Leveraging Deep Learning Algorithms For The Timely Detection Of Diseases in Bean Leaves

by

Driciru Fiona
20241030
Ajani Denish
21301742

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

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 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 cited through full and accurate referencing.

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:

Driciru Fiona	Ajani Denish
20241030	21301742

Approval

The thesis/project titled “ Leveraging deep learning algorithms for the timely detection of diseases in bean leaves.” submitted by

1. Driciru Fiona(20241030)
2. Ajani Denish(21301742)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2024.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD

Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
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

In sub-Saharan Africa, where agriculture is a major commercial activity, the security of staple crops like beans is threatened by persistent diseases, notably bean rust and angular leaf spot. Uromyces appendiculatus, the fungus that causes bean rust, produces rust-colored pustules, whereas Pseudomonas syringae pv. phaseolicola, the cause of angular leaf spot, produces recognizable angular lesions. It is estimated that these diseases cost the agricultural sector millions of shillings each year in Uganda due to reduced bean yields, increased costs for disease control measures as well as the need to remove infected bean crops. A 2017 study found that Angular Leaf Spot caused a major yearly production loss of 384.2 tons especially in the Eastern region of the country where over 63% of the people participate in the activity, this raised major questions. In response to the demand for contemporary, data-driven approaches, this study presents a Deep Learning-based approach for the rapid and precise detection of angular leaf spot and bean rust by utilizing CNN algorithms with the free and open-source TensorFlow package and a public dataset of bean leaf images. This study has trained five models to detect bean rust and angular leaf spot in bean leaves. The prediction accuracies of the models were evaluated and the accuracies were 96%, 95%, 94%, 33% and 88% for Xception, ResNet50, DenseNet201, VGG19 and InceptionV3 respectively. Additionally, the performance of the models is evaluated using different metrics like F1-score, Precision and Recall. The Xception model with the highest prediction accuracy and Recall of 0.96 stood out as the top-performing model which was selected for further usage where the model was tested on images of two unhealthy classes and a healthy class. These algorithms demonstrate increased diagnostic accuracy and present a viable way to reduce the financial burden that agricultural diseases impose on Uganda and sub-Saharan Africa. Furthermore, the precise bean leaf disease identification system uses explainable AI frameworks such as LIME (Local Interpretable Model-Agnostic Explanations) to improve interpretability by visualizing the layer-wise feature extraction. These frameworks present an understanding of the attributes driving the categorization of diseases and provide details about the Deep Learning models' choices hence promoting trust in the diagnostic results.

Keywords: Deep Learning; Convolutional Neural Networks(CNN); Bean rust; Angular Leaf spot; Image processing; Classification

Acknowledgement

First and foremost, we are grateful to GOD, who has guided us throughout the entire process of writing our thesis with minimal interference.
We would like to express our gratitude to our supervisor, Dr.Amitabha Chakrabarty, for his kind assistance and insightful guidance during the course of our study. He was always willing to lend a hand when we needed it.
We would also like to express our heartfelt appreciation to Mr. MD.Fahim-Ul-Islam for his guidance and support which were essential to the accomplishment of our research.

Finally, we would like to thank our parents and friends for their unwavering support and prayers, without which this accomplishment would not have been possible. Their support has been instrumental in getting us to this point where we can officially graduate.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Beans (Phaseolus vulgaris), a member of the legume family are an important commercial crop[9] as well as a staple food. They are one of the most widely grown crops in Uganda. In addition to being a valuable food source, beans are cultivated because they can withstand soils with low nitrogen levels, which reduces the need for excessive fertilizer application[8]. They are highly nutritious and provide important nutrients such as fiber, antioxidants, minerals, vitamins, and protein. Surprisingly, beans are the region's second-largest source of protein for consumers after maize, and they account for the third-largest portion of caloric intake after maize and cassava. The Uganda Bureau of Statistics (UBOS) conducted a thorough study in 2017 that highlights the widespread cultivation of beans, with over 63% [Jjagwe, K. et al, 2022] of Ugandan households—particularly those in the Eastern region—engaging in the practice. The majority of bean farmers in this area view bean farming as their main source of income. The Food and Agriculture Organization (FAO)[11] has reported that, despite widespread cultivation, crop production per unit area has decreased over the past ten years due to problems like crop pests, diseases, and other factors. This makes it necessary to investigate the causes of this decline in more detail and to put strategic plans in place to deal with them and boost Uganda's bean harvest. Bean rust and angular leaf spot are examples of diseases that cause a significant amount of losses in Uganda's bean crop production every year, accounting for more than 18.5%[12] of the total yield. In addition to lowering the earnings of farmers, substantial crop losses from angular leaf spot and bean rust endanger food security and constrain Uganda's export revenues even in the face of present economic recession. Several standard procedures, such as crowdsourcing apps[7] and smart farming technologies, are available for early detection of bean rust and angular leaf spot. Nevertheless, these processes require a testing facility, time, and technical know-how.
The figure 1.1 below shows the annual crop distribution in Uganda with Beans dominating all the other crops.

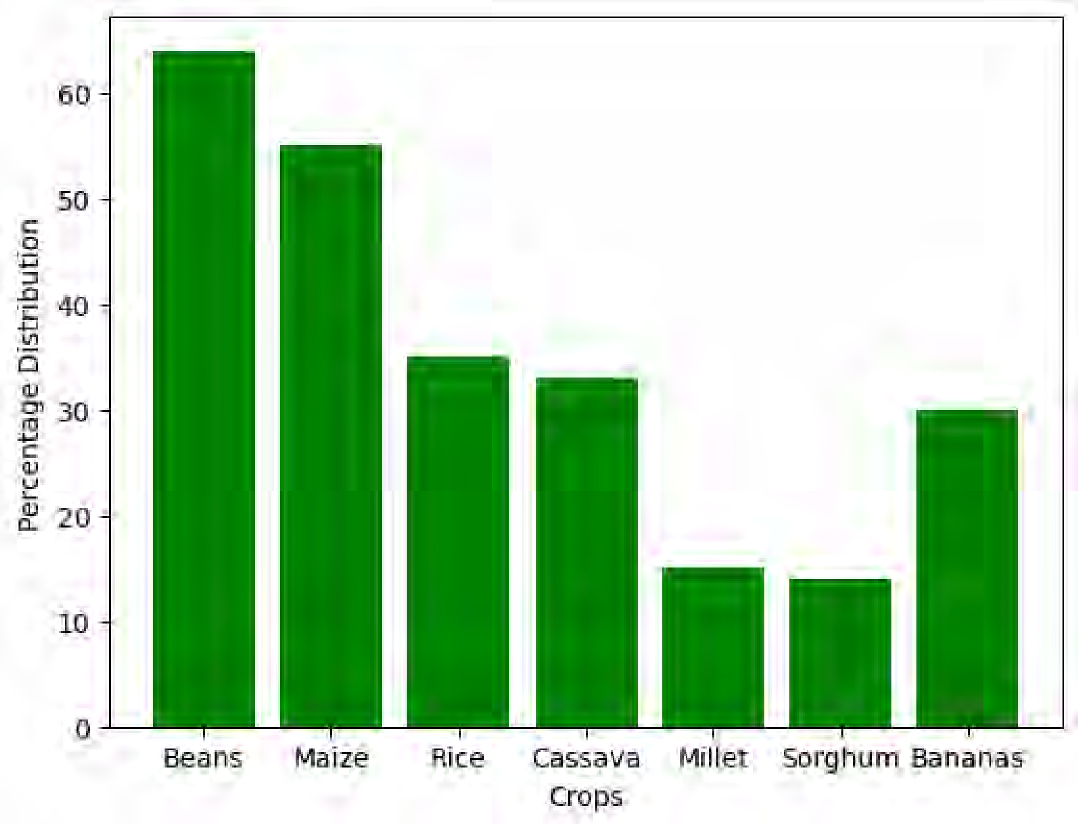

Figure 1.1: Annual Crop Production in Uganda

This research aims to leverage deep learning algorithms for bean leaf disease detection and examine the various classifications. Getting a publicly accessible dataset is the initial stage in our process. Next, we will augment the images to prevent overfitting and boost variety then we will train each individual CNN model on the pre-processed images. We'll start with a pre-trained model that has been trained on a sizable dataset of pictures because we'll be using the transfer learning approach. After the CNN models are trained, we will assess how well they function using a test set of pictures. Finally, a range of metrics, such as accuracy, f1-score, and recall, will be used to assess the performance. This technology has the potential to aid in early identification and treatment of bean rust and angular leaf spot, hence assisting farmers and other stakeholders. Technological advancements can contribute to increased treatment effectiveness and safeguard the overall health and yield of bean crops by giving a prompt and precise diagnosis. By raising the health and yield of beans and lowering the expense of treating these diseases, this approach can benefit Uganda's agricultural sector.

1.1 Motivation

Our goal is to provide an additional tool through the use of Deep Learning in the detection of the Bean leaf diseases in Uganda. These technologies can be very helpful in monitoring crop health and disease diagnosis, even though they are not meant to replace agronomic diagnosis in an agricultural setting. This becomes particularly important during crop-affecting outbreaks or in situations where resources are scarce.

Through acknowledgment and assistance from mentors and organizations, we hope to progress our studies and improve accessibility via online and mobile platforms. In addition to increasing awareness of possible crop illnesses, this improved accessibility will add to the body of information pertaining to the intersection between agricultural technology and intelligent machines.

1.2 Thesis Contribution

1.2.1 Problem Statement

The human-based diagnosis of bean crop diseases poses a serious problem to Uganda's agricultural industry, and the lack of an automated system for prompt identification of diseases has a significant negative influence on crop production. When diseases like angular leaf spot and bean rust are identified manually, it can result in errors, inconsistencies, and disruptions. This lack of automated detection technologies makes it more difficult for the agricultural society to take action quickly to new risks and put preventative actions in place in a timely manner.As a result, the production of bean harvests is hampered, which affects farmers and prevents the agricultural industry's total progress. In order to enable preventive and timely identification of bean crop diseases in Uganda, it is essential that a dependable, automated system that makes use of cutting-edge technologies—particularly deep learning algorithms—be developed and put into place.

1.2.2 Solutions

Our work contributes significantly to the prompt detection of illnesses in bean leaves. Using a publicly available dataset, we methodically trained deep learning models intended to identify a class of healthy beans along with common diseases like angular leaf spot and bean rust.We performed a thorough evaluation of these models' performances due to their varying degrees of accuracy, which include Xception, ResNet50, DenseNet201, VGG19, and InceptionV3. A Flask web-based app was developed to improve the accessibility and functionality. It functions as a simple user interface for prediction and presenting disease classification results. Our contribution transcends the technical domain by giving Ugandan stakeholders and agricultural experts a useful tool. Our work aims to greatly enhance the effectiveness, precision, and responsiveness of identifying diseases by providing a computerized approach to the prevalent issue of human-based identification of diseases in bean fields. In the end, this all-encompassing strategy may reduce financial losses, improve the productivity of crops, and support the expansion of Uganda's agriculture industry as a whole.

1.3 Summary of Contributions

The study presents a unique method for the timely identification of diseases in bean leaves by utilizing deep learning algorithms. By focusing on the common problems of bean rust and angular leaf spot, the system successfully classifies three types of bean rust, angular leaf spot, and healthy bean leaves with accuracy by using deep learning models that have been trained on a public dataset. By using a Flask web application, user accessibility is improved in addition to predictions and results being shown. This scientific technique offers a computerized and effective way to recognize diseases that negatively impact bean crop yields, filling a major gap in Uganda's farming system where human-based detection methods fall short. By providing farmers and other agricultural experts with a useful tool for controlling and safeguarding bean crops, the suggested method advances precision agriculture.

1.3.1 Methodology

The research approach utilized in this study focuses on bean rust and angular leaf spot, along with a healthy class of bean leaves and it uses deep learning algorithms to detect diseases in bean leaves in a timely manner. The research focuses on training deep learning models that can distinguish between the three aforementioned categories-using a public dataset. To improve the model's capacity to generalize to various contexts, image augmentation techniques are utilized. After training, the models are incorporated into a Flask web application to enable accessible user interfaces and instantaneous prediction. Further details regarding the methodology including training parameters, model architectures, evaluation metrics are discussed in the subsequent Chapters.

1.4 Thesis Outline

Chapter 1
A summary of the economic significance of beans (Phaseolus vulgaris) in Uganda is given in this chapter. It highlights how important beans are as a commercial crop and as an essential food supply. The chapter presents the central idea of the thesis, which is leveraging deep learning models for the timely detection of bean leaf diseases, with a focus on angular leaf spot and bean rust. It provides context for the thesis by emphasizing the motivation, problem statement, and the thesis contributions.

Chapter 2
In this chapter, a thorough assessment of the literature review is conducted with an emphasis on studies that have used CNN algorithms—especially deep learning approaches—to detect diseases in a variety of crops. The chapter provides insight on the development and efficacy of deep learning techniques in the field of agricultural disease detection by critically analyzing the various CNN model architectures used in these research.

Chapter 3
This chapter turns its emphasis to a thorough examination of the dataset. It describes the repository from which the data was obtained and offers an understanding of its components and characteristics. It also explores the image pre-processing methods used, including augmentation, shedding light on the tactics used to improve the quality and diversity of the dataset. In addition, the chapter reveals details of the data splitting strategies used, providing a solid basis for further study and model

training.

Chapter 4
A thorough description of the methodology, specifically the deep learning workflow used in the study is given in this chapter. The study's use of transfer learning models is described in great length. Additional details regarding the model selection and training parameters deepen the reader's comprehension and provide context for the analysis and results that follow.

Chapter 5
This chapter provides an illustrative and statistical focus as it details the study's implementation and results. The chapter explores performance measures with charts showing losses and accuracies of the models. Extensive classification reports that include important evaluation metrics such as recall, precision, and F1 score offer a thorough analysis of the models. A detailed analysis of the models enhances the discussion by illuminating their respective benefits and drawbacks. In order to provide a more sophisticated understanding of the models' decision-making processes, the chapter also adds a layer of interpretability through the use of LIME to visualize image samples.

Chapter 6
This chapter discusses how the Flask web workflow is implemented, explaining its features and its patterns of user interaction. It addresses limitations of the system by providing an interactive investigation of integrating the trained models into a user-friendly online application. This chapter functions as a link between complex deep learning models and practical applications, offering insightful information on the operations and possible drawbacks of the platform .

Chapter 7
This Chapter summarizes our Conclusions and describes our Future work. It summarizes the main conclusions drawn from the research and suggests directions for more investigation. Our goal is to come up with a scale-down version of the best-performing model that can be incorporated into lightweight IoT devices. This optimistic chapter highlights how the suggested system is still evolving and lays the groundwork for future developments and useful applications.

Chapter 2

Literature Review

Within the field of machine learning, deep learning (DL) attempts to extract sophisticated features from unprocessed data. It is made up of several stages of neural networks, with lower levels identifying basic components like margins and lines and higher layers specializing in more complex qualities. The output of the layer before it is used as the input for each subsequent layer. Bean yields are severely impacted by diseases, which result in a loss of more than 18.5%[12] of the overall output. Timely diagnosis and identification of afflicted plants is an essential approach for dealing with this problem. The ability to predict and prevent diseases in bean plants has greatly improved with the use of sophisticated Artificial Intelligence and deep learning[25] classification techniques. By combining various classification algorithms, diseases that harm bean leaves, such as Bacterial Blight or Rust, may now be predicted to some extent. For instance, machine learning algorithms outperformed conventional methods in a study when it came to forecasting outbreaks of bean illnesses such bacterial blight. Experiments on plant disease detection techniques were conducted , with an impressive accuracy rate of 96%[1]. With the use of picture datasets and different data classifiers, data engineers have been concentrating their efforts on improving the identification, evaluation, and forecasting capacities of diseases. The imaging interpretation that these studies focus on is done by CNNs, a type of deep learning method. A model that has been trained with the initial architecture and learnt weights unaltered is leveraged in this manner. Employing the feature extraction expertise it acquired from its first training, the CNN algorithm processes a set of photographs of a different type in this phase. That is, an entirely fresh network that does the intended categorization task is built using the features that were captured. This methodology is commonly employed in order to avoid the substantial computational expenses linked with training extremely deep networks from the very beginning or to maintain the development of significant feature extractors during the initial stage. A comprehensive literature review of earlier research articles utilizing the same dataset is essential when using a previously processed dataset for study. Analysis of the current literature provides us with crucial insights on the methods, models, as well as strategies used by previous researchers. This allows us to expand on their work, spot any limitations or discrepancies, and add to the body of research already known in the area of study. In addition, examining the results and outcomes of earlier research gives us a benchmark that allows us to measure the effectiveness and applicability of our chosen methodology. It enables us to assess the accuracy of our outcomes, pinpoint areas in need of enhancement

and to contrast the results we obtained with the ones of similar studies.
Currently, CNN algorithms have been able to attain human-like proficiency in several complex pictures classifications, such as image analysis and identification. Since Yann LeCun's 1998 creation of LeNet[16], one of the first effective CNN designs, which was used to recognize scribbled numbers, numerous CNN architectures have been developed across the course of twenty years since then. LeNet is viewed as a simple architecture with three convolutional, two average pooling, and two fully linked layers, even if it is contrasted to more sophisticated CNN algorithms nowadays. In the parts that follow, a brief description of the CNN architectures used in the examined research, their usage in identifying bean leaf illnesses, and their finding results are given.

2.1 Deep Learning Algorithms

The emergence of artificial intelligence (AI) is revolutionizing the detection and categorization of agronomic images enabling farmers to more successfully battle destructive bean diseases like angular leaf spot and bean rust. Computerized models can outperform humans in time and precision when assessing everything from aerial shots to close-up shots of leaves, allowing for swift diagnosis and prompt actions. With its remarkable flexibility and lightning-fast responsiveness, precision agriculture[29] can achieve enormous potential by enabling proactive disease prevention and monitoring in real time. Food availability can be improved, environmentally friendly agriculture can be encouraged, and losses in yield can be greatly decreased by early diagnosis of angular leaf spot and bean rust. In the rapidly developing field of artificial intelligence, deep learning is the foundation for image analysis, while machine learning serves as the general data manipulator. Deep learning's adaptability and context-aware learning skills have made it a rising star in agricultural research. Deep learning addresses the intricacies of data related to agriculture by imitating the sharp vision of experienced farmers. It does this by extracting critical elements that are necessary for the identification and diagnosis of grave bean diseases like bean rust and angular leaf spot. The effectiveness of it is derived from three primary learning techniques: unsupervised learning, which looks for hidden patterns in unlabeled data, semi-supervised learning, which uses a combination known and unlabeled information to forecast the disease and supervised learning[13] which uses labelled data. A variety of techniques based on deep learning (DL) have shown promise in recognizing and classifying a variety of diseases affecting bean leaves, such as angular leaf spot and bean rust. Convolutional neural networks (CNNs) are particularly notable among them due to their outstanding image analysis performance. CNNs have the potential to greatly increase the precision of disease detection[27] algorithms by enhancing the visual quality of bean leaf pictures. But it's important to recognize that, despite the enormous potential that DL has to transform bean disease management, there are still issues that need to be addressed through further research and improvement in order to safeguard its broad and dependable use in agricultural contexts. The rising threat to crop diseases, especially in the production of maize, emphasizes the urgent need for effective management through prompt identification and precise severity assessment. In order to improve feature representation, this paper presents a CNN model that combines features from the ResNet101 and Inception-V3 models for transfer learning[31].An attention layer concentrates on important disease-related characteristics to further improve performance. The model's impressive accuracy of 0.956 and high specificity of 0.985, obtained through hyperparameter tuning and a 5-fold study, demonstrate the model's effectiveness in primary stage disease diagnosis. With the use of cutting-edge technologies, this proactive approach tackles challenges associated with growing maize, underscoring the paper's commitment to the advancement of agricultural sustainability through innovative disease diagnosis and control. Both the productivity and quality of mango cultivation are seriously threatened by mango leaf diseases[30], which make accurate diagnosis more difficult to achieve than with a casual observer. Recently, computer-aided methods and machine learning have been used to classify diseases on mango leaves. Unfortunately, these techniques' efficiency has been hampered by issues including increasing overfitting, computational cost, and feature dimensionality, in addition to a deficiency of discriminative feature qualities. In order to overcome these drawbacks, the present work presents a brand-new classification method for diseases of mango leaves that includes four crucial phases: feature selection, learning and classification, performance evaluation, and data processing.With 1,536 photos from the open Kaggle database that are divided into categories of healthy and unhealthy images, the dataset used by the suggested system has excellent performance metrics. The top model, for example, has a sensitivity of 96.2% and an accuracy of 97.9%. These results highlight how well the system that was built was able to classify diseases of the mango leaf, providing a viable way to improve the production and management of mango farming. According to the Ghana Statistical Service, cocoa is a crucial cash crop that makes a substantial economic contribution to Ghana[33]. It accounts for around 3% of the country's GDP and about 20% of all export earnings. But recent obstacles, especially the spread of cocoa diseases like Black Pod and Swollen Shoot, have threatened income and resulted in an 11% reduction in crop yield. This research aims to strengthen early detection and diagnosis of these major diseases affecting cocoa output by utilizing mobile technology and machine learning (ML) approaches, realizing the need of action. The method entails creating a distributed mobile application that gives farmers the ability to film or take pictures of their cocoa plantations. With a focus on Swollen Shoot and Black Pod, the system uses deep Convolutional Neural Networks (CNNs) for image analysis, classification, and illness diagnosis. Four different CNN models are trained using an extensive dataset of 2,828 cocoa photos distributed among three class labels: CenterNet ResNet50 V2, EfficientDet D0, SSD MobileNet V2, and SSD ResNet50 V1 FPN.SSD MobileNet V2 is the most effective and broadly applicable of these models, with an impressive detection confidence score of about 88.0%. This study is at the vanguard of using cutting-edge technology to manage cocoa diseases. It provides farmers with an easy-to-use tool for quick and accurate diagnosis, which will increase the productivity and sustainability of Ghana's cocoa production industry. As a basic staple food for many nations, rice depends on consistent productivity, and early detection of rice leaf diseases becomes essential to preserving this sustainability. The conventional approaches of disease identification, which are primarily manual and equipment-dependent, are time-consuming and ineffective, which adds to the rising expenses of chemical testing and visual identification. By utilizing Deep Learning (DL) and transfer learning approaches, this work aims to address these issues by accurately identifying and classifying rice leaf diseases. A comprehensive dataset of 5932 self-generated pictures of rice leaves[35] is curated by the research, supplemented with benchmark datasets, and classified into 9 classes that represent various disease states. These classes include leaves in varying stages of blast, brown spot, blight, tungro, and health. To enhance image diversity, data augmentation comes after meticulous manual labeling and dataset segmentation that has been verified by horticulture specialists. Together with well-known transfer learning techniques (VGG16, Xception, ResNet50, DenseNet121, Inception ResnetV2, and Inception V3), the suggested customized Convolutional Neural Networks models are carefully assessed. With a 99.94% generalization accuracy, the unique VGG16 model in particular performs exceptionally well, outperforming current state-of-the-art benchmarks. Moreover, the study uses explainable AI to improve interpretability by visualizing the layer-wise feature extraction. This research offers an effective and precise way to reduce the difficulties associated with manual identification and chemical testing in rice cultivation. By successfully fusing DL techniques with transfer learning, it significantly advances the field of rice leaf disease detection. As a common grain crop in semi-arid areas, millets have drawn interest due to their possible healing properties for cancer as well as their nutritional value, which includes fiber, magnesium, phosphorus, zinc, and serotonin. In particular, it has been found that Foxtail Millet Bran (FMBP)[28] has active ingredients that stimulate increased ROS production, hence inhibiting the STAT3 pathway and slowing the proliferation of cancerous cells in the intestine. The importance of millets in cancer therapy is increased by their potential health benefits, notably the fact that they are gluten-free. In the medical field, colonoscopy, histological image analysis, and magnetic resonance imaging have historically been used to identify colon polyps. On the other hand, manual methods are recognized for being less precise and time-consuming. Remarkably, there is a marked deficiency in the availability of specialized models designed to assess how well millet eating heals colon cancerous cell growth. By putting out a deep learning-based MobileNet-V3 model for the identification and categorization of histology images, this study seeks to close this gap. Researchers and medical professionals will find the envisioned model useful as it has the potential to automate the evaluation of malignancy status both before and after millet eating. This work presents a revolutionary way that could transform the assessment of the therapeutic effect of millets on colon health by utilizing deep learning techniques. This method offers a more accurate and efficient substitute for conventional approaches. Plant diseases present a constant threat to agriculture, which is the essential basis for human subsistence and greatly affects agricultural output. Though widely used, traditional detection techniques are labor-intensive and prone to errors, therefore it's important to look into faster, more scalable, and more effective alternatives. With a major focus on Convolutional Neural Networks (CNNs) and MobileNet architectures, this research explores the revolutionary potential of Deep Learning (DL) models[32] for the accurate and early diagnosis of plant diseases. By showing these models' decision-making processes, the integration of eXplainable Artificial Intelligence (XAI) using GradCAM improves interpretability and provides insights into disease indications in plant images. After undergoing extensive testing, the CNN model shows an outstanding 89% accuracy rate, 96% precision and recall rates, and a 96% F1-score. However, the accuracy achieved by the MobileNet design is only 96%, while the precision, recall, and F1-scores are only marginally higher at 90%, 89%, and 89%, respectively. These findings highlight how DL is revolutionizing plant disease detection strategies and offer a strong substitute for established methods. By adding XAI, DL models become even more interpretable, boosting decision-making confidence and bringing in a new era of enhanced agricultural security due to better disease detection capabilities.

The architectures of several deep learning models are thoroughly discussed in the sections below, and their intricate details and guiding principles are clarified to demonstrate how effective they are at transforming the detection of plant diseases.

2.1.1 MobileNetV2

In 2018, MobileNetV2, a convolutional neural network architecture specifically designed for mobile and edge computing, was unveiled. It was an evolution of Google's original MobileNet. First presented in a research paper titled "MobileNetV2: Inverted Residuals and Linear Bottlenecks,"[10] the novel design approach of MobileNetV2 centers on maximizing the delicate balance between computational effectiveness and accuracy of models.The use of inverted residuals and linear bottlenecks, which combine lightweight depthwise separable convolutions and linear projections to drastically cut computational costs while maintaining the fundamental strength of representations, is its main innovation. Using 1x1 convolutions with linear activation functions as linear bottlenecks accelerates channel dimensionality prior to more complex processes, resulting in a smaller model size and improved computational performance. MobileNetV2, which is well-known for its scalable and modular architecture, enables flexibility in adjusting the model size to meet the resource limitations of various mobile and edge devices. With its ability to smoothly balance the requirements of computational economy and model performance, this innovative technique has driven MobileNetV2 into widespread usage and made it a cornerstone in mobile applications. The system is ideal for effective field-based disease tracking because it is built on top of MobileNetV2[37]. Figure 2.1 shows the architecture of MobileNetV2 which was developed by Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, as introduced in their research paper 'MobileNetV2: Inverted Residuals and Linear Bottlenecks' (2018).

$$Y = ReLU(Wp(ReLU(WdX))) \tag{2.1}$$

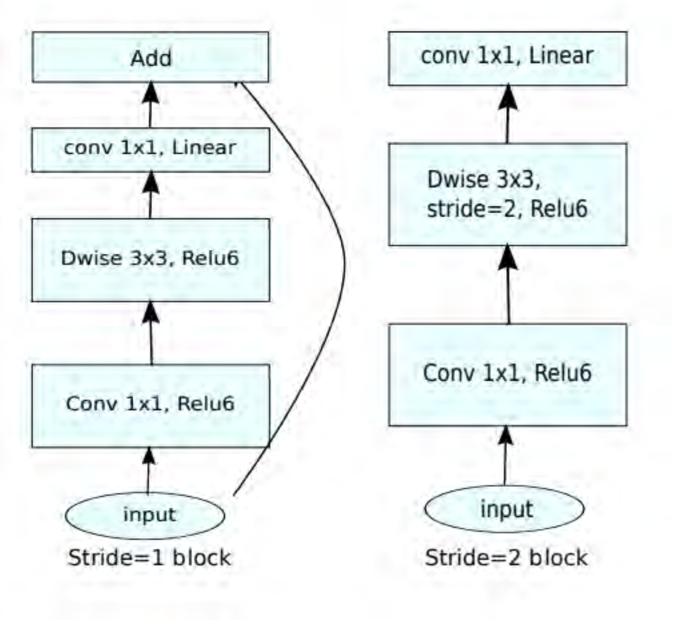

Figure 2.1: Layer Visualization of the MobileNetV2 model

2.1.2 DenseNet201

Being a member of the DenseNet family, DenseNet201 is a convolutional neural network architecture distinguished by its dense connectivity layout. DenseNet201, created by Gao Huang, Zhuang Liu, and Laurens van der Maaten, expands on the concept of strongly connecting each layer in a feedforward manner to every other layer[5]. In addition to improving gradient flow throughout the network and facilitating feature reuse, this architecture helps to mitigate the vanishing gradient problem. Every layer in DenseNet201 receives direct input from every layer that came before it, and every layer that came after it receives its output. DenseNet201 is especially useful for tasks involving big datasets because of its dense connection, which enhances parameter efficiency, model compactness, and accuracy. The connection structure and layer interactions are shown in the figure 2.2 below to help visualize the DenseNet201 design. The DenseNet201 architecture's authors, Gao Huang, Zhuang Liu, and Laurens van der Maaten, designed this architecture diagram.

$$Xl + 1 = Hl([X0, X1, X2, ..., Xl]) \tag{2.2}$$

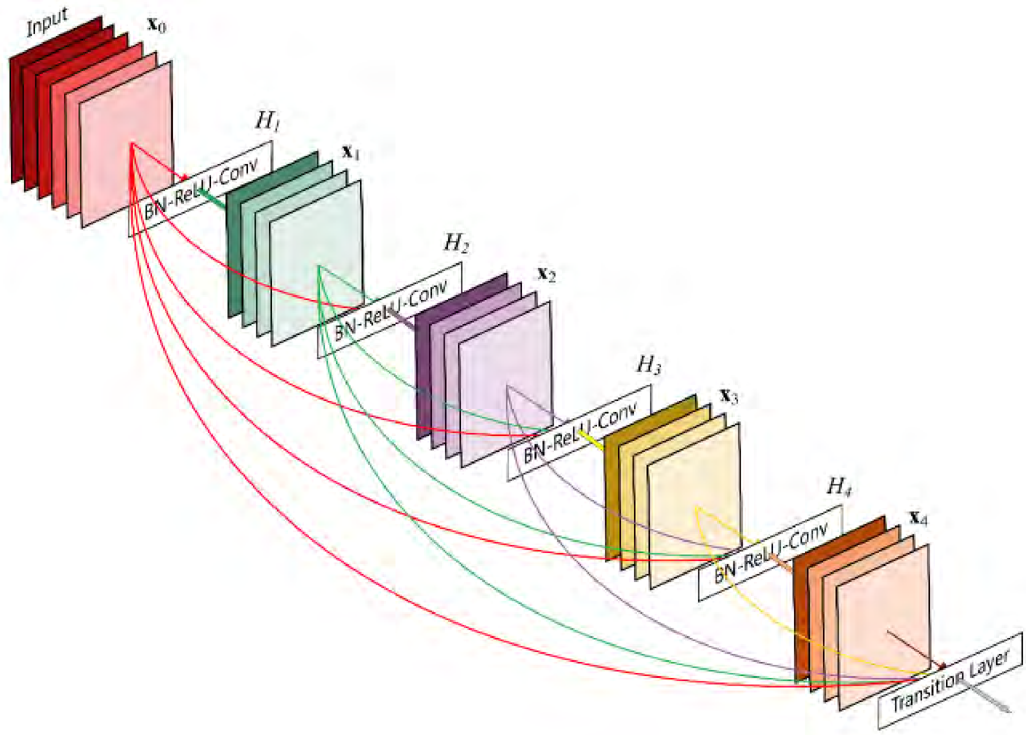

Figure 2.2: Layer Visualization of the DenseNet201 model

2.1.3 ResNet50

According to Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun's publication "Deep Residual Learning for Image Recognition" (2016)[4], ResNet50, which refers to Residual Network with 50 layers, is a deep convolutional neural network design. A member of the ResNet family, ResNet50 rose to prominence when it solved the vanishing gradient issue and improved the training of extremely deep neural networks.The inclusion of residual blocks, which provide skip or shortcut connections to the conventional convolutional layers, is one of ResNet50's key characteristics. The gradient can move through the network more directly thanks to these shortcuts, which makes training remarkably deep networks easier.ResNet50 primarily consists of 50 convolutional layers pooling and fully connected layers subsequently, in addition to convolutional, activation, and batch normalization layers. As a popular model in the computer vision society, the architecture achieves outstanding results on a variety of image classification tasks.
Xiangyu Zhang, Shaoqing Ren, Jian Sun, and Kaiming He developed ResNet50, a groundbreaking deep convolutional neural network design. They invented the idea of residual blocks and greatly enhanced the training of deep neural networks with their ground-breaking work on residual learning, which is described in the paper "Deep Residual Learning for Image Recognition" (2016). With its distinctive use of residual blocks for improved performance, ResNet50's architecture is shown in the figure 2.3 below.

$$y = F(x, (Wi)) + x \tag{2.3}$$

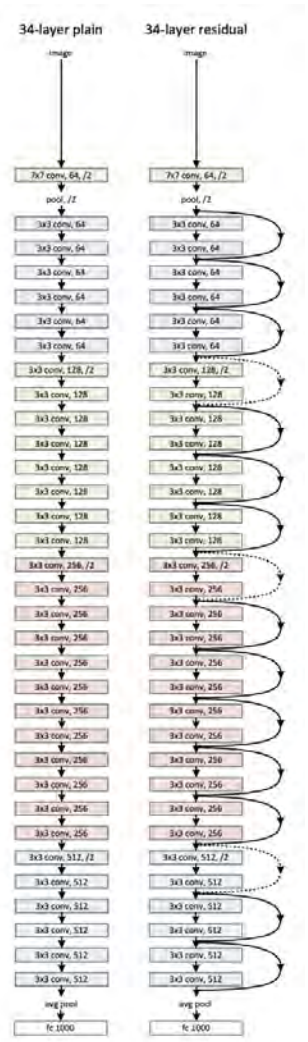

Figure 2.3: Layer Visualization of the ResNet50 model

2.1.4 Xception

Xception, which stands for "Extreme Inception," is an architecture for a deep neural network that uses depthwise separable convolutions. While it draws inspiration from the Inception architecture, depthwise separable convolutions are used in place of normal convolutions to produce a more parameter-efficient system. Depthwise separable convolutions are made up of a pointwise convolution (a 1x1 convolution to aggregate information across channels) after a depthwise convolution (a single filter applied to each input channel). This division keeps expressive power while lowering the number of parameters.
In the academic paper "Xception: Deep Learning with Depthwise Separable Convolutions[6], François Chollet presents the ground-breaking Xception deep neural network architecture (2017). The Xception architecture is shown in the figure 2.4 below with reference to the above author, which also highlights the creative way in which depthwise separable convolutions are used to improve computing efficiency.

$$y = a(DW(x)) * PW(a(x)) \tag{2.4}$$

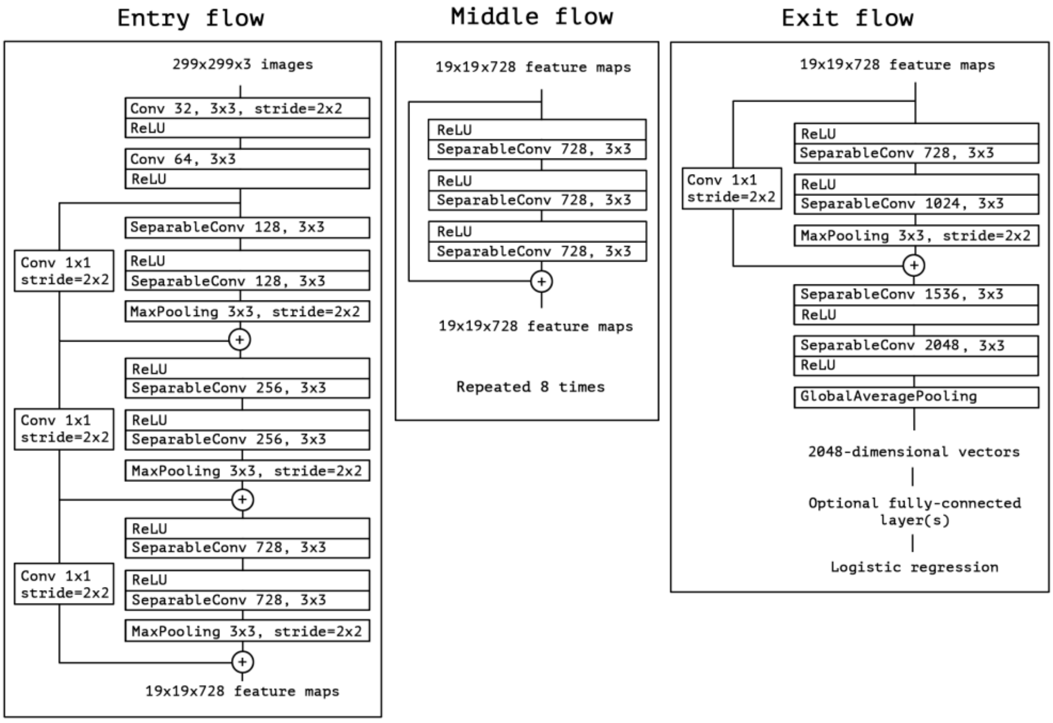

Figure 2.4: Layer Visualization of the Xception model

2.1.5 InceptionV3

The popular convolutional neural network architecture InceptionV3 was first presented by the research team of Google researchers Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke[3]. The aforementioned design is a development of InceptionV1, its predecessor, and it integrates new technologies to enhance model performance and training speed. The usage of inception modules, which are made up of several parallel convolutional procedures with different kernel sizes, by InceptionV3 is well recognized. The system can gather and interpret data at various spatial scales thanks to the use of a number of inception modules by InceptionV3. These modules include pooling techniques and convolutions of sizes 1x1, 3x3, and 5x5, which enable the extraction of various features. The design of InceptionV3 strikes a compromise between expressive power and computational efficiency. To enhance gradient flow, the architecture incorporates additional auxiliary classifiers and factorized 7x7 convolutions during training. When it comes to image classification challenges, InceptionV3 has performed better in general. The InceptionV3 architecture, which was introduced by Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke from Google Research, is shown in figure 3.5 below.

$$Inception(X) = [Concat(Branch1(X), Branch2(X), Branch3(X), Branch4(X))] \tag{2.5}$$

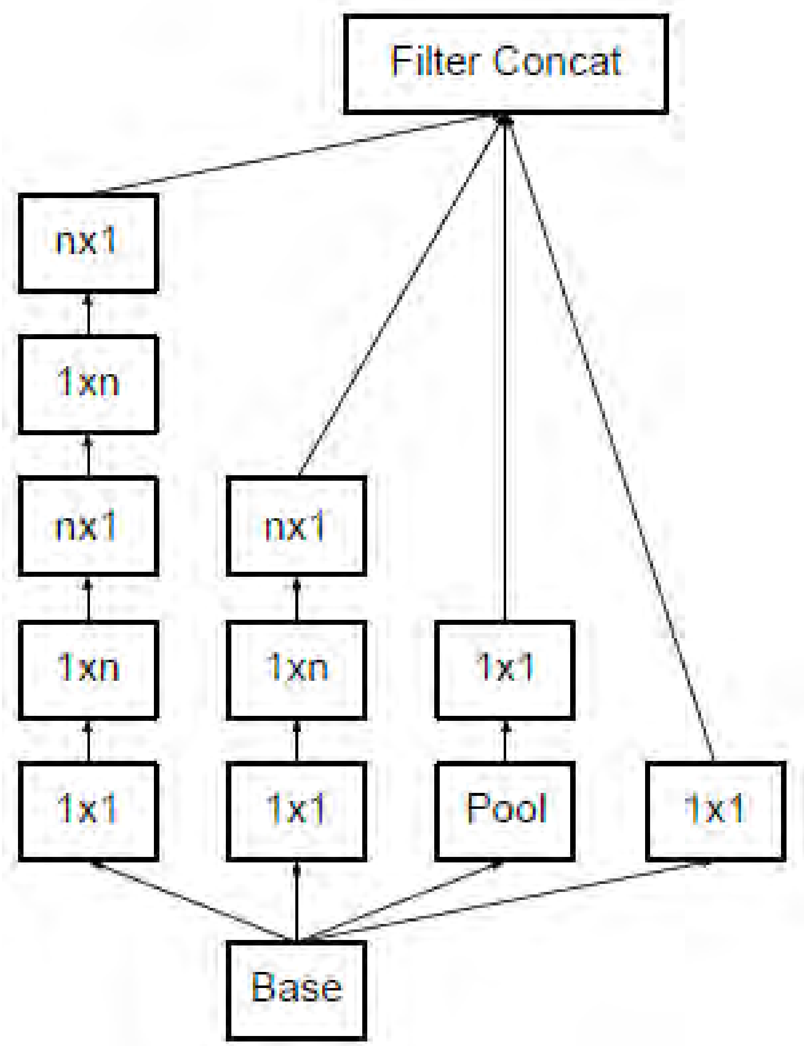

Figure 2.5: Layer Visualization of the InceptionV3 model

2.1.6 EfficientNet

The EfficientNet family of convolutional neural network designs was presented in the 2019 research paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" by Mingxing Tan and Quoc V. Le[14]. With the goal of increasing efficiency in terms of both model size and computational cost, EfficientNet is intended to offer a methodical and fundamental strategy for convolutional neural network scaling.The depth, width, and resolution of the network are all consistently scaled thanks to a compound scaling technique introduced by EfficientNet. To maximize model performance, these scaling factors should be balanced. The architecture starts with a baseline network and scales it up or down according to a compound coefficient that the user defines. Because of its ability to adapt efficiently to varying resource restrictions, EfficientNet is especially well-suited for a broad spectrum of devices and applications.EfficientNet has maintained effectiveness while achieving state-of-the-art performance on a variety of image classification tasks. Because the architecture strikes a compromise between accuracy and resource needs, it has been widely utilized for edge computing and mobile applications.

The architecture of EfficientNet, developed by Quoc V. Le and Mingxing Tan, is shown in Figure 2.6

$$CompoundCoefficient : n = b^C * p \tag{2.6}$$

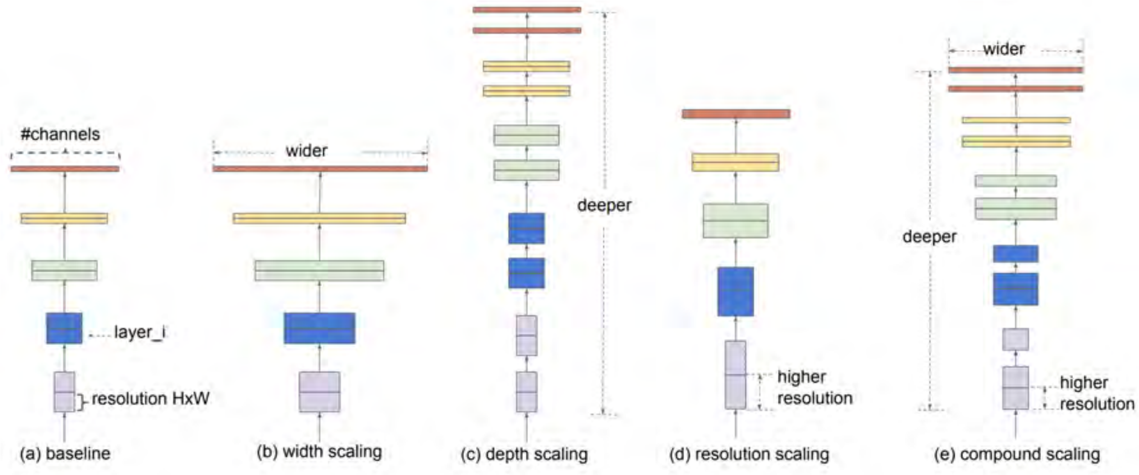

Figure 2.6: Layer Visualization of the EfficientNet model

2.1.7 VGG19

Introduced by the University of Oxford's Visual Geometry Group (VGG), VGG19 is a deep convolutional neural network design. The VGG16 architecture, which was first introduced in the 2014 publication "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman[2], is expanded upon herein. VGG19 has a consistent architecture and is defined by its simplicity. Its 19 layers include 3 fully linked layers and 16 convolutional layers. Max-pooling layers with 2x2 filters are used for spatial down-sampling, and the convolutional layers use tiny 3x3 filters with a stride of 1. The architecture is renowned for having a deep, uniform structure that makes it simple to comprehend and use. Even though VGG19 may not be as computationally efficient as other later architectures, it is nevertheless a popular choice and a reliable starting point for a variety of computer vision problems.
The figure 2.7 below illustrates an overview of the VGG19 architecture which was developed by the Visual Geometry Group at the University of Oxford.

$$Y = a(WX + b) \tag{2.7}$$

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.7: Layer Visualization of the VGG19 model

2.2 Related Works

For effective plant disease forecasting Sutaji and Yıldız suggested LEMOXINET[22], an ensemble model that combines Xception and MobileNetV2. Xception improves MobileNetV2, designed for mobile devices, to capture important aspects, leading to higher accuracy. Having an accuracy of 99.10% on the data set from Plant Village, the ensemble model outperformed each of the models. Relative to other designs, this method allows mobile incorporation with smaller sample files and variables and it outperforms seven current CNN models.

Aarizou and Merah presented a study at the 7th International Conference on Image and Signal Processing and its Utilization (ISPA) in 2022[20].Their research focused on using transfer learning to identify plant diseases in intricate picture data. The utilization of transfer learning approaches to improve the efficacy of infectious disease identification algorithms was investigated by the writers.

Utilizing several deep learning-powered already trained models—MobileNetV2, EfficientNetB6, and NasNet. Singh, Chug, and Singh looked at the categorization of illnesses plaguing bean leaves. The influence of different ways to optimize on various Convolutional Neural Network (CNN) models was investigated by scientists using transfer learning techniques[21] on a dataset of 1295 bean leaf images with three different classifications, encompassing bean rust and angular leaf spot disorders. EfficientNetB6 surpassed other models, according to the experimental investigation, which showed an accuracy of 91.74%. The work offers tangible uses of the optimal model for farmers, enabling timely preventive interventions and avoiding plant yield loss, in addition to shedding light on the roles played by various optimizers on CNN models. Growth in the economy and increased agricultural output could result from this.

Joel Kennedy, Joshua Alfred, and Karthik published a paper in Ecological Informatics that presented a novel method for detecting coffee leaf disease[23]. Understanding how important a precise diagnosis is to maintaining the health of coffee plants and producing high-quality beans, the researchers created a deep learning-based classification system. To accomplish accurate disease categorization, the suggested network combines global context, multi-head interest, and inception modules. Multiple-scale feature extraction is made possible by Inception modules, while high-level context-based data is obtained by channel attention in the General Context Block. Sophisticated pattern interactions are captured by the multiple focus heads module, which strengthens the input's interpretation. The suggested network beat previous models after being trained on the BRACOL dataset, with an astounding accuracy of 98.57% and an F1 score of 98.55%.
The academic publications that we analyzed, together with the models they used and their estimated accuracy, are listed in Table 2.1.

Ref	Author	CNN Model	Crop Type	Dataset	Accuracy
[15]	Abed, S. et al.	Dense-Net121	Beans	Manually collected Bean Leaves	98.31%
[27]	Vimal, A. et al.	Efficient-NetB6	Beans	Dataset of 1295 images	91.74%
[23]	Karthik, J. et al.	Inception modules	Coffee	BRACOL dataset	98.57%
[22]	Deni Sutaji, O. et al.	Mobile-NetV2	Various plants	Plant Village dataset	98.30%
[22]	Deni Sutaji, O. et al.	Xception	Various plants	Plant Village Dataset	99.10%
[19]	Sahu, P. et al.	VGG16	Beans	1296 Leaf Images	95.31%
[26]	Abebech, A. et al.	CNN-LSTM	Various plants	Many Datasets	92.55%
[34]	Marriam, T. et al.	CoffeeNet	Coffee	Arabica coffee leaf repository	98.54%
[38]	SERTTAŞ, S. et al.	ResNet50	Beans	1295 images	98.33%
[24]	Koklu, M. et al	Dense-Net201	Dry Beans	60 images	100%
[17]	ChowdhurY, M. et al.	Efficient-Net-B4	Tomatoes	18,161 images	99.89%
[18]	Rakesh, M. et al	VirLeaf-Net-1	Vigna mungo	Manually collected leaves	91.234%
[18]	Rakesh, M. et al.	VirLeaf-Net-2	Vigna mungo	Manually collected leaves	96.429%
[18]	Rakesh, M. et al.	VirLeaf-Net-3	Vigna mungo	Manually collected leaves	97.403%
[36]	Vivek, A. et al	ClGanNet	Maize	Real Field Images	99.97%

Table 2.1: Research papers on Disease Detection of Various crops using Deep Learning

While our work builds on previous research, it stands out by addressing issues that other studies have frequently ignored. Although prior studies mostly used pre-trained models, they seldom took the time to thoroughly analyze the complex structures that underlie these models. Our work, on the other hand, attempts to close this gap by carefully examining the architectures and illuminating the intricacies that affect the models' functionality such as the number of layers and kernel sizes of the models. We are dedicated to analyzing the complexities of these deep learning models, which will enable a better comprehension and more accurate application in the context of bean disease detection. Our work goes beyond simple application of the models.

2.3 Chapter Summary

To sum-up, this chapter, which focuses on CNN models and deep learning algorithms, offers an in-depth review of previous research in the field of agricultural disease identification. Although previous studies have achieved great progress using pre-trained models, our work aims to stand out by using these models together with a more in-depth analysis of their architectures. This strategy is expected to fill important gaps in the literature by offering fresh perspectives and improving the effectiveness of deep learning models in the context of prompt and precise bean disease diagnosis.

Chapter 3

Dataset

For our study, we used a large dataset that included 5,000 initial pictures that were then further enhanced by augmentation to produce a total of 10,000 images. Training powerful and effective deep learning models requires a wide and representative dataset, which is made possible by the augmentation process. With its large image collection, this dataset was made publically available in order to further research and applications in the areas of agricultural optimization and efficient bean crop disease control. Easily obtainable via the Makerere University Artificial Intelligence Laboratory Repository(published on July 20, 2022). Researchers and practitioners interested in investigating novel approaches for the identification of bean rust and angular leaf spot in beans can use this invaluable resource . With healthy bean leaf pictures acting as a control experiment, our models identify patterns and variances in pixel values linked to distinct classes during the learning phase. The dataset is essential for training our algorithms to correctly detect and classify images as having bean rust, angular leaf spot, or being healthy.Figures 3.1 and 3.2 show a few examples of images taken from the used dataset before and after augmentation respectively.

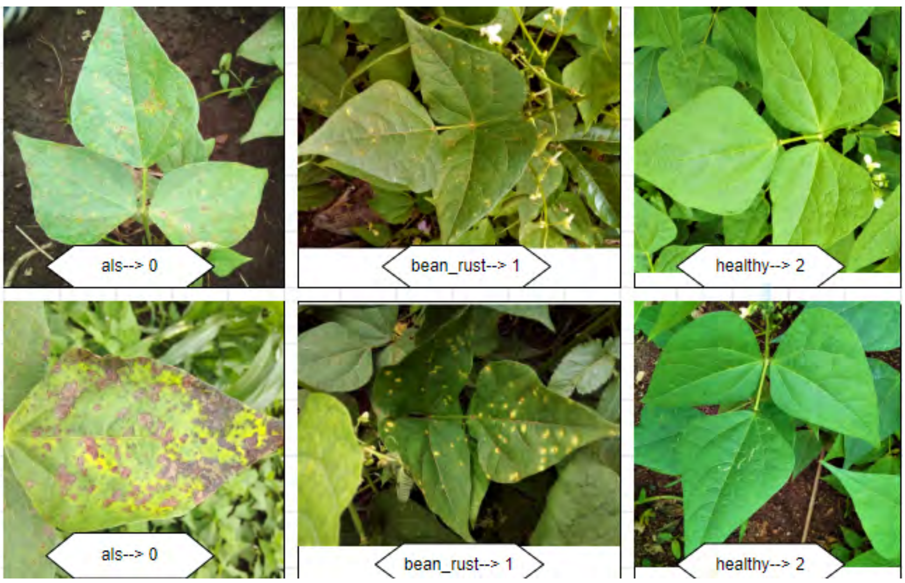

Figure 3.1: Image samples before Augmentation

Figure 3.2: Image samples after Augmentation

3.1 Data pre-processing

The 5,000 images in the dataset are split into three classes; angular leaf spot, healthy and bean rust images and data augmentation is achieved through random transformations and perturbations, encompasses a range of techniques aimed at generating "new" training samples from existing ones, all while preserving the original class labels. The primary objective of data augmentation is to enhance the model's ability to generalize. By introducing slight modifications to input images without altering their class labels, data augmentation increases the number of images to 10,000 hence it proves to be a logical and straightforward approach for image processing tasks. Although various data augmentation techniques are commonly employed to enhance model performance, in our case, we specifically employ the "Dataset generation and expansion" approach to address the highlighted problem and provide a solution. The key objective is to maintain a reasonable ratio (train, test, validate), and this entails increasing the number of diseased images during the augmentation process. The workflow follows a series of steps to accomplish this task. Initially, all input images are loaded from the disk. Subsequently, the ImageDataGenerator is applied to augment the input data. The class is instantiated, and various data augmentation parameters are customized within the class constructor. Multiple strategies, along with pixel zooming techniques, are then implemented. After manually processing each image, an array is constructed, which is subsequently fed into the datagen object using the flow method to handle these numerous images efficiently. Throughout training, these enhancements increased the dataset with every epoch in addition to adding new variants to the pictures. The table 3.1 below summarises the preprocessing techniques that were applied to the dataset.

3.2 Data Splitting

To ensure accuracy and appropriate image dimensions, the input data is separated into three categories (training, testing, and validation) after it has been enhanced. 80% is allocated to training, 10% is allocated to validation and 10% to testing, as shown in figure 3.2 below.

Model compilation	Model optimizer='Adam' Loss Method='binary-Cross Entropy'
Iteration set	EPOCHS=50
Data Augmentation	Preprocessing_function = preprocessing_input, rotation_range = 40, width_shift_range = 0.2, height_shift_range = 0.2, shear_range = 0.2 zoom_range = 0.2 horizontal_flip=True

Table 3.1: Procedure for preprocessing

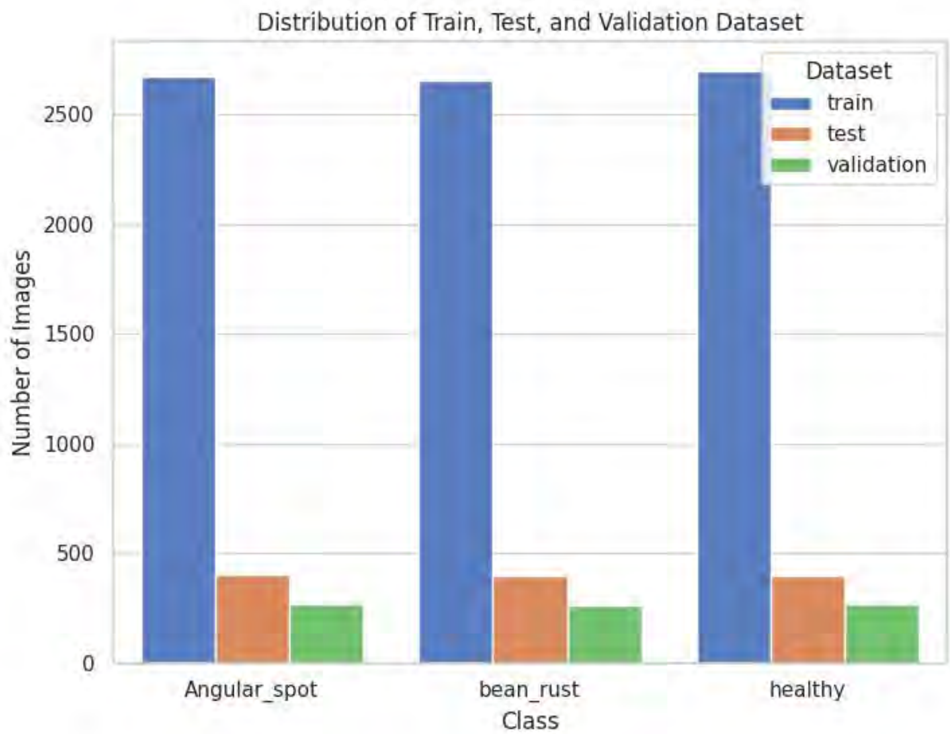

Figure 3.3: Dataset Distribution

As seen in Table 3.2, the dataset has been grouped for greater efficiency while training and evaluating the models. 8020 samples are included in the training stage, while 1202 and 802 samples are included in the testing and validation stages, respectively. By organizing the processes in portions, the model during training and evaluation processes become more computationally efficient. In order to guarantee that the Convolutional Neural Network algorithms are reliable and competent of correctly classifying the bean leaf classes, the methodical dataset segmentation combined with layered splits provides a solid basis for the models' testing and training.

Class	Train	Test	Validation
Angular Leaf Spot	2669	404	268
Bean Rust	2655	398	266
Healthy Leaves	2696	400	268
Total	8020	1202	802

Table 3.2: Summary of the Dataset

3.3 Chapter Summary

We explored the details of our dataset acquisition in this extensive chapter, offering explanations for its source and accessibility via a public repository. Stressing the importance of having a large and varied image collection, we described the preprocessing techniques we used, such as the augmentation process that increased the size of our dataset to 10,000 photos. We also provide insights into the careful dataset separation methods used to guarantee successful model training. In search of efficient bean crop identification and agricultural productivity, this chapter provides a foundational overview, outlining the key elements that form the basis for our more advanced deep learning approaches.

Chapter 4

Methodology

This section outlines the thorough approach that reinforces our suggested technique for the identification and diagnosis of bean rust and angular leaf spot in beans. Our approach combines modern deep learning methods to create a strong model that can recognize subtle signs that are symptomatic of these prevalent bean infections. The emphasis is on systematic and practical considerations, describing the techniques and tactics used to ensure precise and effective disease diagnosis.

4.1 Deep Learning Workflow

Following the image augmentation from chapter 3, the goal is to enhance the model ability to generalize to new and unknown images. Training robust and flexible models depends critically on this augmentation process therefore the resulting augmented image dataset is carefully split using predetermined split ratios of 80%, 10%, and 10% for training, testing, and validation sets, respectively. Through the provision of discrete datasets for training, testing, and validation, this methodological division guarantees a neutral assessment of the models' performance. Within the framework of the current classification problem, the input data is divided into three separate classes: angular leaf spot, bean rust, and healthy. Five advanced neural network designs are used to perform the classification task: ResNet50, InceptionV3, DenseNet201, Xception, and VGG19. After evaluating each model's distinct architectural strengths, each is chosen based on its proven performance in picture classification tasks. Utilizing the updated datasets, the neural network models undergo extensive training, testing, and validation processes. Vital parameters including batch size, image height, and width are closely monitored during this procedure. For accurate predictions to be made and for the models to be optimized for the given dataset, these parameters are essential. In order to enhance the models, a thorough fine-tuning procedure is employed. The process of fine-tuning entails modifying the models to optimize their performance on the particular dataset, taking into account aspects like batch size, image width, and height. The iterative process of fine-tuning guarantees that the models develop to comprehend complex patterns and the details present in the dataset, hence improving their overall performance. The empirical results highlight the effectiveness of using a variety of neural network models—DenseNet201, ResNet50, InceptionV3, VGG19, and Xception—in conjunction with reliable data augmentation methods. In picture classification tasks, the combination of these models and augmentation procedures works wonders to achieve high precision. This all-encompassing strategy, which includes a variety of neural network topologies as well as careful parameter optimization, validates the models' ability to correctly classify images into the assigned categories. The figure below represents the proposed deep learning workflow.

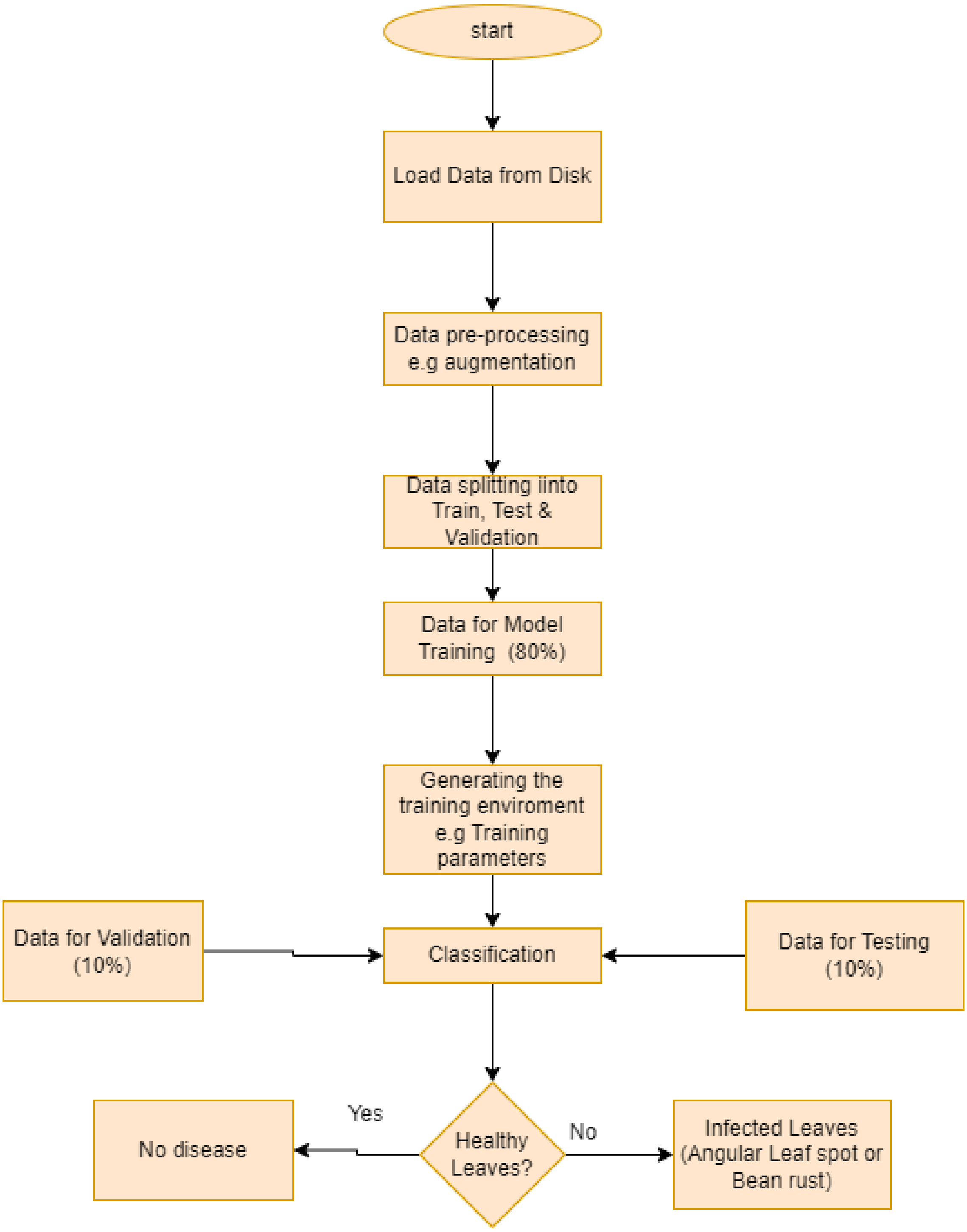

Figure 4.1: Proposed Deep Learning workflow

4.2 Working Plan

With the assistance of our thesis instructor, we were able to locate a suitable dataset after examining numerous datasets. As seen in Figure 2, that dataset includes leaves with bean rust, angular leaf spot, and healthy leaves. Next, in order to increase the diversity of the dataset and improve the model's capacity to apply its expertise to new or unfamiliar material, we employed pre-processing techniques including data augmentation. The model was able to learn broader features from many versions of the original photos by applying techniques such as rotation, scaling, and flipping to create novel variants of the initial images.

We also read through a number of research articles concerning the outcomes of deep learning algorithms. Then, because to their excellent performance, we selected Xception, Inception-v3, VGG19, DenseNet201, and ResNet50 for our transfer learning model. DenseNet201 was chosen because of its improved feature and reuse capabilities, which are made possible by its dense connections, which also make it possible to employ features from earlier layers more effectively. It provides parameter efficiency and captures fine-grained information, demonstrating promising outcomes in tasks of a similar nature. VGG-19 was selected due to its ease of interpretation and simplicity. Its simple architecture, consisting of stacked pooling and convolutional layers, facilitates comprehension and analysis. VGG-19 is a popular choice for a variety of image classification problems due to its strong generalization capabilities.

ResNet50 was picked for its deep architecture and the inclusion of residual connections to the network. By reducing the vanishing gradient issue, these connections make it possible to train deeper networks successfully. Strong performance, excellent accuracy, and suitability for transfer learning using pre-trained models have all been demonstrated by ResNet50. Because of their creative inception modules—which use several filter sizes within a single layer—InceptionV3 and Xception were selected. This allows the network to handle both large-scale and fine-grained patterns in the input, allowing it to capture characteristics at multiple scales. In comparison to other deep architectures, the inception modules reduce the number of parameters, allowing for a more effective use of computational resources.

4.3 Transfer Learning

In the field of deep learning, transfer learning is a very useful method that has shown to be very successful in agricultural image analysis, including the task of crop disease identification. In this case, transfer learning is using pre-existing models that were previously trained on large datasets, such as millions of annotated photos in different categories found in ImageNet. These pre-trained models are able to capture crucial visual patterns because they have acquired complex and discriminative characteristics from their training data.

There are various advantages of transfer learning. Initially, pretrained models have previously been trained on big datasets like ImageNet, which comprise a variety of photos from different categories. Furthermore, the problem of insufficient training data is mitigated via transfer learning. Deep learning models may not be readily accessible for training when an extensive amount of labeled data is needed, such as for specific tasks like bean leaf disease identification. Using pretrained models to start, we can refine our learned representations on the comparatively smaller image dataset. By applying its learned characteristics to the current task, the model can perform better by employing this method.

We took advantage of the information and representations discovered from extensive image datasets by applying pretrained models like DenseNet201, Inception-v3, Xception, VGG19, and ResNet50. The approach that has been presented involves initializing the pretrained models with their parameters and fine-tuning them by freezing every layer until the final one. The last layer is particularly trained for the bean disease detection task, while the first several layers are frozen to guarantee that the models maintain their learnt representations and do not change during training. By substituting an alternative configuration for the last layer of the models, we intended to improve its suitability for our objective of identifying angular leaf spot and bean rust.

4.3.1 DenseNet-201

DenseNet201 is very well suited for bean leaf disease detection, as its architecture (Chapter 2.1.2) illustrates. Its architecture's densely connected blocks facilitate efficient information flow and feature propagation, which is useful for detecting little disease-related trends in bean plants. The complex spatial associations that the densely connected layers demonstrate are in line with the subtle ways that different bean diseases present themselves. Moreover, DenseNet201's compact and parameter-efficient architecture, as seen in Figure 2.2 in particular, improves its capacity to identify intricate disease patterns while maximizing computational resources. DenseNet201 is an outstanding choice for precise and resource-efficient bean disease diagnosis because of its structure, which makes effective use of gradient flow and feature reuse possible.

4.3.2 ResNet-50

ResNet50 provides significant benefits for bean leaf disease detection, as its unique architecture is described in Chapter 2.1.3. Residual connections help information go through the network more easily, which mitigates the problem of vanishing gradients and makes it possible to successfully understand complex patterns linked to different bean diseases. As seen in Figure 2.3, ResNet50's deeper architecture enables the model to capture hierarchical features, which are crucial for differentiating intricate disease presentations. Furthermore, the residual blocks support the stability of the model, which helps the model to manage the variation in disease manifestations among various bean plants. ResNet50 is an ideal choice for reliable and accurate bean disease detection because of its deep representation abilities and robust architecture, particularly in situations where diagnosis depends on hierarchical and complex patterns.

4.3.3 VGG-19

As discussed in Chapter 2.1.7, VGG19 offers a simple yet efficient architecture for the identification of bean leaf diseases. An organized method to feature extraction is provided by the uniformity and simplicity of the VGG19 design, which consists of 19 layers, including numerous convolutional layers and fully connected layers. The use of tiny 3x3 convolutional filters makes it easier to capture localized patterns, which is important for identifying distinct bean leaf indications. Moreover, the recognition of disease-related features by the model is improved at different scales due to the spatial down-sampling that is accomplished by max-pooling layers. VGG19 is appropriate for situations where interpretability and a hierarchical understanding of disease traits are crucial because of its distinct layer structure, which is shown in Figure 2.7. The simplicity and versatility of VGG19 may be useful for efficient and transparent bean disease detection, even with its slightly higher computational cost. This is especially true in situations where a clear hierarchy of characteristics is essential for diagnosis.

4.3.4 Inception-V3

With its complex and multi-scale architecture, InceptionV3, as explained in Chapter 2.1.5, is a desirable choice for bean leaf disease detection. The model can effectively capture features at numerous spatial scales because of the inception modules, which perform concurrent convolutional operations with different kernel sizes. This is especially helpful in identifying various bean disease presentations, as signs might differ in complexity and magnitude. As illustrated in Chapter 2.1.5, the factorised 7*7 convolutions and auxiliary classifiers improve training stability and gradient flow, which are essential for identifying subtle illness patterns. Bean leaf diseases are complicated and varied, but InceptionV3 is capable of managing them thanks to its flexibility and adaptability, which are demonstrated in its architecture in Figure 2.5. Because of its capacity to handle data at several scales concurrently, InceptionV3 is presented as a reliable option for thorough and accurate bean leaf disease diagnosis.

4.3.5 Xception

The detection of bean leaf diseases can benefit from Xception's distinct and effective architecture, which is covered in depth in Chapter 2.1.4. The neural network can record intricate spatial hierarchies and patterns within bean leaves since it makes use of depthwise separable convolutions. Because of the factorized convolutions, Xception can be used in situations where processing power is scarce. Furthermore, as described in Chapter 2.1.4, Xception's fine-grained detail capability is useful for identifying small visual clues linked to a variety of bean leaf disorders. Figure 2.4's architecture, which demonstrates the novel approach to the model creation, makes Xception a viable option for precise and resource-efficient bean disease diagnosis, especially when lightweight feature and computational efficiency are crucial.

The architectural features of the Transfer Learning Models are displayed in Table 2.

Model Architecture	Number of Layers	Filter Sizes(Examples)
DenseNet201	201	1*1, 3*3, 5*5
Xception	Varied(depthwise separable convolutions)	Varied, relies on depthwise separable convolutions
ResNet50	50	1*1, 3*3
VGG19	19	3*3
InceptionV3	Varied	Varied

Table 4.1: Architectural characteristics of the Transfer Learning models

4.4 Chapter Summary

The deep learning methodology utilized for the prediction algorithms was thoroughly described in this chapter. We offered a thorough working plan that outlined the several stages we took while conducting our research. In order to improve the effectiveness of our leaf image disease prediction mechanism, we also concentrated on transfer learning models and used pre-trained neural networks. We talked about the selection of transfer learning models and explained the models we trained on our dataset. The objective of the chapter was to guarantee transparency and consistency in our study by elucidating the methodical approach that was followed.

Chapter 5

Deep Learning Implementation and Results

5.1 Implementation

To choose the algorithm that performs best in identifying angular leaf spot and bean rust in the bean image dataset, the process of implementation involves examining the five pre-trained(ResNet50, XCception, VGG19, DenseNet201 and InceptionV3) models' performances. Following the path specification, the model is initialized and its state dictionary is loaded. A DataLoader object is utilized to load the benchmark dataset. We use the eval() method to set the model to evaluation mode. On the basis of the test images, it then creates predictions as it goes through the test loader. The confusion matrix and classification report are generated using the features of scikit-learn and the NumPy library once the predicted labels and true labels have been gathered. All stages of implementation, testing, and validation involved the use of Keras deep learning classifiers By showing the proportion of accurate and inaccurate predictions, the confusion matrix heat map offers insights into the model's performance and allows for a thorough knowledge of the model's accuracy and loss rates. In order to give a thorough overview of the model's assessment metrics, the classification report is printed and contains measures like precision, recall, F1-score, and support. The model's learning progress is visualized by plotting the accuracy and loss curves for training and validation against the number of epochs.To show the results in terms of accuracy and loss, we used the MATLAB software suite.
The DESKTOP-5PJD0IA with an Intel(R) Core(TM) i7-7700 CPU @3.60GHz and 16.0GB RAM is used for all experiments. It runs Windows 10 pro Single Language.

5.2 Model Results

5.2.1 ResNet50

Using input data, the ResNet50 model achieves 95% accuracy after 50 epochs. The confusion matrix for the model has been created following the execution of all training parameters. The number of occurrences that were accurately classified as positive for each of the labels is known as True Positive (TP). In this instance, it was determined that 596, 597, and 682 cases had angular leaf spot, bean rust, and healthy, respectively. The number of cases that were accurately classified as negative

is known as True Negative (TN), and in a multi-class classification, it is determined by adding the values of all columns and rows other than the class for which we are calculating the values. In this case, it is 684,687 and 1269 for Angular leaf spot, Bean rust and healthy respectively. The number of cases that were incorrectly reported as positive is known as False Positive (FP), and it is equal to the total of the values in the relevant column with the exception of the TP value. The incorrectly identified instances for angular leaf spot, Bean rust and healthy were 69,58 and 3 respectively. The number of instances that were incorrectly listed as negative is known as the False Negative (FN) and is determined by adding the values of all related rows, excluding the TP value. They were 41,38 and 51 for Angular leaf spot, Bean rust and healthy respectively.
This study's training and validation accuracy graphs offer a thorough analysis of the model's performance and assist in identifying possible problems like overfitting or underfitting.
Figure 5.1 shows a graphic illustration of the accuracy and loss(b) for the ResNet50 model as well as the visual representation of the heatmapx(a).

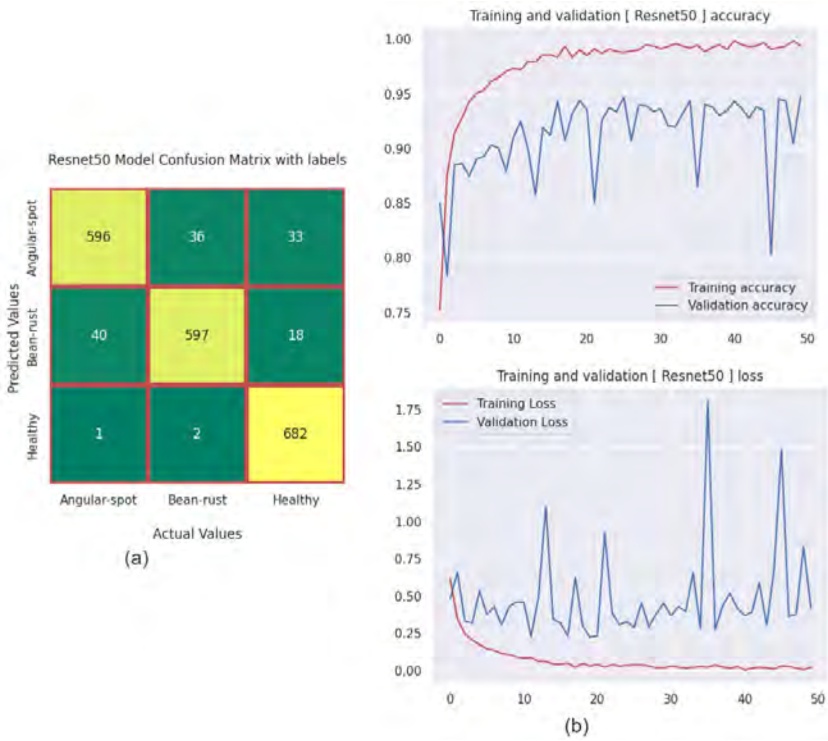

Figure 5.1: (a)Heatmap Visualization (b)Accuracy and Loss plots for ResNet50

Classification Report

Table 5.1 below displays the classification report, which summarizes this model's performance on the test data set for which the true values are known. Precision, recall, F1-score, and support are the measures that are employed and displayed.

	Precision	Recall	f1-score	Support
Angular-spot	0.95	0.93	0.94	665
Bean-rust	0.96	0.93	0.95	655
Healthy	0.94	0.98	0.96	685
Accuracy			0.95	2005
macro avg	0.95	0.95	0.95	2005
weighted avg	0.95	0.95	0.95	2005

Table 5.1: Classification Report for ResNet50

5.2.2 DenseNet201

The DenseNet201 model demonstrated 94% accuracy for comparison analysis after 50 epochs using input data. This shows that the model is gaining knowledge from the training set and is capable of making good generalizations to newly developed, untested data. After executing every training setting for that model, the confusion matrix has been created. The number of occurrences in each prediction category is shown by the values in the confusion matrix(a). The number of occurrences that were accurately classified as positive for each of the labels is known as True Positive (TP). In this instance, it was determined that 596, 597, and 682 cases had angular leaf spot, bean rust, and healthy, respectively. The number of cases that were accurately classified as negative is known as True Negative (TN), and in a multi-class classification, it is determined by adding the values of all columns and rows other than the class for which we are calculating the values. In this case, it is 684,687 and 1269 for Angular leaf spot, Bean rust and healthy respectively. The number of cases that were incorrectly reported as positive is known as False Positive (FP), and it is equal to the total of the values in the relevant column with the exception of the TP value. The incorrectly identified instances for angular leaf spot, Bean rust and healthy were 69,58 and 3 respectively. The number of instances that were incorrectly listed as negative is known as the False Negative (FN) and is determined by adding the values of all related rows, excluding the TP value. They were 41,38 and 51 for Angular leaf spot, Bean rust and healthy respectively.

To create predictions, the DenseNet201 model focuses on specific portions of the image, as shown in Heatmap(a) and plots(b). Knowing where the model is making conclusions and identifying possible areas for improvement may both benefit from this. Furthermore, a simple way to compare the model's performance is to look at Figure 5.2, which shows the accuracy graphically.

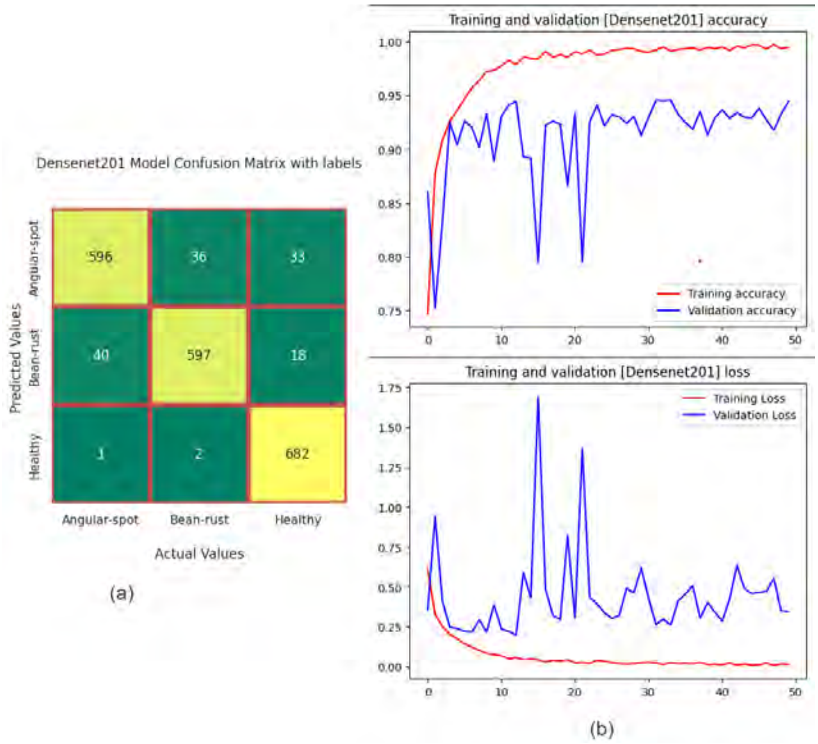

Figure 5.2: (a)Heatmap Visualization (b)Accuracy and Loss plots for DenseNet201

Classification Report

Table 5.2 below displays the classification report, which summarizes this model's performance on the test data set for which the true values are known. Precision, recall, F1-score, and support are the measures that are employed and displayed.

	Precision	Recall	f1-score	Support
Angular-spot	0.94	0.90	0.92	665
Bean-rust	0.94	0.91	0.93	655
Healthy	0.93	1.00	0.96	685
Accuracy			0.94	2005
macro avg	0.94	0.93	0.93	2005
weighted avg	0.94	0.94	0.93	2005

Table 5.2: Classification Report for DenseNet201

5.2.3 Xception

The Xception model demonstrated 96% accuracy for comparison analysis after 50 epochs using input data. This shows that the model is gaining knowledge from the training set and is capable of making good generalizations to newly developed, untested data. After executing every training setting for that model, the confusion matrix has been created. The number of occurrences in each prediction category is shown by the values in the confusion matrix. The number of occurrences that were accurately classified as positive for each of the labels is known as True Positive (TP). In this instance, it was determined that 640, 614, and 673 cases had angular leaf spot, bean rust, and healthy, respectively. The number of cases that were accurately classified as negative is known as True Negative (TN), and in a multi-class classification, it is determined by adding the values of all columns and rows other than the class for which we are calculating the values. In this case, it is 647,695 and 1298 for Angular leaf spot, Bean rust and healthy respectively. The number of cases that were incorrectly reported as positive is known as False Positive (FP), and it is equal to the total of the values in the relevant column with the exception of the TP value. The incorrectly identified instances for angular leaf spot, Bean rust and healthy were 25,41 and 12 respectively. The number of instances that were incorrectly listed as negative is known as the False Negative (FN) and is determined by adding the values of all related rows, excluding the TP value. They were 42,14 and 22 for Angular leaf spot, Bean rust and healthy respectively.

To create predictions, the Xception model focuses on specific portions of the image, as shown in the Heatmap(a) and plots(b). Knowing where the model is making conclusions and identifying possible areas for improvement may both benefit from this. Furthermore, a simple way to compare the model's performance is to look at graphs(b), which shows the accuracy and loss graphically.

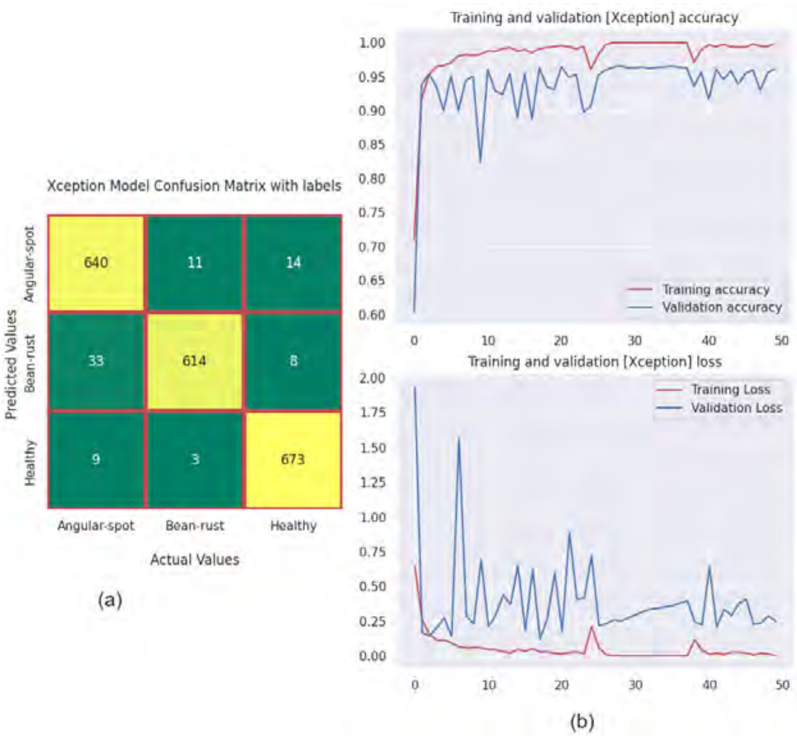

Figure 5.3: (a)Heatmap Visualization (b)Accuracy and Loss plots for Xception

Classification Report

Table 5.3 below displays the classification report, which summarizes this model's performance on the test data set for which the true values are known. Precision, recall, F1-score, and support are the measures that are employed and displayed.

	Precision	Recall	f1-score	Support
Angular-spot	0.94	0.96	0.95	665
Bean-rust	0.98	0.94	0.96	655
Healthy	0.97	0.98	0.98	685
Accuracy			0.96	2005
macro avg	0.96	0.96	0.96	2005
weighted avg	0.96	0.96	0.96	2005

Table 5.3: Classification Report for Xception

5.2.4 InceptionV3

The accuracy of the InceptionV3 model after 50 epochs of input data is 88%. The classification and identification results of the model are displayed in a heatmap of the confusion matrix. Plots representing the model's accuracy throughout training and validation are visualized.

Figure 5.4 shows the visual presentation of the accuracy and loss plots(b) for the InceptionV3 model as well as the visual representation of the heatmap(a).

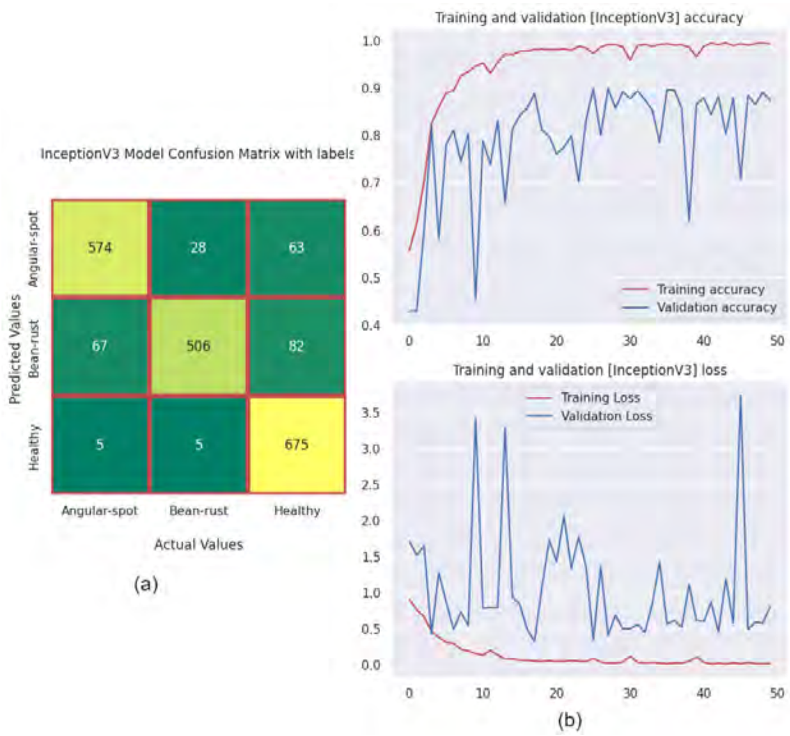

Figure 5.4: (a)Heatmap Visualization (b)Accuracy and Loss plots for InceptionV3

Classification Report

Table 5.4 below displays the classification report, which summarizes this model's performance on the test data set for which the true values are known. Precision, recall, F1-score, and support are the measures that are employed and displayed.

	Precision	Recall	f1-score	Support
Angular-spot	0.89	0.86	0.88	665
Bean-rust	0.94	0.77	0.85	655
Healthy	0.82	0.99	0.90	685
Accuracy			0.88	2005
macro avg	0.88	0.87	0.87	2005
weighted avg	0.88	0.87	0.87	2005

Table 5.4: Classification Report for InceptionV3

5.2.5 VGG19

The accuracy of the VGG19 model after 50 epochs of input data is 33%. The classification and identification results of the model are displayed in a heatmap of the confusion matrix. Plots representing the model's accuracy throughout training and validation are visualized.

Figure 5.5 shows the visual representation of the accuracy and loss plots(b) for the VGG19 model as well as the visual representation of the heatmap(a).

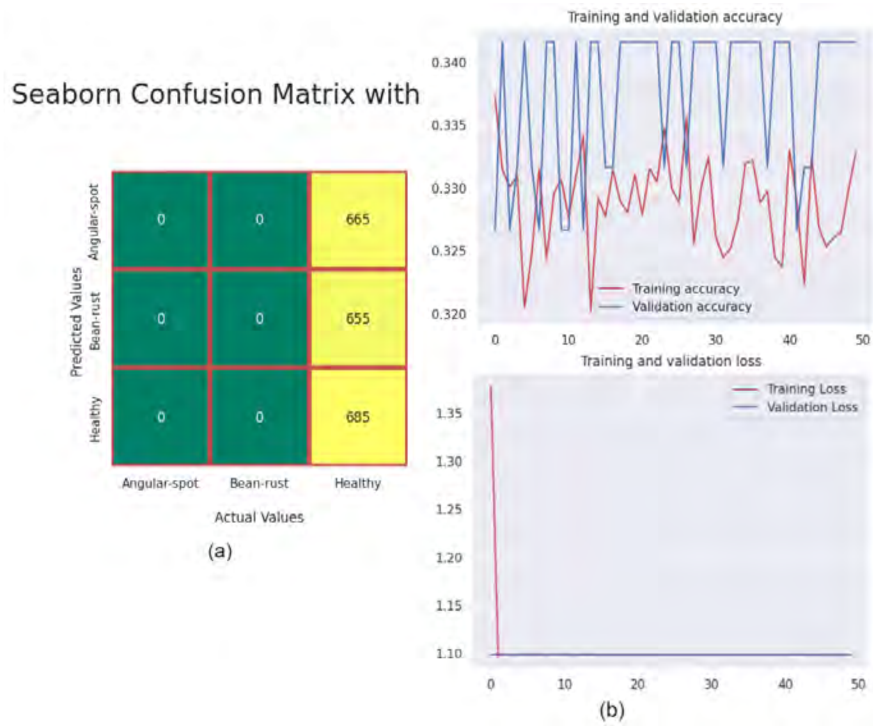

Figure 5.5: (a)Heatmap Visualization (b)Accuracy and Loss plots for VGG19

Classification Report

Table 5.5 below displays the classification report, which summarizes this model's performance on the test data set for which the true values are known. Precision, recall, F1-score, and support are the measures that are employed and displayed.

	Precision	Recall	f1-score	Support
Angular-spot	0.33	1.00	0.50	665
Bean-rust	0.00	0.00	0.00	655
Healthy	0.00	0.00	0.00	685
Accuracy			0.33	2005
macro avg	0.11	0.33	0.17	2005
weighted avg	0.11	0.33	0.17	2005

Table 5.5: Classification Report for VGG19

5.3 Comparison and Analysis

Given an accuracy of 96% after 50 epochs with the input dataset, the Xception model output the best results. The accuracy of the ResNet50 model was 95%, that of DenseNet201 was 94%, that of InceptionV3 was 88%, and that of VGG19 was 33%. In general, the Xception model outperformed the other four models, showing that it is the best model overall. Due to its cutting-edge performance on image classification metrics and its versatility in handling an extensive variety of image sizes and scales from the other four models, the Xception model proved to be an especially powerful and appropriate model for this purpose. Based on photos of the bean leaves, this implies that the model can learn from the input information efficiently and forecast the health state of the beans.
The Xception model's modular architecture and application of picture augmentation techniques like various activation functions and fine-tuning probably helped the model do well on this assignment. Additionally, Wider kernels, intricate structures, and greater non-linearity allowed ResNet50 and InceptionV3 to perform well. Whereas InceptionV3's varied filter widths and effective feature recycling enhanced its learning capabilities, ResNet50's disconnected connections maintained essential data. This demonstrates how crucial model selection is for particular tasks like bean rust and angular leaf spot identification.

However, the VGG19 model's accuracy drastically declined when it came to identifying individual instances of angular leaf spot and bean rust in the confusion matrix. This shows that it tripped over these particular diseases, largely because of its weak irregularities that struggle to understand complex symptomatic correlations and its narrow development that misses important information. Xception performed exceptionally well in these domains, demonstrating the influence of model selection on disease diagnosis, thanks to its wider kernels and increased non-linearity. In addition to highlighting the need for focused research on difficult diseases, VGG19's difficulties provide insight into the development of specific models for reliable bean disease categorization.
The model comparison is displayed in the table 5.6 and figure 5.6 below.

Classification Algorithm	f1-score	Precision	Recall	Accuracy
Xception	0.96	0.96	0.96	96%
ResNet50	0.95	0.95	0.95	95%
DenseNet201	0.93	0.94	0.94	94%
InceptionV3	0.87	0.88	0.87	88%
VGG19	0.17	0.11	0.33	33%

Table 5.6: A Comparison Table of the Models' Performances.

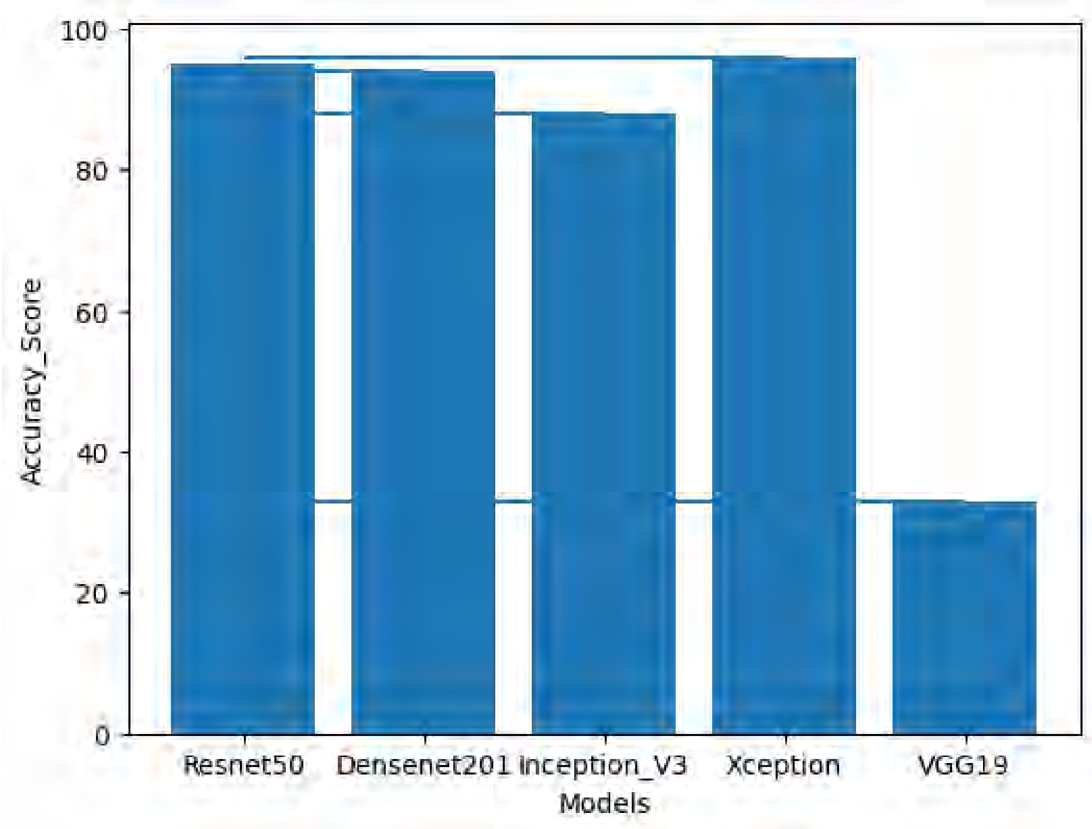

Figure 5.6: Histogram showing a comparison of the model accuracies

5.4 Visualization

Although classifying bean diseases with remarkable accuracy is obviously important, it's also important to have more knowledge of the models' fundamental mechanisms for making decisions. This is when the useful Explainable Artificial Intelligence(XAI) frameworks like LIME (Local Interpretable Model-agnostic Explanations) come into play, revealing the precise visual details in the bean image structure that have a big impact on the projected outcomes of the model. Unlike the black-box model, LIME offers clear and contextualized justifications for every categorization. LIME displays the precise physical cues—like small color shifts in angular leaf spot or distinctive texturing features in bean rust—that inform the model's choices by emphasizing each of the pixels and zones that are significant within each bean leaf. This smooth understanding goes beyond specific image examinations, creating an auditory language of illness indicators throughout the collection.

Further than clarification, LIME's potency is demonstrated by its ability to promote real-world applications across a range of industries. Farmers can identify earlier illness indicators and take prompt action by using its easily understood graphic descriptions as an instructional tool. LIME's insights can drive model structures to concentrate on the essential causes of illnesses instead of external details, and they may additionally guide focused data gathering tactics for the creation of potential models. This will ensure that relevant and useful visual aspects are recorded. LIME essentially creates a link that connects the comprehensible realm of images and the transparent results of models. LIME changes bean disease identification from an accuracy-driven task to a based on knowledge route by simplifying the grid layout selection procedure. This reveals the complex visual representation of disease and opens the door for future developments in creating models and farming methods. Figure 5.7 shows the procedures involved in creating the explanation for the prediction of an input instance inform of a functional block diagram.

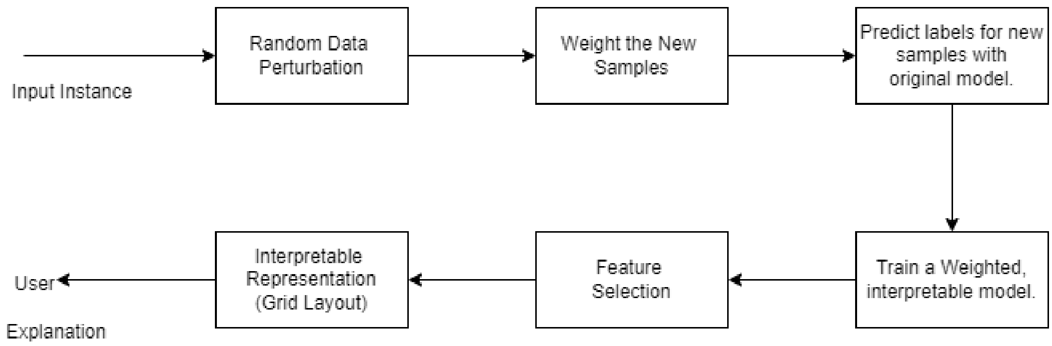

Figure 5.7: Functional Block Diagram Of LIME

The grid arrangement of this image dataset is shown in Figure 5.8.

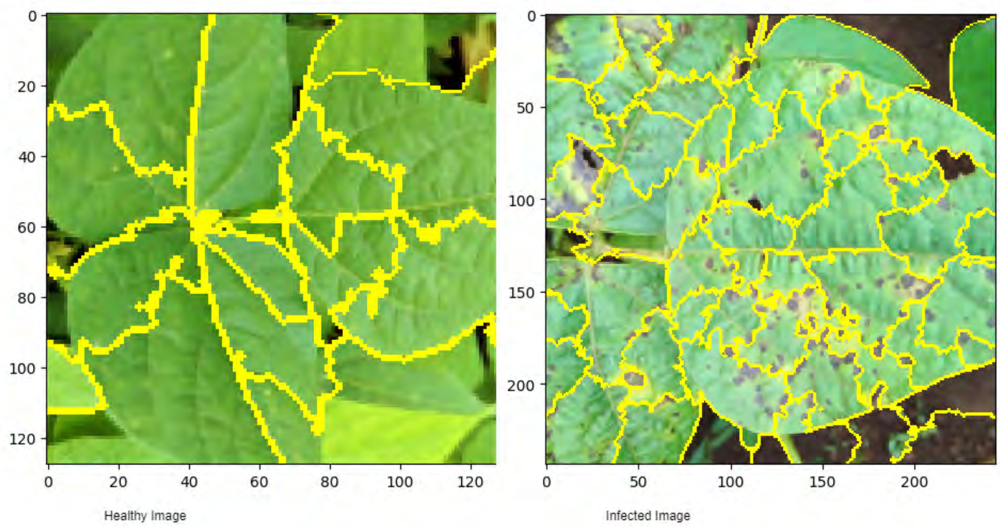

Figure 5.8: Grid Layout of the image samples

5.5 Chapter Summary

This chapter describes how we implemented deep learning models on our local system and showed their performance using both informative heatmaps and accuracy and loss graphs. For clarity, the evaluation method included a comprehensive analysis of each model utilizing many metrics, including F1-score, recall, and precision, which were combined into a classification table. A tabular comparative analysis provided a brief summary of the performance of each model. In addition, our study of Explainable Artificial Intelligence (XAI) methods—most notably LIME—gave us a better understanding of the models' interpretability, shedding light on how they make decisions and improving transparency in the complex field of deep learning.

Chapter 6

Website Deployment

This web application uses the Flask web framework, a Python web framework, in its architecture to provide a reliable web server. Flask makes it easier to manage HTTP requests and routes them to particular views and routes inside the application. Users can easily submit images because of the user-friendly interface that is offered to them. After being submitted, these images go through a processing stage before being shown on the online interface. The system stores the uploaded pictures in a special directory on the server to guarantee effective file management. The application's core engine uses transfer learning techniques to enable the model to perform complex picture analysis and inference tasks. This entails applying a pre-trained model that has been adjusted to meet the particular needs of the application. The results of this analysis procedure appear as evaluations that match the predictions of the model. After that, these rankings are displayed on the original images, providing viewers with insightful information about the examined pictures.Users can view the processed images, which are enhanced with details about the degree of infection on bean leaves and their anticipated class. In addition to improving user experience, this thorough integration of Flask, transfer learning, and result visualization also demonstrates the potential of fusing web frameworks with cutting-edge deep learning methods for significant image analysis as well as classification in the context of bean disease detection.
The process of this implementation is shown in Figure 6.1.

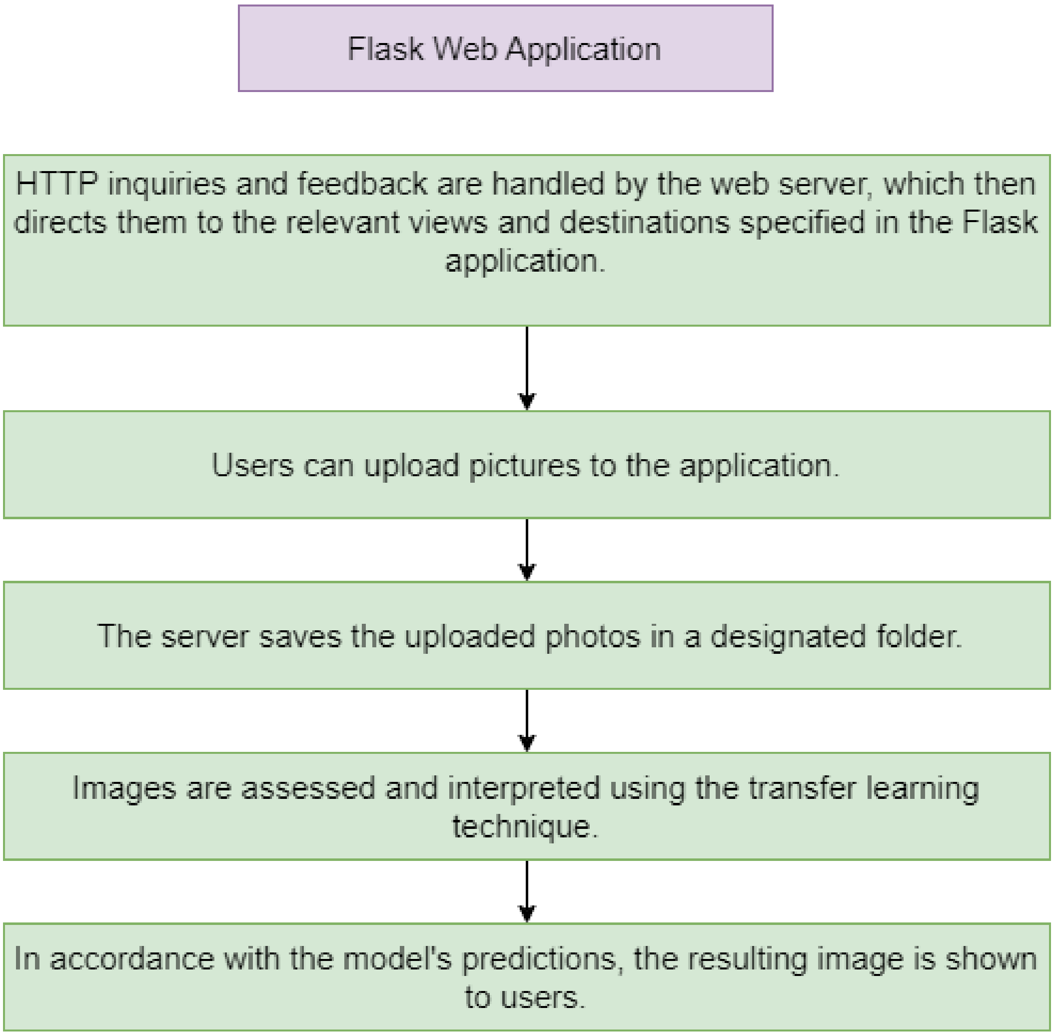

Figure 6.1: Workflow of the flask website

Figure 6.2 shows how the website uses the image categorization model in practice.

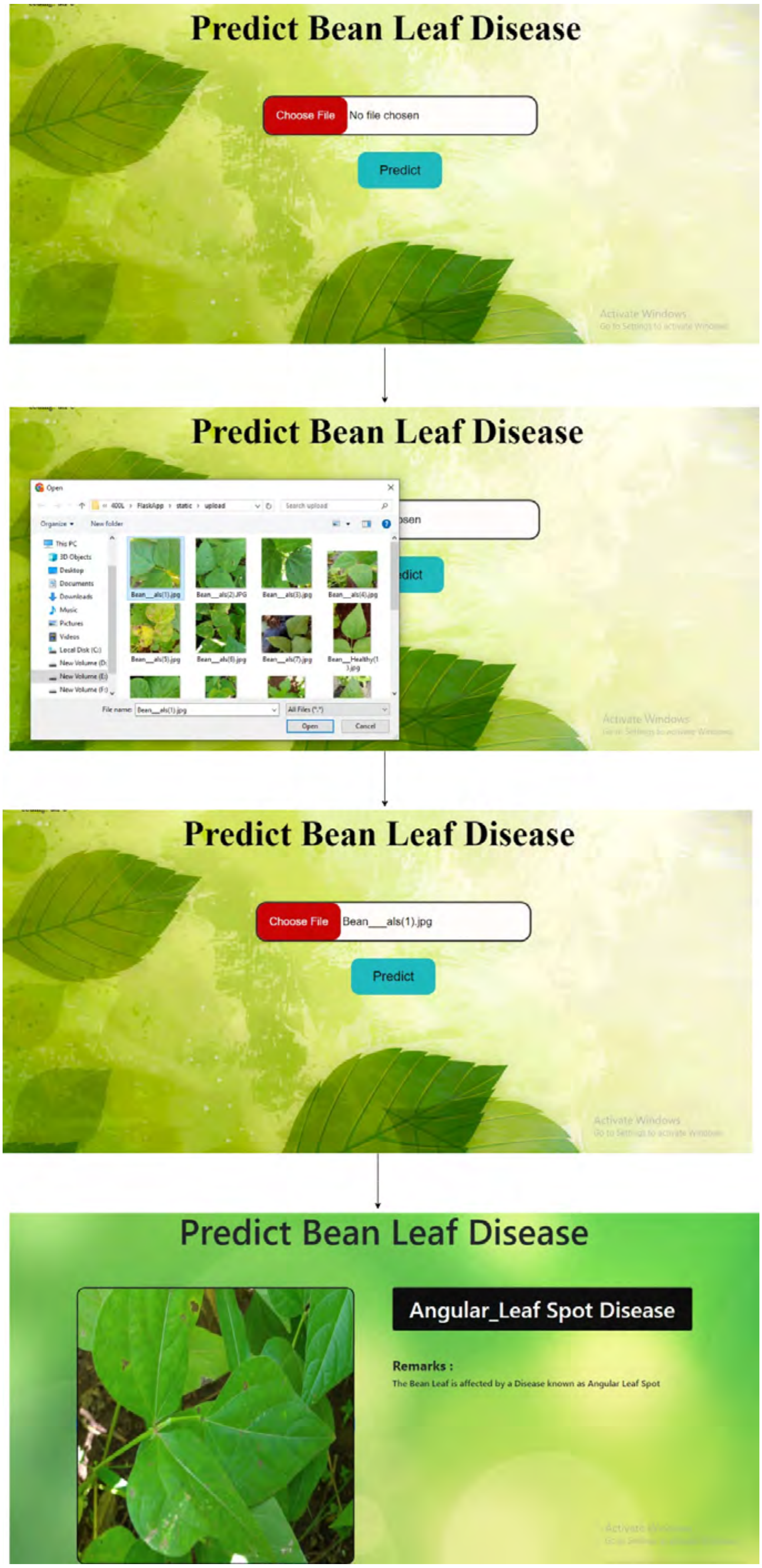

Figure 6.2: Flask Website Implementation

6.1 Limitations

Creating a web application to help farmers identify diseases in bean leaves is hampered by limited access to technology. Farmers in rural areas of Uganda might not have access to smartphones, or a dependable internet connection. This restriction reduces the potential impact and reach of the online application by preventing them from utilizing digital tools for disease detection. Creating a lightweight, offline-capable application, taking into account substitute devices like feature phones, and putting awareness campaigns into place to encourage farmers in these areas to adopt technology are some of the measures needed to overcome this obstacle.

6.2 Chapter summary

We centered on the FlaskWeb deployment for bean leaf image disease prediction in this chapter. It offered a thorough analysis of the deployment pipeline, describing the procedures needed to enable user access to the web-based application. We dug into the web application's practical implementation, showing how image processing helps the software accurately predict leaf diseases. However, we also tackled particular limitations, most notably the obstacles associated with GPU utilization in Uganda's remote regions. In spite of the deployment's effectiveness, the chapter emphasizes the necessity to take these geographic limitations into account and overcoming them in order to guarantee the predictive model's broad accessibility and influence in areas with insufficient GPU resources.

Chapter 7

Future Work and Conclusion

7.1 Conclusion

By examining the bean leaves dataset, our study on these deep learning models has demonstrated their ability to predict if a bean leaf is infected or not. Furthermore, the performance of Xception and ResNet50 is almost identical, with exceptional evaluation metrics across all classes and good accuracy overall although Xception achieves somewhat better evaluation metrics for every class. For all classes, VGG19's overall prediction accuracy is the least among the models due to its difficulty in distinguishing between the leaves infected with bean rust and angular leaf spot as both classes have almost similar physical characteristics. Based on these criteria, we are therefore convinced that our transfer learning models—which include our top five performing models—have a high predictive accuracy in detecting diseases that affect bean leaves. Finally, by creating a prediction accuracy score, we try to offer more thorough justifications for the predictions made by our transfer learning model.

To sum up, a viable way to mitigate the financial burden of these diseases on the agricultural sector in Uganda and other sub-Saharan African nations is the suggested system for detecting bean rust and angular leaf spot in beans using Deep Learning techniques. The system's goal is to identify such diseases quickly and accurately by offering a prompt and precise identification and diagnosis, which will ultimately save expenses and increase treatment effectiveness.

7.2 Future Work

In the future, we plan to implement a resource-effective deep learning system on resource-constrained IoT devices for real-time bean disease identification. The Xception approach, although incredibly effective, requires modification to integrate smoothly into these small systems. Employing real-time images acquired by the IoT device's camera, we hope to develop an economical Xception component that can quickly identify diseases through strategies like information distillation. Growers are empowered with practical information for timely action and enhanced yields thanks to this feedback in real time loop, which can be viewed via an extensive online application.

Bibliography

[1] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and electronics in agriculture*, vol. 72, no. 1, pp. 1–13, 2010.

[2] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 `[cs.CV]`.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, 2015. arXiv: 1512.00567 `[cs.CV]`.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[5] G. Huang, Z. Liu, and K. Weinberger, "Densely connected convolutional networks," p. 12, Aug. 2016.

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

[7] J. Minet, Y. Curnel, A. Gobin, *et al.*, "Crowdsourcing for agricultural applications: A review of uses and opportunities for a farmsourcing approach," *Computers and Electronics in Agriculture*, vol. 142, pp. 126–138, 2017.

[8] K. Kaizzi, A. Cyamweshi, C. Kibunja, *et al.*, "Bean yield and economic response to fertilizer in eastern and southern africa," *Nutrient Cycling in Agroecosystems*, vol. 111, pp. 47–60, 2018.

[9] E. Ronner, K. Descheemaeker, C. J. Almekinders, P. Ebanyat, and K. E. Giller, "Farmers' use and adaptation of improved climbing bean production practices in the highlands of uganda," *Agriculture, ecosystems & environment*, vol. 261, pp. 186–200, 2018.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," Jun. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[11] L. Arata, E. Fabrizi, and P. Sckokai, "A worldwide analysis of trend in crop yields and yield variability: Evidence from fao data," *Economic Modelling*, vol. 90, pp. 190–208, 2020.

[12] E. M. Katungi, E. Nduwarigira, N. Ntukamazina, *et al.*, "Food security and common bean productivity: Impacts of improved bean technology adoption among smallholder farmers in burundi," 2020.

[13] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," *arXiv preprint arXiv:2006.05278*, 2020.

[14] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, 2020. arXiv: 1905.11946 [cs.LG].

[15] S. H. Abed, A. S. Al-Waisy, H. J. Mohammed, and S. Al-Fahdawi, "A modern deep learning framework in robot vision for automated bean leaves diseases detection," *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 2, pp. 235–251, 2021.

[16] D. Bhatt, C. Patel, H. Talsania, *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[17] M. E. Chowdhury, T. Rahman, A. Khandakar, *et al.*, "Automatic and reliable leaf disease detection using deep learning techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, 2021.

[18] R. C. Joshi, M. Kaushik, M. K. Dutta, A. Srivastava, and N. Choudhary, "Virleafnet: Automatic analysis and viral disease diagnosis using deep-learning in vigna mungo plant," *Ecological Informatics*, vol. 61, p. 101 197, 2021.

[19] P. Sahu, A. Chug, A. P. Singh, D. Singh, and R. P. Singh, "Deep learning models for beans crop diseases: Classification and visualization techniques," *International Journal of Modern Agriculture*, vol. 10, no. 1, pp. 796–812, 2021.

[20] A. Aarizou and M. Merah, "Transfer learning for plant disease detection on complex images," in *2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)*, IEEE, 2022, pp. 1–6.

[21] R. Kumar, A. Chug, A. P. Singh, and D. Singh, "A systematic analysis of machine learning and deep learning based approaches for plant leaf disease classification: A review," *Journal of Sensors*, vol. 2022, 2022.

[22] D. Sutaji and O. Yıldız, "Lemoxinet: Lite ensemble mobilenetv2 and xception models to predict plant disease," *Ecological Informatics*, vol. 70, p. 101 698, 2022.

[23] R. Karthik, J. J. Alfred, and J. J. Kennedy, "Inception-based global context attention network for the classification of coffee leaf diseases," *Ecological Informatics*, vol. 77, p. 102 213, 2023.

[24] R. Kursun, K. K. Bastas, and M. Koklu, "Segmentation of dry bean (phaseolus vulgaris l.) leaf disease images with u-net and classification using deep learning algorithms," *European Food Research and Technology*, pp. 1–16, 2023.

[25] M. P. J. Mahenge, H. Mkwazu, R. R. Madege, B. Mwaipopo, and C. Maro, "Artificial intelligence and deep learning based technologies for emerging disease recognition and pest prediction in beans (phaseolus vulgaris l.): A systematic review," 2023.

[26] P. Midhunraj, K. Thivya, and M. Anand, "An analysis of plant diseases on detection and classification: From machine learning to deep learning techniques," *Multimedia Tools and Applications*, pp. 1–24, 2023.

[27] V. Singh, A. Chug, and A. P. Singh, "Classification of beans leaf diseases using fine tuned cnn model," *Procedia Computer Science*, vol. 218, pp. 348–356, 2023.

[28] A. B. Dash, S. Dash, and S. Padhy, "Identification of foxtail millet solution for malignant colorectal cell growth reduction using deep learning-based mobile net model," in *The Role of Women in Cultivating Sustainable Societies Through Millets*, IGI Global, 2024, pp. 42–59.

[29] J. Dhakshayani, B. Surendiran, and J. Jyothsna, "Artificial intelligence in precision agriculture: A systematic review on tools, techniques, and applications," *Predictive Analytics in Smart Agriculture*, pp. 37–57, 2024.

[30] I. Gupta and A. Singh, "Mango leaf disease detection using deep convolutional neural networks," in *Manufacturing Technologies and Production Systems*, CRC Press, 2024, pp. 389–396.

[31] V. C. Khade and S. B. Patil, "A hybrid deep learning approach for crop disease severity level prediction," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 9s, pp. 215–224, 2024.

[32] M. M. Khalid and O. Karan, "Deep learning for plant disease detection," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 75–84, 2024.

[33] R. K. Lomotey, S. Kumi, R. Orji, and R. Deters, "Automatic detection and diagnosis of cocoa diseases using mobile tech and deep learning," *International Journal of Sustainable Agricultural Management and Informatics*, vol. 10, no. 1, pp. 92–119, 2024.

[34] M. Nawaz, T. Nazir, A. Javed, S. T. Amin, F. Jeribi, and A. Tahir, "Coffeenet: A deep learning approach for coffee plant leaves diseases recognition," *Expert Systems with Applications*, vol. 237, p. 121 481, 2024.

[35] P. I. Ritharson, K. Raimond, X. A. Mary, J. E. Robert, and J. Andrew, "Deeprice: A deep learning and deep feature based classification of rice leaf disease subtypes," *Artificial Intelligence in Agriculture*, vol. 11, pp. 34–49, 2024.

[36] V. Sharma, A. K. Tripathi, P. Daga, M. Nidhi, and H. Mittal, "Clgannet: A novel method for maize leaf disease identification using clgan and deep cnn," *Signal Processing: Image Communication*, vol. 120, p. 117 074, 2024.

[37] R. A. A. H. M. APONG, "A systematic literature review on plant disease detection: Motivations, classification techniques, datasets, challenges, and future trends,"

[38] S. SERTTAŞ and E. DENİZ, "Disease detection in bean leaves using deep learning," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 65, no. 2, pp. 115–129,