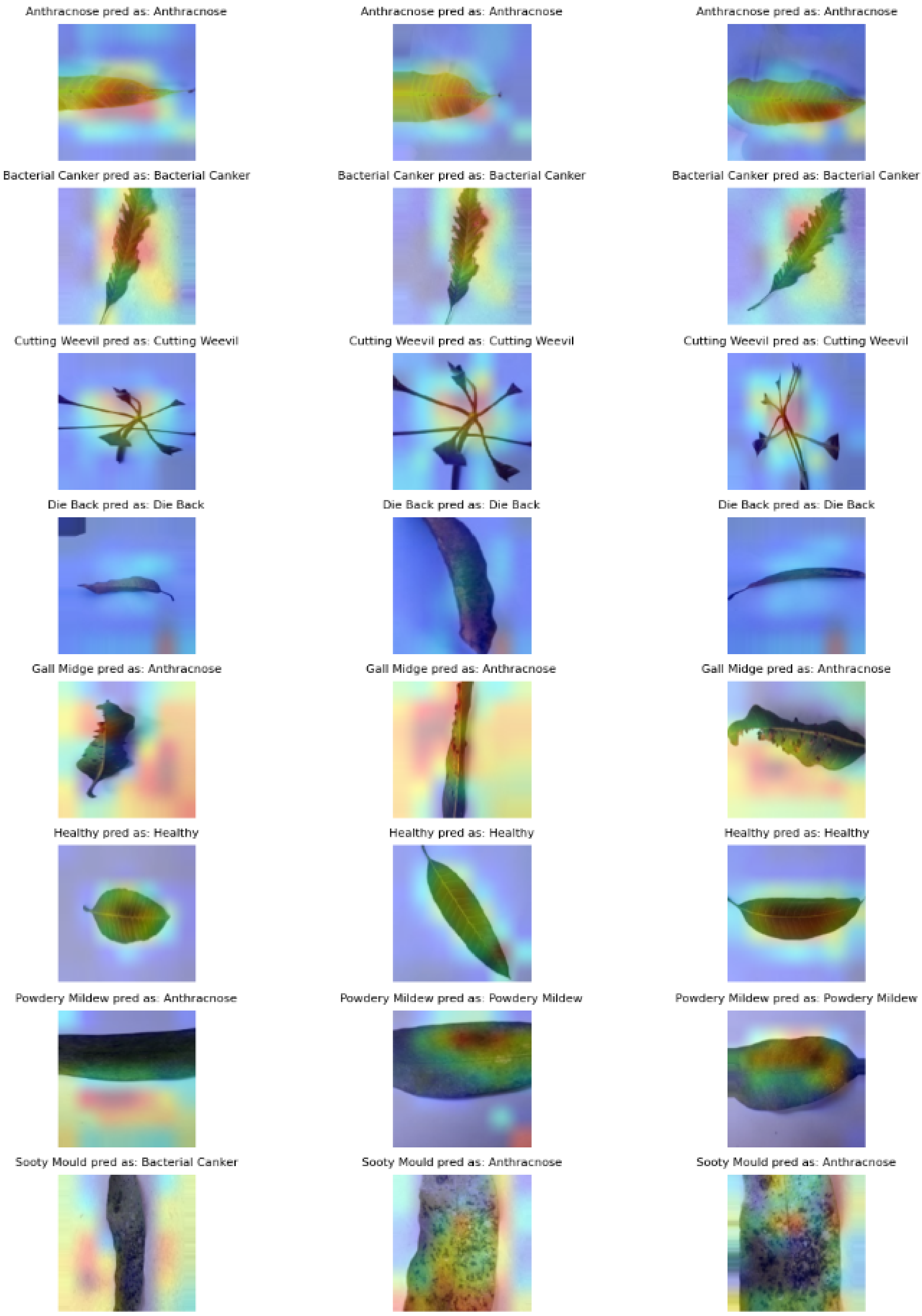


Figure 4.12: Grad-Cam applied on DenseNet201 model

ResNet50

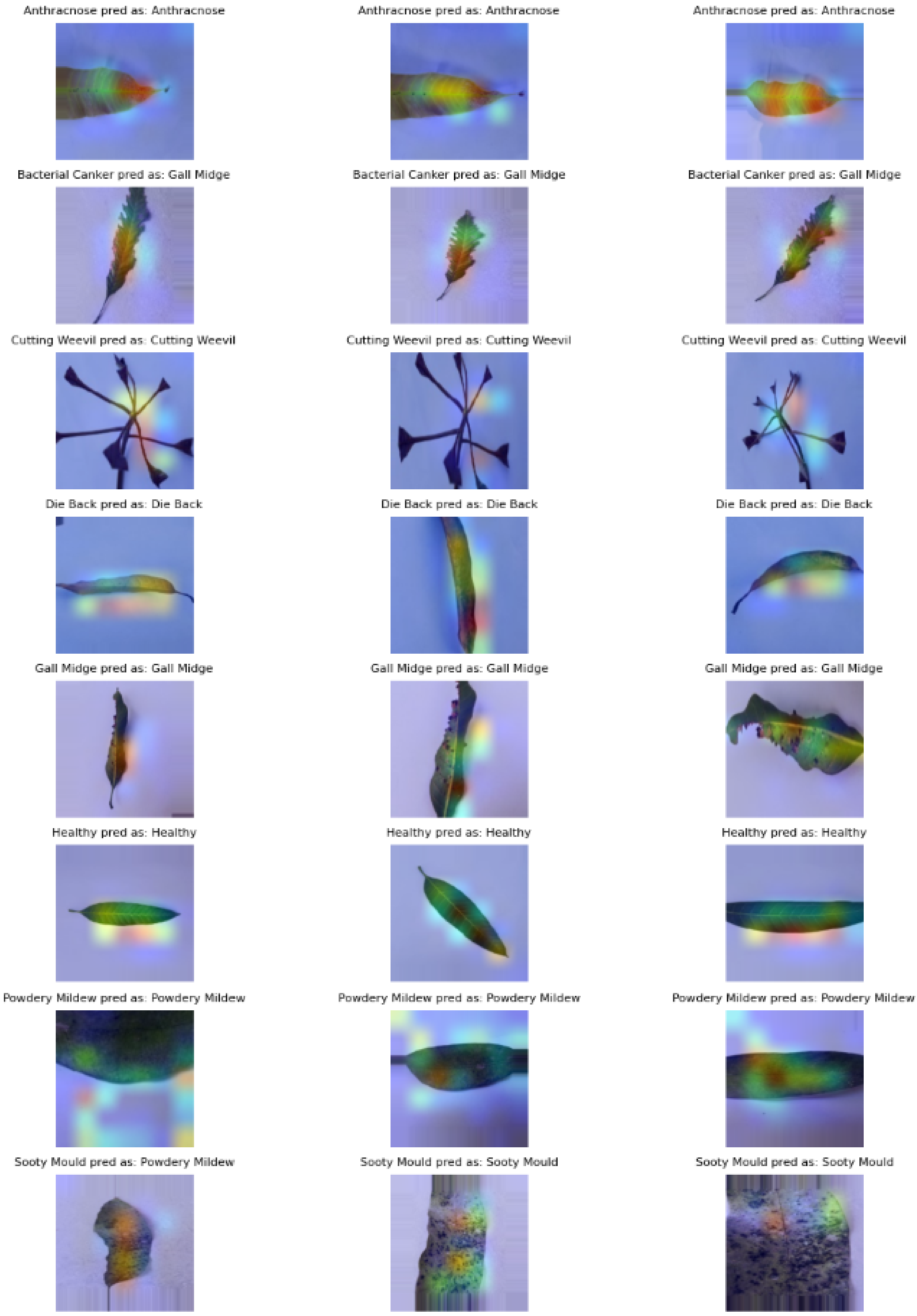


Figure 4.13: Grad-Cam applied on ResNet50 model

Fig. 4.10, 4.11, 4.12, 4.13 represents our findings after applying it on 3 images of each of the 8 classes for Xception, MobileNetV2, DenseNet201, ResNet50 models respectively. Grad-CAM produces heat-maps that show the areas of an input image that are thought to have the greatest influence on the model's verdict. The coloring scheme employed in Grad-CAM heat-maps serves as a visual aid to convey the relative importance of different regions within the image. Typically, brighter colors, such as red or yellow, indicate areas of higher importance, while darker colors, such as blue or purple, denote regions of lower significance. This gradient representation of attention allows observers to discern the model's focus across the image, with varying intensities of color representing the degree of attention attributed to each pixel. The Grad-CAM heat-map, overlaid on the original image, offers insightful information about the model's method of decision-making by graphically representing the areas in which the model focuses its attention. This allocation is based on the magnitude of gradients with respect to pixel values, with higher gradients corresponding to regions that significantly influence the model's prediction. Consequently, regions with higher heat-map values are interpreted as being more critical to the model's decision, while regions with lower heat-map values are considered less influential.

4.4.6 LIME

Figure 4.14: Original Image from our dataset of Anthracnose class

Xception

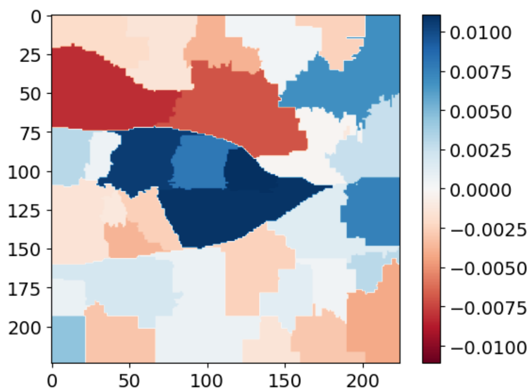


Figure 4.15: Detailed explanation of our original image after applying LIME

MobileNetV2

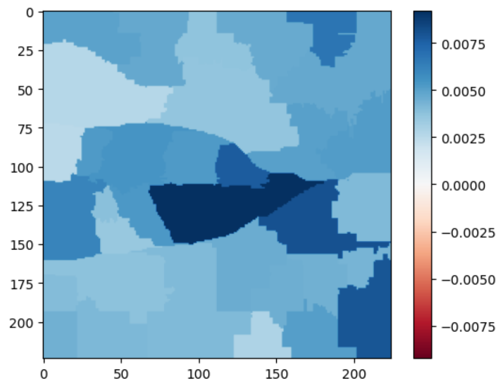


Figure 4.16: Detailed explanation of our original image after applying LIME

DenseNet201

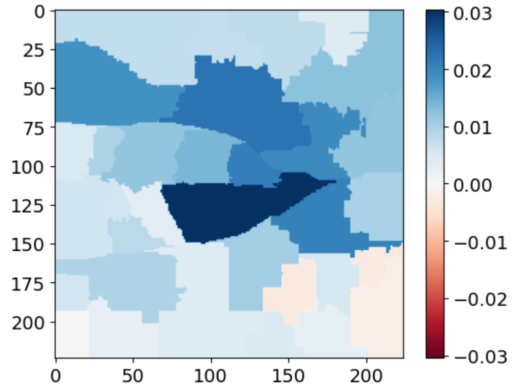


Figure 4.17: Detailed explanation of our original image after applying LIME

ResNet50

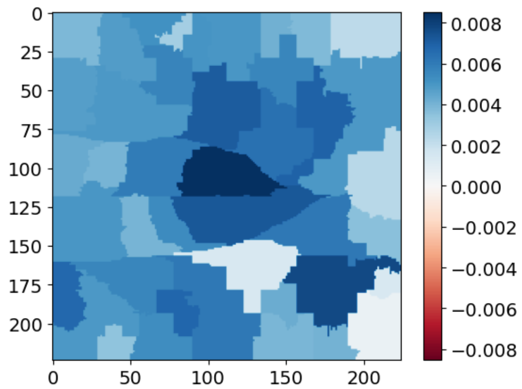


Figure 4.18: Detailed explanation of our original image after applying LIME

Fig. 4.14 visualize the original image we chose for our purpose and the result after applying lime to all 4 models, are given in Fig. 4.15, 4.16, 4.17, 4.18 for Xception, MobileNetV2, DenseNet201, ResNet50 respectively. All of the models are emphasizing on the same diseased area to predict the outcome, along with some incorrect sections in certain instances, which can be improved by retraining our models or changing the architecture. LIME allows us to see how the model is working in predicting the disease. The numerical value on the right side determines the influence of that particular part in indicating a disease. The color for the most influential part is dark blue, then light blue, and then gradually going towards the least influential part. Negative values impact the model predicting the outcome negatively. It means that area of picture is inconsistent with its outcome based on how it was trained. While the dark blue is supporting the outcome it has provided. With LIME, we can determine which features are positively or negatively affecting the model and it can aid in modifying the training process, feature selection.

4.4.7 Comparison

Table 4.6: Models Comparison

Paper	Model	Accuracy	Precision	Recall	F1 score	Dataset Images	Epochs
Paper 1[40]	MDCN	97.43	96.35	96.24	97.21	4000	20
Paper 2[41]	EfficientNetB4+Custom CNN (hybrid model)	93.01	93.23	94.00	93.07	4873	50
Our paper	ResNet50	98	98	98	98	4000	10

Table 4.6 indicates the differences of accuracy, precision, recall and F1 score of the models that achieved highest accuracy of each paper. We studied other papers which used the same dataset as us and tried to find the different techniques that they used compared to us and the difference between accuracy and results. In this paper[40], the authors propose the use of deep learning, specifically Dense Convolutional Networks (DenseNet), as a promising solution for mango leaf disease classification. However, the authors acknowledge challenges specific to mango leaf disease classification, such as data imbalance, overlapping symptoms, image quality diversity, ambient noise, incomplete data, and computational complexity. Motivated by these challenges, the authors introduce the Modified Dense Convolution Network (MDCN), a novel algorithm tailored explicitly for mango leaf disease classification. At 20 epochs, the MDCN algorithm achieves an accuracy of 97.43%, with 96.35% precision,96.24% recall, and 97.21% F1 score.

This study[41] focused on a Hybrid Deep Learning methodology to detect mango leaf diseases. They used four different models for their purpose - Custom CNN, VGG-16, EfficientNetB4, and a hybrid model. Among them hybrid model produced the best results, with 93.01% accuracy.

Chapter 5

Conclusion

Mangoes are important to Bangladesh's agricultural and economy sector due to its enormous amount of cultivation, but various mango leaf diseases hamper our economy and mango production every year. It is time-consuming to identify mango leaf diseases for the farmers by continual monitoring and observation with naked eyes. As a result, it is hard to precisely identify mango leaf diseases at an initial stage, and there are relatively few traditional techniques for treating these diseases. Since the plant gets its nutrition from the leaves, it is essential to diagnose leaf diseases as early as possible. Initial identification of the mango leaf diseases would aid Bangladesh in developing its economy. A popular and accurate method for quickly identifying mango leaf illnesses is the application of CNN, a deep learning-based methodology. In this paper, we have mainly used four models of CNN machine learning algorithms such as MobileNet, ResNet50, DenseNet201, and Xception. In order to identify and categorize mango leaf illnesses, we trained, compared, and assessed the accuracy of these models. To evaluate the models' accuracy, we have used the mango leaf image dataset from kaggle in the preprocessing phase. We found the highest accuracy for ResNet50 model and it acheived 98% F1 score in detecting the mango leaf diseases. We further applied XAI techniques, such as Grad-CAM and LIME to comprehend our deep learning models' operation from a human perspective to improve our used models. It helps the humans to understand that not only the decision predicted by the models was correct, the process or the reading of the picture was also done in the right way for the right result. In the future, we aim to develop a web version where user can provide a mango leaf picture and our applied models will analyze the picture and provide a solution to assist our agricultural sector.

Bibliography

[1] D. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," Nov. 2015.

[2] I. Mukti and D. Biswas, "Transfer learning based plant diseases detection using resnet50," Dec. 2019, pp. 1–6. DOI: 10.1109/EICT48899.2019.9068805.

[3] M. Rahman, A. A. Biswas, A. Rajbongshi, and A. Majumder, "Recognition of local birds of bangladesh using mobilenet and inception-v3," *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 309–316, Aug. 2020. DOI: 10.14569/IJACSA.2020.0110840.

[4] J. Bai, B. Wang, C. Chen, J. Chen, and Z.-h. Fu, "Inception-v3 based method of lifeclef 2019 bird recognition," Aug. 2019.

[5] P. Dhankhar, "Resnet-50 and vgg-16 for recognizing facial emotions," *International Journal of Innovations in Engineering and Technology*, vol. 13, no. 4, pp. 126–130, Nov. 2019.

[6] C. Lin, G. Zhao, Z. Yang, *et al.*, "Cir-net: Automatic classification of human chromosome based on inception-resnet architecture," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 3, pp. 1285–1293, 2022. DOI: 10.1109/TCBB.2020.3003445.

[7] C. Lin, L. li, W. Luo, K. Wang, and J. Guo, "Transfer learning based traffic sign recognition using inception-v3 model," *Periodica Polytechnica Transportation Engineering*, vol. 47, Aug. 2018. DOI: 10.3311/PPtr.11480.

[8] C. Ferreira, T. Melo, P. Sousa, M. Meyer, A. Campilho, and E. Shakibapour, "Classification of breast cancer histology images through transfer learning using a pre-trained inception resnet v2," in Jun. 2018, pp. 763–770, ISBN: 978-3-319-92999-6. DOI: 10.1007/978-3-319-93000-8_86.

[9] J. M. Patel and N. C. Gamit, "A review on feature extraction techniques in content based image retrieval," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 2259–2263. DOI: 10.1109/WiSPNET.2016.7566544.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973. DOI: 10.1109/TSMC.1973.4309314.

[11] D. L. Padmaja and B. Vishnuvardhan, "Comparative study of feature subset selection methods for dimensionality reduction on scientific data," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 31–34. DOI: 10.1109/IACC.2016.16.

[12] M. I. Khaskheli, “Mango diseases: Impact of fungicides,” in *Horticultural Crops*, H. K. Baimey, N. Hamamouch, and Y. A. Kolombia, Eds., Rijeka: IntechOpen, 2020, ch. 8. DOI: 10.5772/intechopen.87081. [Online]. Available: https://doi.org/10.5772/intechopen.87081.

[13] A. Big. “World mango and guava production by country.” (2021), [Online]. Available: https://www.atlasbig.com/en-gb/countries-by-mango-guava-production.

[14] S. Halder and L. Pangkhua, “Amrapali mango production highest among all varieties,” *The Daily Star*, 2023. [Online]. Available: https://www.thedailystar.net/business/economy/news/amrapali-mango-production-highest-among-all-varieties-3374591?fbclid=IwAR3JQRDuoJ4305laMZIC9GcOSPTv8FdCZtmW1H-6GHhxZYUbOT0D-FTJOEI.

[15] M. R. Mia, S. Roy, S. K. Das, and M. K. Rahman, “Mango leaf disease recognition using neural network and support vector machine,” *Iran Journal of Computer Science*, pp. 185–193, 2020. DOI: 10.1007/s42044-020-00057-z.

[16] R. Hasan, “Dieback disease worries mango growers in chapainawabganj,” *The Daily Star*, 2015. [Online]. Available: https://www.thedailystar.net/country/dieback-disease-worries-mango-growers-chapainawabganj-98728.

[17] N. I. Mahbub, F. Naznin, M. I. Hasan, S. M. R. Shifat, M. A. Hossain, and M. Z. Islam, “Detect bangladeshi mango leaf diseases using lightweight convolutional neural network,” in *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2023, pp. 1–6. DOI: 10.1109/ECCE57851.2023.10101648.

[18] M. Islam, K. Poddar, I. Hossain, M. S. Chowdhury, H. Mehraj, and D. Uddin, “Seedling diseases of mango in four districts of banglaadesh,” vol. 10, pp. 55–61, Apr. 2015.

[19] A. Rajbongshi, T. Khan, M. M. Rahman, A. Pramanik, S. M. T. Siddiquee, and N. R. Chakraborty, “Recognition of mango leaf disease using convolutional neural network models: A transfer learning approach,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1681–1688, 2021. DOI: 10.11591/ijeecs.v23.i3.pp1681-1688.

[20] A. Sharma, R. Bijral, J. Manhas, and V. Sharma, “Mango leaf diseases detection using deep learning,” *International Journal of Knowledge Based Computer Systems*, vol. 10, no. 1, pp. 40–44, Jun. 2022. [Online]. Available: http://publishingindia.com/downloads/6827.pdf.

[21] S. Arya and R. Singh, “A comparative study of cnn and alexnet for detection of disease in potato and mango leaf,” in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, vol. 1, 2019, pp. 1–6. DOI: 10.1109/ICICT46931.2019.8977648.

[22] R. Saleem, J. H. Shah, M. Sharif, and G. J. Ansari, “Mango leaf disease identification using fully resolution convolutional network.,” *Computers, Materials & Continua*, vol. 69, no. 3, 2021.

[23] P. Kumar, S. Ashtekar, S. S. Jayakrishna, K. P. Bharath, P. T. Vanathi, and M. Rajesh Kumar, "Classification of mango leaves infected by fungal disease anthracnose using deep learning," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1723–1729. DOI: 10.1109/ICCMC51019.2021.9418383.

[24] S. V. T. D. Tan Nhat Pham Ly Van Tran, "Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection," *IEEE Access*, vol. 8, pp. 189 960–189 973, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3031914.

[25] S. Arivazhagan and S. V. Ligi, "Mango leaf diseases identification using convolutional neural network," *International Journal of Pure and Applied Mathematics*, vol. 120, no. 6, 2018. [Online]. Available: http://www.acadpubl.eu/hub/Special%20Issue.

[26] J. H. S. Rabia Saleem, M. Y. Muhammad Sharif, and J. C. Hwan-Seung Yong, "Mango leaf disease recognition and classification using novel segmentation and vein pattern technique," *Applied Sciences*, vol. 11, no. 24, 2021. DOI: 10.3390/app112411901. [Online]. Available: https://doi.org/10.3390/app112411901.

[27] M. H. K. Mehedi, A. S. Hosain, S. Ahmed, *et al.*, "Plant leaf disease detection using transfer learning and explainable ai," Oct. 2022.

[28] S. Jain and P. Jaidka, "Mango leaf disease classification using deep learning hybrid model," Feb. 2023, pp. 1–6. DOI: 10.1109/PIECON56912.2023.10085869.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[30] A. Rastogi, "Resnet50," 2022. [Online]. Available: https://blog.devgenius.io/resnet50-6b42934db431.

[31] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[32] V. Martinek, "Cross-entropy for classification," 2020. [Online]. Available: https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, 2018. arXiv: 1608.06993 [cs.CV].

[34] S.-H. Tsang, "Review: Densenet — dense convolutional network (image classification)," 2018. [Online]. Available: https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *Mobilenetv2: Inverted residuals and linear bottlenecks*, 2019. arXiv: 1801.04381 [cs.CV].

[36] Z. M. Chng, "Using depthwise separable convolutions in tensorflow," 2022. [Online]. Available: https://machinelearningmastery.com/using-depthwise-separable-convolutions-in-tensorflow/.

[37] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, 2017. arXiv: 1610.02357 [cs.CV].

[38] “Local interpretable model-agnostic explanations.” (), [Online]. Available: https://interpret.ml/docs/lime.html#:~:text=LIME%20works%20by%20perturbing%20any,set%20for%20the%20glassbox%20model.

[39] S. I. Ahmed, M. Ibrahim, M. Nadim, *et al.*, “Mangoleafbd: A comprehensive image dataset to classify diseased and healthy mango leaves,” *Data in Brief*, vol. 47, p. 108 941, 2023, ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2023.108941. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923000598.

[40] C. Chandrashekar, K. Vijayakumar, K. Pradeep, and A. Balasundaram, “Mdcn: Modified dense convolution network based disease classification in mango leaves,” *Computers, Materials Continua*, vol. 78, pp. 2511–2533, Feb. 2024. DOI: 10.32604/cmc.2024.047697.

[41] V. C. P. and P. K., “Revolutionizing mango leaf disease detection: Leveraging segmentation and hybrid deep learning for enhanced accuracy and sustainability,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4, pp. 121–131, Sep. 2023. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/3460.