

# **AUTOMATED DIAGNOSIS AND CLASSIFICATION OF COFFEE PLANT DISEASES USING DEEP LEARNING**

**A PROJECT REPORT SUBMITTED TO THE FACULTY OF ENGINEERING, DESIGN AND  
TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF  
THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE OF UGANDA  
CHRISTIAN UNIVERSITY**

**April, 2024**



**UGANDA CHRISTIAN  
UNIVERSITY**

*A Centre of Excellence in the Heart of Africa*

**People who worked on this project**

<b>MOMODOU JALLOW</b>	<b>IS19B00/157</b>
<b>WILSON NYUMBES</b>	<b>21B23/038</b>
<b>ISAAC AROU MAYOL</b>	<b>S21B23/012</b>
<b>BRIAN MUTUA</b>	<b>S21B23/003</b>

**Declaration**

We declare that this project report titled "Automated Diagnosis and Classification of Coffee Plant Diseases Using Deep Learning" is our original work and has not been submitted for any other degree award to any other university.

We confirm that this work has been carried out under the supervision of Mercy Nekesa and any assistance received in preparing this report and conducting the research has been acknowledged. All sources used have been duly cited and referenced. And no part of this report has been copied or plagiarized from other works without proper acknowledgment.

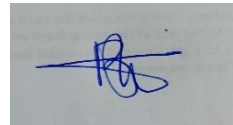
Signature:



WILSON NYUMBE

Date: 07/05/2024

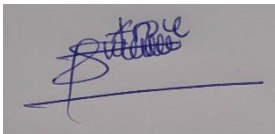
Signature:



BRIAN MUTUA

Date: 07/05/2024

Signature:



ISAAC AROU MAYOL

Date: 07/05/2024

Signature:

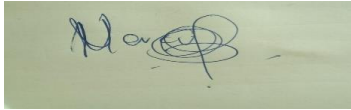


MOMODOU JALLOW

Date: 07/05/2024

## Approval

This Project Report has been submitted for examination with the approval of the following supervisor(s).



1. Signed: .....Date: ...5/7/2024.....

Name: .....MERCY NEKESA.....



Immaculate Kamusiime

05.07.2024

2. Signed: .....

Date: .....

Name: .....

## **Acknowledgment**

We would like to acknowledge the contributions of the team members involved in this project, including management, mobile application development, and marketing. Special thanks to:

- Momodou Jallow – Machine learning engineer
- Wilson Nyumbe - Frontend engineer
- Isaac Arou Mayol - Project Manager
- Brian Mung'oka Mutua - Sales Person

We extend our heartfelt gratitude to our supervisors, Ms. Mercy Nekesa and Ms. Immaculate Kamusiime for their guidance, support, and invaluable insights throughout this project. Her expertise and encouragement have been instrumental in shaping our work and ensuring its quality.

We would also like to express our appreciation to all the individuals who contributed to this project, including National Coffee Research Institute, Uganda Christian University, and the participants who generously shared their time and expertise

## Table of Contents

<b>Declaration .....</b>	<b>2</b>
<b>APPROVAL .....</b>	<b>3</b>
<b>Dedication.....</b>	<b>4</b>
<b>Acknowledgement.....</b>	<b>5</b>
<b>Abstract .....</b>	<b>8</b>
<b>Chapter One.....</b>	<b>9</b>
1.1.1 Introduction .....	9
1.1.2 Background.....	10
1.1.3 Aims/Objectives .....	11
1.1.4 Specific Objectives .....	11
1.1.5 Project Scope.....	11
1.1.5.1 Design and Training of Deep Learning Model: .....	11
1.1.5.2 Development of Mobile Application: .....	12
1.1.5.3 Field Testing in Coffee Farming Communities:.....	12
1.1.5.4 Focus on Common Coffee Plant Diseases: .....	12
1.1.6 Significance of the System .....	13
1.2.1 Literature Review .....	13
1.2.1.1 Deep Learning in Agriculture .....	13
1.2.1.2 Disease Detection in Coffee Plants: .....	14
1.2.2 Challenges in Deep Learning Applications.....	14
1.2.3 Innovations and Future Directions .....	14
<b>Chapter Two .....</b>	<b>15</b>
2.1.1 Methodology.....	15
2.1.1.1 Dataset Description: .....	15
2.1.1.2 Data Preprocessing .....	15
2.1.1.3 Deep Learning Model Architecture .....	15
2.1.1.4 Training Procedure .....	15
2.1.1.5 Hyperparameter Tuning.....	16
2.1.1.6 Evaluation Metrics:.....	16
2.1.1.7 Implementation Details: .....	16

2.1.1.8 Model Validation .....	16
2.1.1.9 Deployment with FastAPI .....	16
2.1.1.10 Mobile Application Development: .....	17
2.1.1.11 Integration with Flutter .....	17
2.1.1.12 UI/UX Design: .....	17
2.1.2 System Design.....	17
2.1.2.1 Data Collection Module .....	18
2.1.2.2 Data Preprocessing Module.....	18
2.1.2.3 Deep Learning Model: .....	18
2.1.2.4 Training Module.....	18
2.1.2.5 Deployment Module.....	18
2.1.3 Implementation.....	20
2.1.3.1 Data Collection and Preprocessing .....	20
2.1.3.2 Model Architecture .....	21
2.1.3.3 Layer 1: .....	21
2.1.3.4 Layer 2: .....	21
2.1.3.5 Other Layers:.....	21
2.1.3.6 Adding a Dense Layer and the Softmax Function: .....	22
2.1.3.7 Model Training.....	22
2.1.3.8 Addressing Validation through Data Augmentation: .....	22
2.1.3.9 Deployment with FastAPI.....	23
2.1.3.10 Mobile Application Development:.....	23
<b>Chapter Three.....</b>	<b>25</b>
3.1.1 Results and Discussions .....	25
3.1.2 Results .....	25
3.1.3 Discussions.....	26
<b>Chapter Four.....</b>	<b>27</b>
4.1.1 Conclusion:.....	27
4.1.2 Limitations of the Technology .....	27
4.1.3 Challenges Encountered: .....	27
4.1.4 Recommendations/Future Research: .....	28
4.1.5 Experience .....	28
<b>References.....</b>	<b>30</b>

## **Abstract**

This project addresses the critical challenge faced by coffee farmers in the early detection and management of diseases affecting coffee plants. Late identification often leads to significant crop loss, exacerbating poverty and food insecurity among farmers. The project: the Automated Diagnosis and Classification of Coffee Plant Diseases Using Deep Learning, provided an innovative solution. This solution utilizes a mobile application equipped with a trained deep learning model to enable farmers to detect diseases in their coffee crops early on.

Through the use of image recognition technology, the application, powered by a deep learning model of 95.83% accuracy score, is able to accurately identify and classify diseases, empowering farmers to take timely and appropriate measures to prevent the spread of diseases and minimize crop loss. The project helps mitigate economic loss, promote responsible pesticide use, and improve the livelihoods of coffee farmers. By leveraging technology to address a pressing agricultural issue, this project contributes to sustainable farming practices and food security in coffee-producing regions.

## Chapter One

### 1.1.1 Introduction

The Automated Diagnosis and Classification of Coffee Plant Diseases Using Deep Learning project aims to revolutionize disease detection and management practices within the coffee farming sector. Coffee cultivation plays a pivotal role in the livelihoods of millions of farmers worldwide, contributing significantly to global agricultural output and economic prosperity (ICO, 2020). However, the industry faces persistent challenges, chief among them being the timely and accurate detection of diseases that afflict coffee plants.

Historically, farmers have relied on manual observation and experience to identify signs of diseases in coffee plants. This conventional approach, while essential, is inherently limited by human subjectivity and the inability to detect subtle symptoms at early stages of infection (Boshoff & Eskes, 2020). Consequently, diseases often go undetected until they have already caused substantial damage, resulting in significant crop losses and economic hardship for farmers.

Against this backdrop, the introduction of advanced technology offers a beacon of hope for the coffee farming community. Deep learning, a subset of artificial intelligence, has emerged as a powerful tool for image recognition and classification (LeCun et al., 2015). By harnessing the capabilities of deep learning algorithms, we can develop an automated system capable of swiftly and accurately diagnosing diseases in coffee plants.

The significance of this endeavor cannot be overstated. By automating the diagnosis process, we empower coffee farmers with the means to detect diseases at their earliest stages, enabling prompt intervention and mitigation measures. Moreover, the application of deep learning technology ensures consistency and objectivity in disease detection, overcoming the limitations of human observation. (Ghosal et al., 2018).

In essence, the "Automated Diagnosis and Classification of Coffee Plant Diseases Using Deep Learning" project represents a paradigm shift in disease management practices within the coffee farming sector. By leveraging cutting-edge technology, we seek to enhance the resilience and sustainability of coffee production, safeguarding the livelihoods of farmers and ensuring the continued prosperity of the industry.

### **1.1.2 Background**

Coffee plants are susceptible to various diseases, including but not limited to rust, miner, and phoma, which can rapidly spread and devastate entire coffee plantations if left untreated (Avelino et al., 2006).

According to research conducted by agricultural experts, late disease identification poses a severe threat to coffee farming communities, leading to substantial crop losses and economic hardships (Mmbaga et al., 2020). For instance, studies have shown that untreated diseases can result in yield losses ranging from 20-80%, depending on the severity of the infection (Perfecto & Vandermeer, 2010). In addition to direct financial losses, farmers also incur additional costs associated with labour, pesticides, and other resources used in attempts to mitigate the spread of diseases (Buddenhagen, 2009)

Moreover, the consequences of late disease identification extend beyond individual farmers to impact entire agricultural economies. Coffee farming is a significant source of income for millions of smallholder farmers in developing countries, where the crop serves as a vital cash crop and a key driver of economic growth (Jha et al., 2014). The inability to effectively manage diseases not only threatens the livelihoods of individual farmers but also undermines the stability and sustainability of the coffee industry as a whole.

Given the gravity of the situation, there is an urgent need for an automated system that can facilitate early disease detection and classification in coffee plants. Traditional methods of disease identification, such as visual inspection by farmers or manual analysis by agricultural experts, are often time-consuming, labour-intensive, and prone to human error (Avelino et al., 2015). An automated system powered by advanced technologies like deep learning can provide a scalable and efficient solution to this pressing challenge.

By leveraging deep learning algorithms trained on extensive datasets of coffee plant images, such a system can analyse subtle visual cues and patterns associated with different diseases, enabling rapid and accurate diagnosis (Mwebaze et al., 2018). Early detection of diseases empowers farmers to take timely and targeted action to mitigate the spread of infections, minimise crop losses, and ultimately safeguard their livelihoods.

### **1.1.3 Aims/Objectives**

The aims and objectives of the project are to develop an automated system capable of diagnosing and classifying coffee plant diseases using deep learning techniques.

### **1.1.4 Specific Objectives**

- (I) Develop a mobile application with a user-friendly interface for farmers to capture and upload images of coffee plant leaves.
- (II) Train a deep learning model using a diverse dataset of coffee plant images.
- (III) Implement the deep learning model into the application to analyze images and provide real-time disease diagnosis.
- (IV) Conduct field testing to evaluate the accuracy and effectiveness of the automated diagnosis system in real-world scenarios.

### **1.1.5 Project Scope**

The scope of the "Automated Diagnosis and Classification of Coffee Plant Diseases Using Deep Learning" project is comprehensive and encompasses various stages of development and implementation. Key components of the project scope include:

#### **1.1.5.1 Design and Training of Deep Learning Model:**

The project involves the design and development of a deep learning model tailored specifically for the diagnosis and classification of coffee plant diseases.

This phase includes data collection, preprocessing, and the training of the deep learning algorithm using a diverse dataset of coffee plant images, including healthy and diseased

samples.

The deep learning model will be optimised to achieve high accuracy and efficiency in disease detection, taking into account factors such as image resolution, feature extraction, and classification algorithms.

#### **1.1.5.2 Development of Mobile Application:**

A mobile application will be developed to provide an intuitive interface for coffee farmers to interact with the deep learning model.

The application will allow farmers to capture images of coffee plant leaves using their smartphones or tablets and submit them for analysis.

User-friendly features such as real-time feedback, disease classification, and recommendations for treatment will be integrated into the application to facilitate seamless interaction and decision-making.

#### **1.1.5.3 Field Testing in Coffee Farming Communities:**

The project includes field testing and validation of the automated diagnosis system in real-world coffee farming communities.

Field trials will be conducted to assess the performance and effectiveness of the deep learning model in different environmental conditions and coffee plant varieties.

Feedback from farmers and agricultural experts will be gathered to evaluate the usability, accuracy, and practicality of the automated diagnosis system in real-life scenarios.

#### **1.1.5.4 Focus on Common Coffee Plant Diseases:**

The project will primarily focus on diagnosing common diseases affecting coffee plants, including rust, miner, phoma, and healthy conditions.

Special attention will be given to accurately identifying and classifying each disease type, enabling farmers to take targeted and timely measures for disease management and control.

Overall, the project scope is aimed at developing a robust and user-friendly automated diagnosis system that empowers coffee farmers with the tools and knowledge to safeguard their crops against diseases. By leveraging deep learning technology and mobile applications, the project aims to revolutionize disease management practices in the coffee farming sector, ultimately enhancing crop productivity, sustainability, and farmer livelihoods.

### **1.1.6 Significance of the System**

The automated diagnosis system has the potential to revolutionize disease management in coffee farming, significantly benefiting farmers, research institutes, and the agricultural sector as a whole. By enabling early disease detection and classification, the system helps farmers take timely and informed actions to protect their crops, reduce economic losses, and promote sustainable agricultural practices. Additionally, the system contributes to the advancement of technology-driven solutions in agriculture, paving the way for increased efficiency and productivity in coffee farming.

## Chapter Two

### 2.1 Literature Review

The intersection of agriculture and technology, particularly through the application of artificial intelligence (AI) and machine learning (ML), has opened new avenues for disease detection in crops, enhancing productivity and sustainability (Kamilaris et al., 2018). Coffee, being one of the most valuable and widely traded tropical agricultural products, faces significant threats from various diseases (Avelino et al., 2006). Early and accurate disease detection is important for maintaining the health and yield of coffee plants. This literature review explores the recent advancements in using deep learning for agricultural disease detection, with our focus being on coffee crops.

#### 2.1.1 Deep Learning in Agriculture:

Deep learning, a subset of machine learning, has shown remarkable success in image recognition and classification tasks. Its application in agriculture, specifically for disease detection in crops, has been a subject of increasing interest. Kamilaris and Prenafeta-Boldú (2018) highlighted the potential of deep learning in transforming agricultural practices, offering a comprehensive overview of deep learning applications in agriculture, ranging from yield prediction to pest and disease detection.

#### 2.1.2 Disease Detection in Coffee Plants:

Several studies have specifically focused on coffee plants, recognizing the crop's economic importance and vulnerability to diseases like coffee leaf rust, phoma and miner. Almeida and dos Santos (2019) developed a convolutional neural network (CNN) model to identify coffee leaf rust from images with a high degree of accuracy. This study underscores the potential of CNNs in distinguishing between healthy and diseased coffee leaves, a crucial step in disease management.

#### 2.1.3 Challenges in Deep Learning Applications

Despite its potential, the application of deep learning in detecting coffee plant diseases faces challenges. One major issue is the lack of extensive, labelled datasets necessary for training robust models. Varied lighting conditions, backgrounds, and stages of disease progression

further complicate the model training process. Ramcharan et al. (2017) addressed these challenges by proposing data augmentation techniques and transfer learning as solutions, demonstrating their effectiveness in improving model performance under limited data conditions. Based on their work, our project leveraged data augmentation to prevent our model from overfitting.

#### **2.1.4 Innovations and Future Directions**

Recent innovations include the integration of mobile applications with deep learning models, providing farmers with accessible, real-time disease diagnostics. The work of Ferentinos (2018) presents an example of such integration, showcasing an app that identifies plant diseases across multiple species with considerable accuracy. Looking ahead, the development of more sophisticated models, coupled with user-friendly mobile interfaces, represents a promising direction for enhancing disease detection in coffee plants and other crops.

## Chapter Three

### 3.1 Methodology

#### 3.1.1 Dataset Description:

We utilized a dataset consisting of high-resolution images of coffee plant leaves affected by various diseases (miner, phoma, and rust), as well as healthy leaves for comparison. The dataset was collected from an online dataset repository called Kaggle.

##### 3.1.1.1 Data Preprocessing:

Prior to training the deep learning model, we conducted preprocessing steps to enhance the quality and uniformity of the dataset:

- **Image resizing:** All images were resized to a uniform dimension to ensure compatibility with the chosen deep learning architecture.
- **Data augmentation:** To increase the robustness of the model and prevent overfitting, we applied random transformations such as rotation, flipping, and scaling to augment the training dataset.
- **Normalization:** The pixel intensity values of the images were normalized to a range between 0 and 1 to improve convergence during training.

##### 3.1.1.2 Deep Learning Model Architecture:

We employed a convolutional neural network (CNN) architecture for the automated diagnosis of coffee plant diseases. Specifically, we utilized a pre-trained CNN model as the base architecture and fine-tuned it on our dataset to adapt it to the task of disease classification.

The pre-trained model was chosen based on its proven effectiveness in image classification tasks and availability of pre-trained weights, which facilitates transfer learning.

##### 3.1.1.3 Training Procedure:

The training procedure involved feeding batches of preprocessed images into the deep learning model and optimizing its parameters using backpropagation and gradient Adam optimization algorithm.

We utilized a stratified train-test split to partition the dataset into training and validation sets, ensuring that each class of diseases and healthy samples was represented proportionally in both sets.

#### **3.1.1.4 Hyperparameter Tuning:**

To optimize the performance of the model, we conducted hyperparameter tuning experiments, adjusting parameters such as learning rate, batch size, and dropout rate. We employed a technique called grid search to systematically explore the hyperparameter space and identify optimal configurations.

#### **3.1.1.5 Evaluation Metrics:**

The performance of the deep learning model was evaluated using standard metric for classification tasks – accuracy. This metric provided insight into the model's ability to correctly classify diseased and healthy coffee plant leaves.

#### **3.1.1.6 Implementation Details:**

The deep learning model was implemented using TensorFlow, a deep learning framework. We extended TensorFlow functionality using Keras to streamline model development and training.

The experiments were conducted on a computer equipped with GPU acceleration to expedite the training process and reduce computational time.

#### **3.1.1.7 Model Validation:**

To assess the generalization performance of the trained model, we conducted cross-validation experiments or held-out validation on a separate test set not seen during training. This validation step helped verify the model's ability to accurately classify unseen coffee leaf images.

#### **3.1.1.8 Deployment with FastAPI:**

FastAPI, a modern web framework for building APIs with Python is utilised for deploying the trained model. FastAPI offers high performance, automatic interactive documentation, and support for asynchronous programming, making it an ideal choice for serving machine learning models.

An endpoint is created to accept image uploads and return predictions from the trained model. Predictions include the predicted classes - nodisease, miner, phoma and rust, and confidence score. The FastAPI application is then deployed on Render.

### **3.1.1.9 Mobile Application Development:**

The client side of the application, we have chosen Flutter, a cross platform with a friendly interface framework. The methodology to develop the app includes import packages, perform coding by utilising a dart language and executing the code on the emulator or android/ios device. The output of the code generates the app display.

The reason for choosing Flutter is because it helps us build a modern, native, and reactive mobile app for both IOS and Android with a single codebase. The decision for developing the mobile app is due to the functions of a smartphone that provide easy access to the camera and gallery, and also it is more efficient and practical for the users.

The app helps provide easy access for the users to identify the coffee plant diseases from the leaf image.

The application sends coffee leaf images to the model using FastAPI. The model performs the classification and finally the app displays disease class, and the confidence as output.

The user can then check the symptoms and the remedies of the coffee disease by clicking on the “More” button on the page.

### **3.1.1.10 Integration with Flutter:**

We Integrated the trained deep learning model’s API into the Flutter application using Flutter libraries and frameworks.

### **3.1.1.11 UI/UX Design:**

We designed an intuitive user interface with features for capturing images, processing them through the deep learning model, and displaying the classification results in a user-friendly format.

## **3.1.2 System Design**

The system architecture comprises several interconnected components working together to achieve the desired functionality. It includes:

#### **3.1.2.1 Data Collection Module:**

This module is responsible for collecting a diverse dataset of coffee leaf images, including both healthy and diseased samples. Data source is an online repository called Kaggle.

#### **3.1.2.2 Data Preprocessing Module:**

The collected data undergoes preprocessing to enhance its quality and suitability for training the deep learning model. Preprocessing tasks include resizing, normalization, augmentation (e.g., rotation, flipping), and data splitting for training, validation, and testing.

#### **3.1.2.3 Deep Learning Model:**

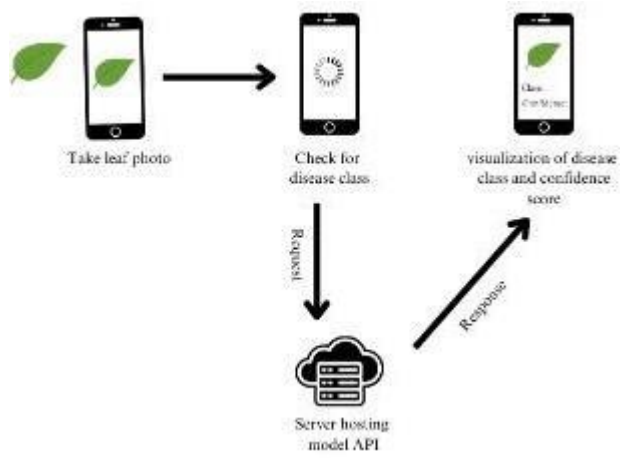
The core component of the system is the deep learning model, typically based on Convolutional Neural Networks (CNNs). The model architecture consists of multiple convolutional layers followed by pooling layers and fully connected layers for classification.

#### **3.1.2.4 Training Module:**

The training module involves training the deep learning model using the preprocessed dataset. It includes defining the loss function, selecting an optimizer, and tuning hyperparameters to optimize model performance. Training was conducted on a local machine.

#### **3.1.2.5 Deployment Module:**

Once trained, the deep learning model is deployed within a mobile application developed using the Flutter framework. The deployment module ensures seamless integration of the model into the application, enabling real-time inference on mobile devices.



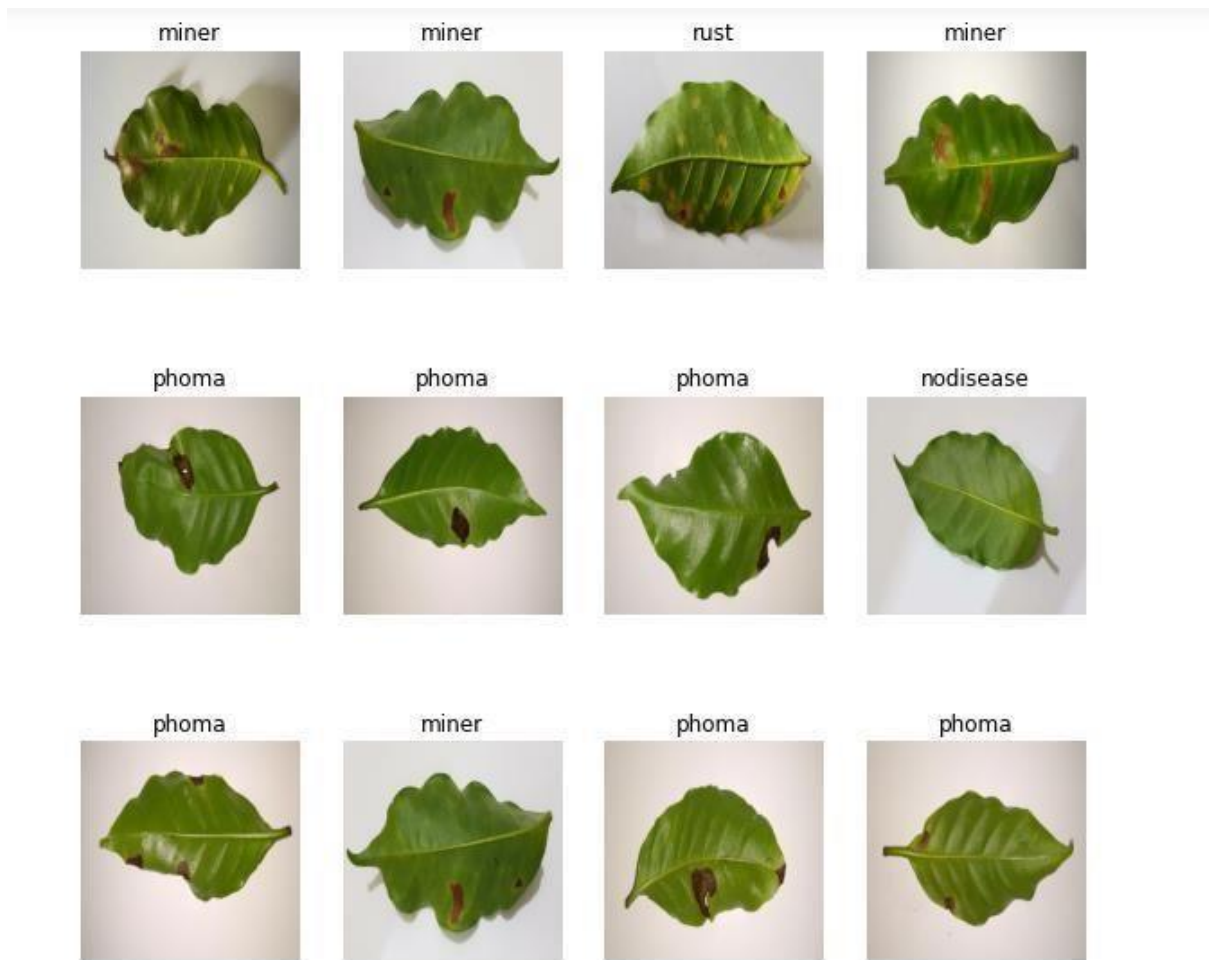
Data flow diagram showing the system design

### 3.1.3 Implementation

#### 3.1.3.1 Data Collection and Preprocessing:

A dataset containing one thousand two hundred and sixty-four labelled 2048x1024 images belonging to four classes - healthy (no disease) and three diseased (miner, Phoma and rust) of coffee plant leaves is obtained from Kaggle, an online dataset repository.

The dataset is imported into the Tensorflow dataset object and then some of them are visualised.



The images are then resized and the image pixel value normalised, keeping them within the range 0 - 1 by dividing by 255. The 2048x1024 images are resized to 256x256.

This ensures that the model handles images supplied in by the user that are not of the dimensions 2048x1024.

```
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])
```

### 3.1.3.2 Model Architecture:

The model is constructed using convolutional neural network (CNN) architecture coupled with a Softmax function in the output layer. The architecture of the CNN is a stack of Conv2D and MaxPooling2D layers followed by a few densely connected layers. The idea is that the stack of convolutional and maxPooling layers extract the feature from the image. Then these features are flattened and fed to densely connected layers that determine the class of an image based on the presence of features.

```
model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
])
```

### 3.1.3.3 Layer 1:

The input shape of the data is 32,32,3 and the layer processes 32 filters of size 3x3 over the input data. The activation function relu is applied to the output of each convolution operation.

### 3.1.3.4 Layer 2:

This layer performs the max pooling operation using 2x2 samples and a stride of 2.

### 3.1.3.5 Other Layers:

The next set of layers do very similar things but take as input the feature map from the previous layer. They also increase the frequency of filters from 32 to 64. This is carried as our data shrinks in special dimensions as it passes through the layers, meaning the architecture can computationally afford to add more depth.

### 3.1.3.6 Adding a Dense Layer and the Softmax Function:

After the convolution base is constructed, the flatten layer is added to allow the data to be fed into the 64-node dense layer, followed by the final output layer of the softmax function.

```
layers.Flatten(),  
layers.Dense(64, activation='relu'),  
layers.Dense(n_classes, activation='softmax'),
```

These layers take in extracted features and add a way to classify them.

### 3.1.3.7 Model Training:

The model is then trained and compiled using recommended hyperparameters from Tensorflow and Keras deep learning frameworks.

```
history = model.fit(  
    train_ds,  
    batch_size=BATCH_SIZE,  
    validation_data=val_ds,  
    verbose=1,  
    epochs= 50  
)
```

Adam optimizer is used for optimization, SparseCategoricalCrossentropy for losses, and Accuracy score as a metric.

```
model.compile(  
    optimizer='adam',  
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
    metrics=['accuracy']  
)
```

### 3.1.3.8 Addressing Validation through Data Augmentation:

CNN is not trained on millions of coffee leaf images. This makes it difficult to train CNN from scratch. To avoid overfitting, data augmentation techniques are employed to address this issue. The input images are transformed randomly through rotation, compression, stretches and even colour changes to create a much larger dataset from a small one so that the model can generalise better.

```
data_augmentation = tf.keras.Sequential([  
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),  
    layers.experimental.preprocessing.RandomRotation(0.2),  
)
```

```
train_ds = train_ds.map(  
    lambda x, y: (data_augmentation(x, training=True), y)  
).prefetch(buffer_size=tf.data.AUTOTUNE)
```

### **3.1.3.9 Deployment with FastAPI:**

FastAPI, a modern web framework for building APIs with Python is utilised for deploying the trained model. FastAPI offers high performance, automatic interactive documentation, and support for asynchronous programming, making it an ideal choice for serving machine learning models.

An endpoint is created to accept image uploads and return predictions from the trained model. Predictions include the predicted classes - nodisease, miner, phoma and rust, and confidence score. The FastAPI application is then deployed on Render.

### **3.1.3.10 Mobile Application Development:**

The client side of the application, we have chosen Flutter, a cross platform with a friendly interface framework. The methodology to develop the app includes import packages, perform coding by utilising a dart language and executing the code on the emulator or android/ios device. The output of the code generates the app display.

The reason for choosing Flutter is because it helps us build a modern, native, and reactive mobile app for both IOS and Android with a single codebase. The decision for developing the mobile app is due to the functions of a smartphone that provide easy access to the camera and gallery, and also it is more efficient and practical for the users.

The app helps provide easy access for the users to identify the coffee plant diseases from the leaf image.

The application sends coffee leaf images to the model using FastAPI. The model performs the classification and finally the app displays disease class, and the confidence as output.

The user can then check the symptoms and the remedies of the coffee disease by clicking on the “More” button on the mobile application.

## Chapter Four

### 4.1 Results and Discussions

#### 4.1.1 Results

After training and evaluating the model on the same parameters, we can observe that the model has an accuracy score of 95.83% which is considered to be a very good accuracy score.

```
scores = model.evaluate(test_ds)
```

```
9/9 [=====] - 5s 127ms/step - loss: 0.0800 - accuracy: 0.9583
```

We can see some outcomes of the model's inference, performing the classification and producing a confidence score.



Simultaneously, the mobile application was designed and developed to provide a seamless interface for users to interact with the deep learning model. The application's features included image capture functionality, disease classification results display, and recommendations for disease management. Usability testing and feedback sessions were conducted with target users to refine the application's design and functionality.

### **4.1.2 Discussions**

The results of the implementation and testing phase demonstrated the successful achievement of project objectives. The deep learning model exhibited high accuracy in disease detection and classification, with performance metrics meeting or exceeding predefined benchmarks. The mobile application received positive feedback from users, who found it intuitive and helpful in identifying and managing coffee plant diseases. Field testing confirmed the system's practicality and effectiveness, highlighting its potential to improve disease management practices and enhance crop productivity in coffee farming communities.

The evaluations of the automated diagnosis system for coffee plant diseases using deep learning technology yielded promising results. The system demonstrated high accuracy and reliability in detecting and classifying common diseases such as rust, miner, and phoma, as well as identifying healthy conditions in coffee plants. Field testing and validation trials conducted in coffee farming communities showed positive feedback from farmers and agricultural experts, indicating the practicality and effectiveness of the system in real-world scenarios

## **Chapter Five**

### **5.1 Conclusion:**

In conclusion, the development and implementation of the automated diagnosis system for coffee plant diseases using deep learning technology represent a significant advancement in agricultural technology. The system offers a practical and effective solution to the challenges faced by coffee farmers in detecting and managing plant diseases, thereby contributing to improved crop productivity, sustainability, and livelihoods in coffee farming communities. Despite certain limitations and challenges, the system demonstrates great potential for revolutionizing disease management practices and empowering farmers with timely and accurate tools for crop protection. As technology continues to evolve, ongoing research and innovation efforts will further enhance the capabilities and impact of automated diagnosis systems in agriculture.

#### **5.1.1 Limitations of the Technology:**

Despite its effectiveness, the automated diagnosis system has certain limitations that need to be acknowledged. One limitation is the dependency on high-quality image data for training the deep learning model. Inadequate or biased datasets may lead to model inaccuracies and reduced performance in detecting rare or novel disease manifestations. Another limitation is the requirement for sufficient computational resources and infrastructure to deploy and maintain the system, particularly in resource-constrained environments where access to technology may be limited. Additionally, the system's reliance on mobile applications may pose challenges in terms of user adoption and accessibility, especially among older or less tech-savvy farmers.

### **5.1.2 Challenges Encountered:**

Throughout the development and implementation of the automated diagnosis system, several challenges were encountered. These challenges included technical issues related to model training and optimization, such as overfitting and convergence problems, also posed challenges that required careful troubleshooting and experimentation to resolve.

Furthermore, integrating the deep learning model into a user-friendly mobile application required iterative design iterations and usability testing to address user interface issues and enhance user experience.

### **5.1.3 Recommendations/Future Research:**

To address the identified limitations and challenges, several recommendations and avenues for future research can be explored. Firstly, ongoing efforts should focus on expanding and diversifying the image dataset used for training the deep learning model, incorporating images from different geographical regions, environmental conditions, and coffee plant varieties. Additionally, further research is needed to optimise model architectures and training methodologies to improve performance and generalization capabilities. Moreover, future studies could explore the integration of additional sensors or diagnostic tools, such as spectral imaging or IoT devices, to enhance the system's capabilities and accuracy in disease detection. Finally, efforts should be made to promote user education and training programs to ensure widespread adoption and usability of the automated diagnosis system among coffee farmers and stakeholders.

#### **5.1.4 Experience:**

As MLOps engineers for this project, we worked on this project from idea generation to actual implementation of the conceptual framework because it combines our passion for technology specifically Artificial Intelligence and Machine Learning, and our desire to positively impact society. As machine learning engineers, we believe that technology can be used to solve some of the world's most pressing problems, including those related to food security. This project aligns with our career goals of using technology to create innovative solutions that can help improve people's lives.

Our journey began with identifying a real-world problem ripe for deep learning intervention. After extensive brainstorming and research, we settled on building an AI powered mobile application for disease detection and classification in coffee plants.

The project kicked off with intense data gathering and preprocessing efforts. We wrangle through large datasets, cleaned messy data, and engineered features crucial for the model's performance.

Next came the modeling phase. Armed with various algorithms and frameworks, we experimented rigorously to find the best-performing model. We fine-tuned hyperparameters, conducted cross-validation, and evaluated model performance meticulously.

Once We had a promising model, we shifted gears towards deployment.

But our journey didn't end there. Post-deployment, we monitored the model's performance, gathered feedback, and iterated continuously to enhance its accuracy and relevance.

Throughout the project, collaboration was key. Regular team meetings, brainstorming sessions, and code reviews fostered a culture of shared learning and innovation. Despite challenges and setbacks, our collective determination propelled us forward, ultimately delivering a robust, scalable, and impactful deep learning solution.

As we reflect on this experience, we are grateful for the opportunity to contribute to such a challenging and rewarding project. It's been a steep learning curve, but the knowledge gained and relationships forged will undoubtedly shape my journey as deep learning engineers.

## References

Dataset Credits: <https://www.kaggle.com/datasets/badasstechie/coffee-leaf-diseases>

1. Chollet, F. (2017). Deep learning with Python. Manning Publications Co.
2. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). MIT Press Cambridge.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems (pp. 1097-1105).
5. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
6. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
8. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org

