UNIVERSITE JOSEPH KI-ZERBO

ECOLE DOCTORALE INFORMATIQUE ET CHANGEMENT CLIMATIQUE (EDICC)

BURKINA FASO

La Patrie ou la Mort, nous Vaincrons

Order N° : .....

MASTER RESEARCH PROGRAM

SPECIALITY: INFORMATICS FOR CLIMATE CHANGE (ICC)

MASTER THESIS

Subject:

Developing an application to detect groundnut diseases to enhance decision-making

Presented (14 July) by:

AISSATA DOUSSOU DEMBELE

Examination jury

Pr Amadé OUEDRAOGO Full Professor / Plant Biology and Ecology, University Joseph KI-ZERBO

Dr Abel NANA, Associate Professor/ Plant Pathology, University Joseph KI-ZERBO

Dr Zakaria KIEBRE, Associate Professor/Genetics and plant breeding, University Joseph KI-ZERBO

Dr Bouba TRAORE, Agriculture and Climate Change, International Institute for Crops of Semi-Arid Tropical

Academic year 2023-2025

WASCAL West African Science Service Centre on Climate Change and Adapted Land Use

Federal Ministry of Research, Technology and Space

SUMMARY

DEDICATION v
ACKNOWLEDGEMENTS vi
DECLARATION viii
ABSTRAST xi
RESUME xii
Introduction 1
Chapter 1: Literature review 6
1.1. Overview of groundnut 6
1.2. Description of groundnut leaf diseases and symptoms 10
1.3. Impact of climate change on groundnut diseases spread and severity 16
1.4. Machine learning (ML) techniques 18
1.5. Deep learning technique 20
1.6. Plant disease detection software 25
1.7. Software requirements 27
Chapter 2 Material and methods 32
2.1. Material 32
2.1.1. Study area 32
2.1.2. Python programming language 32
2.2. Methods 33
2.2.1. Data collection 33
2.2.2. Image processing and analysis 35
2.2.3. Different models used for training 36
2.2.4. Software design architecture 39
2.2.5. Application development 40
Chapter 3 Results and discussion 45
3.1. Results 45
3.1.1. CNN model training 45
3.1.2. Confusion matrix of CNN 46
3.1.3. VGG16 model training 47
3.1.4. Confusion matrix of VGG16 48
3.1.5. VGG19 model training 50
3.1.6. Confusion matrix of VGG19 50
3.2. Discussion 52
Conclusion and perspectives 56

Bibliography references.....58
Appendix A: Code VGG19.....I
Appendix B: Code source for the application V

LIST OF FIGURES

Figure 1: An Arachis hypogaea plant 7
Figure 2: Alternaria leaf spot images ..... **Error! Bookmark not defined.**
Figure 3: Plant healthy image ..... **Error! Bookmark not defined.**
Figure 4: Leaf spot (early and late) images ..... **Error! Bookmark not defined.**
Figure 5: Rosette disease images ..... **Error! Bookmark not defined.**
Figure 6: Rust disease images ..... **Error! Bookmark not defined.**
Figure 7: ML and DL comparison 21
Figure 8: Convolutional neural network 22
Figure 9: VGG16 model architecture 24
Figure 10: VGG19 model architecture 25
Figure 11: Use case diagram 29
Figure 12: Sequence diagram 30
Figure 13: Database design 31
Figure 14: Overview of the dataset used for training ..... **Error! Bookmark not defined.**
Figure 15: Groundnut class images ..... **Error! Bookmark not defined.**
Figure 16 : CNN simple implementation 37
Figure 17: VGG16 implementation 37
Figure 18: VGG19 implementation 38
Figure 19: Software design architecture 40
Figure 20: App main page 41
Figure 21: Snap leaf page (A: The submitted image, B: Analysis of diseases) 42
Figure 22: User authentication code snippet. 43
Figure 23: Chatbot code snippet. 44
Figure 24: CNN accuracy and loss 45
Figure 25: CNN matrix 47
Figure 26: VGG16 accuracy and loss 48
Figure 27: Matrix of VGG16 50
Figure 28: VGG19 accuracy and loss 50
Figure 29: VGG19 matrix 52

LIST OF TABLES

Table 1: Types of groundnut diseases 15
Table 2: Number of images for each groundnut group. 34
Table 3: Hyperparameters 38

DEDICATION

To my beloved father, MOUSSA DEMBELE, who left this world before the completion of this work, whose love, guidance, and sacrifices continue to inspire me every day. This achievement is as much yours as it is mine. May your soul rest in peace.

ACKNOWLEDGEMENTS

This work was only made possible by some support and guidance. I want to express from the bottom of my heart:

To the West African Science Service on Climate Change and Adapted Land Use (WASCAL) for providing this wonderful capacity-building framework, and the German Ministry of Education and Research (BMBF) for funding this program.

To the Director of ED-ICC, **Pr. Amadé OUEDRAGO**, for all the efforts he is making for the success of this program.

To the Deputy Director, **Dr. Ousmane COULIBALY,** for his availability and constant support since the beginning of our program.

My heartly thanks to my Major Supervisor **Dr. Zakaria KIEBRE,** for his encouragement, support, and all he did during my work.

My sincere thanks to my Co-Supervisor, **Dr. Bouba TRAORE**, PhD, Agriculture and Climate Change, ICRISAT-Mali.

I would also like to thank the Director of ICRISAT**, Dr. Ayoni OGUNBAYO,** for his approval for my internship.

My sincere thanks to my colleague **Djara Olivier TINA**, who supported me during the internship period.

My sincere thanks to my Brother, **Tangara Boubacar Sidiki**, who supported me during the internship period.

I further extend my gratitude to the entire staff of ED-ICC and students of the school who contributed in diverse ways to the success of this project.

Thank you to my colleagues from all batches of ICC-EDICC Ouaga. With you, I learned a lot and especially learned a lot in myself. Thank you to **Miss Priscille Oladele FALEROUN** for always being there for me; thank you, dear host.

I would also like to thank my entire family, especially my mother, **Ms DEMBELE Diahara MAIGA,** and my Aunty **Ms SEREME Fatoumata MAIGA** for their unwavering support throughout my studies at this university.

Thank you to all those from far and near who contributed directly or indirectly to the facilitation and realization of these studies.

DECLARATION

I certify that the work for this study was carried out under the supervision of Dr. **Zakaria KIEBRE** and Dr. **Bouba TRAORE**. All care was taken in trying to make plain the subject and to put it into words that are easily understood.

Signed…….....

AISSATA DOUSSOU DEMBELE (STUDENT)

Signed……

Dr. Zakaria KIEBRE (MAJOR SUPERVISOR)

Signed…….

Dr. Bouba TRAORE (CO-SUPERVISOR)

ACRONYMS AND ABBREVIATIONS

App : Application

API: Application programming interface

BMBF: Federal Ministry of Education and Research

CNN: Conventional Neural Network

CWI: Centrum voor Wiskunde en Informatica

DL: Deep learning

ED-ICC: Ecole Doctorale d'Informatique pour les Changements Climatiques

FCM: Fuzzy C-Means

ICC : Informatique pour les Changements Climatiques

ICRISAT: International Institute for Crops of Semi-Arid Tropical

IDE: Integrated Development Environment

UI: User Interface

KNN: K-Nearest Neighbors

ML: Machine learning

MySQL: My Structured Query Language

LLM: Large Language Model

NARS: National Agricultural Research Systems

VGG: Visual Geometry Group

SVM: Support Vector Machine

TL: Transfer learning

RDBMS: Relational Database Management System

ReLU: Rectified Linear Unit

WASCAL: West African Science Service Centre for Climate Change and Adapted Land Use

ABSTRACT

Agricultural development faces significant threats from plant diseases, particularly in regions where climate change amplifies the spread of pests and pathogens. Traditional methods of detecting these diseases, such as visual observation of symptoms or sending samples to laboratories, can be time-consuming, costly, and difficult for small farmers to access. To help farmers to address these issues, this study aims to develop a mobile application that uses deep learning to identify and sort groundnut leaf diseases by analyzing pictures. The application uses three different deep learning models, CNN, VGG16, and VGG19, that were trained on a dataset from Mendeley with over 1,700 labeled images of healthy and sick groundnut leaves. Out of the learning models, VGG19 performed the best with 96% accuracy results for four main diseases and healthy categories: Alternaria, leaf spot, rosette, rust, and healthy leaves. The system is built with Flutter and offers features like logging in securely, working offline with SQLite for diagnosis, and a chatbot that helps farmers. It is designed to be simple to use, even in rural areas where the internet might be spotty. Images were pre-processed and augmented to improve the AI's performance, making the model better at handling different situations. Its accuracy was tested using metrics like precision, recall, F1-score, and confusion matrices, which all showed that the VGG19 model is dependable. This innovative tool gives farmers quick, real-time insights to help them make smarter decisions, cut down on crop losses, and support sustainable farming. By combining artificial intelligence with an intuitive mobile interface, the study shows how smart systems can boost climate-resilient and precise agriculture, helping farmers adapt to changing conditions more easily.

Keywords: Disease, deep learning, groundnut, mobile application, climate change.

RESUME

Développement d'une application permettant de détecter les maladies de l'arachide afin d'améliorer la prise de décision.

L'agriculture, pilier économique de nombreux pays africains, est confrontée à de graves défis liés aux maladies des plantes, amplifiés par les changements climatiques. Les méthodes traditionnelles de détection des maladies, basées sur l'inspection visuelle ou des analyses de laboratoires sont souvent inefficaces ou peu accessibles. Cette étude propose le développement d'une application mobile basée sur l'intelligence artificielle pour détecter les maladies foliaires de l'arachide à partir d'images. L'application exploite des modèles d'apprentissage profond tels que CNN, VGG16 et VGG19. Des modèles utilisés, VGG19 s'est montré plus concluant avec 96% de précision pour la détection des maladies courantes de l'arachide comme l'alternariose, la rouille, la rosette et la tache foliaire. Avec une interface conviviale, une base de données locale (SQLite), et un chatbot intégré, l'outil fournit un diagnostic en temps réel et des recommandations personnalisées. Cette solution innovante renforcera la résilience des agriculteurs face aux maladies, améliorera la productivité et favorisera une agriculture intelligente et durable.

Mots-clés: Maladie, apprentissage profond, l'arachide, application mobile, changement climatique.

Introduction

Context and justification

The majority of African economies are based mostly on agriculture (Banda, 2022). It is the largest contributor to Gross Domestic Product (GDP), the leading source of foreign exchange, generating about 40% of the continent's foreign currency earnings, and a vital driver of savings and tax revenues (Banda, 2022). Nevertheless, agricultural systems are constantly threatened by many abiotic and biotic stressors, including pests, pathogens, and climate change, which, if not managed properly, can result in tremendous yield loss (Gimenez et al., 2018). With agriculture as the backbone of most economies, challenges of plant diseases are further exacerbated under varying climatic conditions and limited access to modern agricultural technologies (Banda, 2022). Mitigating diseases that harm staple crops, including millet, sorghum, rice, and groundnut, is important for food security and economic stability. These crops, essential for subsistence farming and commercial agriculture, are susceptible to many diseases that can affect yield, putting farmers at risk of losing their livelihood and threatening the country's food supply (Appiah et al., 2024).

Plants are susceptible to numerous diseases, which may result in significant economic, social, and ecological damage (Saif & Nureize, 2023). Alterations in the surrounding environment due to climate change, such as temperature, precipitation, and soil nutrient levels, can make crops vulnerable to viral, fungal, and bacterial diseases. A major issue that all farmers face is recognizing plant diseases. One of the earliest methods employed by farmers to recognize plant diseases is visual inspection, where they analyze changes and effects in plant leaves to identify the type of disease (Gai & Wang, 2024). Nowadays, with technological progress, this method is considered outdated for an obvious reason: certain illnesses have the same symptoms. Plant diseases are divided into four groups according to the pathogen type: nematode, bacterial, viral, and fungal infections (Gai & Wang, 2024) with symptoms often similar. The species and varieties of host plants, the pathogenicity and virulence of pathogens, environmental factors like temperature, humidity, light, and soil type, as well as human factors connected to cultivation management practices, all have an impact on the incidence and frequency of plant diseases (Pokhrel, 2021). The main ways that pathogens spread among plants are by wind, rain, and insects. By eliminating the limitations of traditional image processing techniques, deep learning has completely transformed the field of plant disease diagnosis. The conventional image recognition approaches rely on manual feature extraction and classifier design, whereas the capacity of deep learning for automatic learning from raw data removes these requirements

(Mohaned, 2024). These functions have stirred significant interest in deep learning technologies for plant diseases the identification.

Many researchers have turned to transfer learning as a practical approach in deep CNN training, due to its computational complexity (Guo, 2023). Transfer learning architectures like VGG16, ResNet-50, VGG19, and Inception have become popular choices in the field. The ImageNet dataset, which includes more than a million training images, 50,000 validation images, and 100,000 test images, is usually used to pre-train these models (Munaf M.K and Karan 2023). The objective of this study is to build an AI-powered application capable of identifying groundnut diseases using image data.

Problem statement

Agriculture faces numerous challenges, including climate change and the spread of plant diseases (Nehemia et al., 2015). Crop production is greatly impacted by plant diseases, endangering global food security (Jonathan & Mahendranathan, 2024). It poses serious challenges to farmers because of gradual temperature increases, increased variability in annual rainfall patterns, and greater occurrences of extreme events such as drought and floods, hence food insecurity (Nehemia, K et al. 2015). Since agriculture is susceptible to changes in the climate, it is in fact seriously threatened by climate change, with consequences that could impact livelihoods, food security, and economic development (Kumar, 2016). Groundnut production is significantly affected by a wide diversity of diseases caused by fungi, viruses, bacteria, and nematodes. Among the most prevalent fungal diseases are early leaf spot (*Cercospora arachidicola*), late leaf spot (*Cercosporidium personatum*), and rust (*Puccinia arachidis*), which can cause up to 70% yield loss under severe infestation if unmanaged (Kankam et al., 2022). Groundnut rosette disease, a complex viral disease transmitted by aphids, is the most devastating in sub-Saharan Africa, with potential yield losses reaching 100% in susceptible varieties . Climate change is increasingly recognized as a major threat to groundnut (Arachis hypogaea L.) production, primarily by altering the patterns, incidence, and severity of plant diseases. Rising temperatures, changing rainfall patterns, and increased humidity create favorable conditions for many fungal pathogens, such as *Cercospora* spp., *Puccinia arachidis* (rust) . Additionally, climate-induced stress weakens plant immunity, making groundnut crops more susceptible to rosette disease, which is already a major constraint in sub-Saharan Africa (James et al., 2023). Overall, the interaction between climate change and disease dynamics not only reduces groundnut yield and quality but also threatens food security

and farmer livelihoods, particularly in vulnerable regions with limited access to adaptive technologies (Hunjan & Lore, 2020).

Visual inspection remains a widely used method for detecting diseases in groundnut fields, especially in low-resource settings (Melesse et al., 2023). However, this approach has significant limitations (Konate et al., 2020). It is subjective, often inaccurate, and highly dependent on the experience and expertise of the observer (Akram et al., 2018). Early-stage symptoms are frequently subtle or confused with abiotic stress, leading to misdiagnosis or delayed intervention (Melesse et al., 2023). Sometimes, disease detection based on visual inspection includes bias, misconceptions, and errors, making it difficult for inexperienced and young farmers. Consequently, the detection of plant diseases needs expert staffing (Chohan et al., 2020). It is also difficult to detect the symptoms and differentiate them from diseases due to shared symptoms, especially in standing water conditions in transplanted fields. Diseases directly lead to crop reduction and reduce the quality and value of agricultural products, increase agricultural inputs and labor costs, and affect farmers' income and livelihood (Türkoğlu & Davut, 2019). In addition, some plant pathogens (such as *Fusarium graminearum* and *Aspergillus flavus*) can also produce toxic secondary metabolites, polluting agricultural products and endangering human and animal health (Sonali et al., 2024). Moreover, manual inspection is time-consuming and not scalable for large farms, making it impractical for early warning systems (Chikezie et al., 2024). In contrast, innovative technologies such as artificial intelligence (AI), deep learning, and remote sensing offer powerful alternatives for precise, rapid, and large-scale disease detection (Chikezie et al., 2024). Image-based machine learning models, particularly convolutional neural networks (CNNs), have shown high accuracy in detecting and classifying groundnut diseases from leaf images (Sasmal et al., 2024). These tools not only reduce human error but also enable real-time monitoring and decision support, which are critical for timely disease management and improving productivity (Huang et al., 2023). Integrating such smart technologies into groundnut farming enhances resilience, especially in the context of climate change and rising disease pressures (Zhang et al., 2022).To address these challenges, the study proposes the development of an AI-driven application designed to detect diseases in groundnut is essential. By leveraging cutting-edge technologies, this solution aims to empower farmers with timely insights, enhance their decision-making in agriculture, and promote sustainable practices**.**

Research question, hypotheses, and objectives

Research questions

The main research question underlying this study is: How effective are deep learning-based image classification models in detecting and differentiating groundnut diseases compared to traditional visual inspection methods?

From the above main question, three specific questions are derived:

- What are the most effective deep-learning algorithms for detecting diseases in groundnut leaves?
- How does the application's accuracy compare with traditional disease detection methods?
- Can the application reliably diagnose multiple diseases across different plant species, such as groundnut?

Research hypothesis

An application leveraging advanced deep learning models and image processing techniques can accurately detect and classify groundnut disease, thereby enhancing decision-making for timely interventions in agriculture.

- Deep learning models such as VGG19 can achieve high accuracy in identifying groundnut diseases.
- The application can detect early-stage symptoms of diseases (anthracnose or leaf spot in groundnuts) more effectively than traditional visual inspection methods.
- Providing real-time disease diagnostics will enable farmers to make informed decisions, reducing crop losses and optimizing pesticide use.

Research objectives

Build an AI-powered application capable of identifying groundnut diseases using image data. To achieve the main objective, the following goals are need to be accomplished:

- Train deep learning models (VGG19, VGG16, and CNN) for disease classification with high accuracy.
- Validate the application's performance across diverse groundnut diseases.
- Develop a user-friendly interface for farmers to upload images and receive diagnostic results instantly, and recommendations.

Thesis structure

This thesis is structured into a general introduction, three main chapters, and a concluding section. The **introduction** outlines the overall context of the study and presents the problem statement, followed by the research questions, hypotheses, and objectives. **Chapter 1** presents

a comprehensive literature review, starting with an overview of groundnut, followed by an in-depth examination of existing approaches to plant disease classification. This includes both machine learning and deep learning techniques, which form the two core categories of the review. The chapter also discusses the influence of climate change on the spread and severity of groundnut diseases and highlights relevant related studies. **Chapter 2** outlines the materials and methods used in the research, including the study area, data sources, model training, and implementation techniques. Finally, **Chapter 3** presents the results of the study, supported by relevant figures and performance metrics, and provides a critical discussion of the findings in the context of the research objectives.

The thesis concludes with a synthesis of the main outcomes and general conclusions.

Chapter 1: Literature review

1.1. Overview of groundnut

The Fabaceae family contains the self-pollinating groundnut or peanut (Arachis hypogaea L. Millsp). Its seeds are the superior ones because they contain 35–56% oil, 25–30% protein, 9.5–19.0% carb, and different minerals like P, Ca, Mg, and K, as well as vitamins E, K, and B it is regarded as the most significant food legume crop in continental Africa due to its numerous applications in food, feed, paints, lubricants, and insecticides (Abady et al., 2019). The crop is used in a wide range of industrial products, including paints, lubricants, food, feed, and insecticides. The crop is used in a wide range of industrial products, including paints, lubricants, food, feed, and insecticides. It is regarded as the perfect crop for crop rotation and intercropping in small-scale subsistence farming systems due to its inherent capacity to fix atmospheric nitrogen in the soil. A lot of biotic and abiotic growth-limiting variables have caused production to fall short of its potential. Therefore, breeding to reduce or eliminate these limiting variables is essential for creating new cultivars that will guarantee higher yield and help secure food security (Suza & Lamkey Eds, 2025).

1.1.1. Crop biology

The self-pollinating peanut or groundnut (*Arachis hypogaea L. Millsp.)* belongs to the Fabaceae family. Groundnuts are disomic allotetraploid with 2n = 4x = 40. Except for the rare formation of the quadrivalent, there is minimal recombination between the A and B genomes due to the strong diplodization of the two sets of chromosomes in *A.* hypogaea. There are approximately 25 species with diploid genomes within the *Arachi*s group A. monticola, which is also tetraploid, and groundnut. Nine taxonomical divisions make up Arachis: Arachis, *Rhizomatosae, Extranervosae, Heteranthae, Trierectoides, Triseminatae, Caulorrhizae*, and *Procumbentes*. The Arachis section includes groundnut (Suza & Lamkey Eds, 2025).

(Bertioli et al., 2016)

Figure 1: An Arachis hypogaea plant

1.1.2. Geographical distribution of the crop

There are 68 species in the genus Arachis, which originated in South America (Alagirisamy, 2016). Due to their aboveground flowering and belowground seeding, Arachis species are easily distinguished from those of other closely related genera (Barkley et al., 2016). Geographical areas in Paraguay-Paraná, the upper Amazon, the Peruvian west coast, Brazil, and the southwestern Amazon region of Bolivia are six of South America's hotspots of groundnut diversity. Africa is also a secondary hub of diversity. It is thought by Holbrook and Stalker (2003) that Arachis hypogaea originated in the South American area comprising southern Bolivia and northern Argentina. A single hybridization event between two diploid *Arachis* species (the A genome from A. *duranensis* and the B genome from A. *ipaensis*) is believed to have created Arachis hypogaea approximately 4,000 years ago. The sterile hybrid then spontaneously doubled its chromosomes to create a fertile allotetraploid (AABB) (Zhang et al., 2022). The low allelic diversity found in contemporary cultivated peanuts is partly due to a strong genetic bottleneck that was created when the resulting allotetraploid was reproductively isolated from its progenitor species, even though its fertility was restored (Zhang et al., 2022).

1.1.3. Crop inflorescence

Simple or compound inflorescences (one to five blooms) can be found on both primary and secondary branches in the leaf axils of cultivated groundnuts, or Arachis hypogaea (Zhang et al., 2022). According to Holbrook and Stalker (2003), each inflorescence usually opens one

flower per day (Barkley et al., 2016). The flower has wing, keel, and regular petals. Deep orange to light yellow is the norm, yet it can occasionally be white (Leal-Bertioli et al., 2024). Five sepals are joined to the long hypanthium to form the calyx (Bertioli et al., 2016). Although flowers appear to be carried on stalks due to their elongated tubular hypanthium or calyx tube, they are actually categorized as sessile (Basuchaudhuri, 2022).

1.1.4. Importance of the crop

Groundnut, also known as Arachis hypogaea L., is an important leguminous oilseed crop that holds significant value, especially in tropical and subtropical areas (Dutta & Kumari, 2023). For a number of reasons, it is essential to agricultural systems (Dutta & Kumari, 2023). Groundnut plays a vital role in cropping systems, particularly in semi-arid tropical regions such as West Africa and South Asia, where it serves as both a food and cash crop (Qureshi, 2024). As a leguminous plant, groundnut contributes to soil fertility through biological nitrogen fixation, reducing the need for synthetic fertilizers and supporting sustainable agriculture (Ghafoor et al., 2024). When used in crop rotation or intercropping systems, groundnut helps break disease and pest cycles, thereby enhancing the productivity of subsequent crops such as cereals (Boubacar et al., 2020). It is also a relatively short-duration crop, making it suitable for integration into various cropping calendars (Boubacar et al., 2020).

Groundnut (Arachis hypogaea L.) is an economically significant crop, especially in developing countries (Pratap et al., 2024). It serves as a major source of income for millions of smallholder farmers, contributing substantially to rural livelihoods and poverty reduction (Pratap et al., 2024). Groundnut is traded both locally and internationally in various forms, raw kernels, processed snacks, oil, and cake, thus stimulating agro-industrial development and value chains (Sivaganesan, 2023). Additionally, the by-products, such as groundnut oil and cake, contribute to the livestock feed industry and food processing sectors, creating employment opportunities across the supply chain (Akram et al., 2018). The crop's adaptability to dry regions and its integration into mixed farming systems also enhance its economic resilience and relevance in climate-prone areas (Boubacar et al., 2020). Overall, groundnut plays a strategic role in ensuring economic stability and food security in many low- and middle-income countries (Sivaganesan, 2023).

1.1.5. Usages of crop

Groundnuts provide a very important food crop for subsistence in the tropics. The main reasons groundnuts are grown are for their meal, edible oil, vegetative residue, and kernels. According

to Abady et al. (2019), groundnut kernels typically contain 25–36% protein and 47–53% oil (Abady et al., 2019). For the rest, they contain roughly 10%–15 % carbohydrate, and P is significant. Groundnuts are also a good source of B and E vitamins. A variety of groundnuts can be used, such as groundnut oil, roasted and salted groundnuts, boiling or raw groundnuts, or as the paste sometimes referred to as groundnut (or peanut) butter. In some regions of West Africa, the tender leaves are added to soups as a vegetable. The crop's most valuable byproduct, groundnut oil, is utilized in both household and commercial settings. The extraction of edible oil accounts for almost 75% of the world's peanut production.

Groundnut oil is the most common and cheapest edible oil in India. It is used in food preparation, margarine and vegetable ghee manufacture, salad oil, deep-frying, shortening bread and pastry products, cosmetics and medicines, lubrication and emulsification in pesticides, and as fuel for diesel engines. The 40-50% protein press cake is used as a fertilizer and as a high-protein animal feed.

The dry pericarp of the mature Pod (these are often called shells or husks) can serve as fuel, as a soil conditioner, as a filler in fertilizers and feeds, or, by means of microorganisms that degrade the lignin, a compost (Abady et al., 2019). The leaves of the crop could be utilized as fodder and silage. Groundnuts provide more oil per hectare than any other food crop, allowing researchers to study the use of this crop as a bio-diesel crop within the context of the recent move towards bio-energy.

1.1.6. Disease constraints of crop

Groundnut mottle virus (*Potyviridae*), groundnut rosette virus, rust (*Puccinia arachidis*), early leaf spot (*Mycosphaerella arachidis*), and late leaf spot (*Mycosphaerella berkeleyi*) are the most dangerous diseases for groundnut (Huang et al., 2023).

1.1.7. Healthy leaf

As illustrated in Fig. 2, the images presented showcase examples of robust, healthy groundnut leaves. While a healthy groundnut leaf should appear to be green, normal-symmetrical shape, smooth, firm, have size as per normal, free from pests or diseases, and should be tightly attached to the plant. A healthy groundnut leaf shows that the plant is actively synthesizing and efficiently photosynthesizing, gas exchanging and transpiring. It also indicates that the plant has sufficient nutrients, water, and sunlight, and that it is not being stressed by disease, pests, or environmental factors. Healthy groundnut leaves play a crucial role in fostering the growth and development of the plant, ultimately leading to enhanced yields and superior crop quality.

By routinely monitoring these leaves for indications of pests, diseases, or nutrient deficiencies, and implementing timely management strategies, one can help maintain their health. This vigilance is essential for ensuring optimal performance and yield of the groundnut plants (Buddhadev et al., 2024).

Figure 2: Healthy leave image

1.2. Description of groundnut leaf diseases and symptoms

1.2.1. Alternaria leaf spot

Alternaria disease of groundnut [sample shown] in Fig. 3. Alternaria leaf spot is an important foliar disease caused by the *fungus Alternaria* species that affects groundnut leaves (Kumar, 2016). It is a widespread disease that can lead to severe yield losses when not managed. Symptoms of *A. tenuissima* are described as tan to dark brown discoloration of the apical segments of leaflets. *A. arachidis* produces brown, irregularly-shaped lesions surrounded by yellow haloes. The advanced stages of infection cause infected leaves to wither and become fragile while curling inward. *A. alternata* causes small, but pale water-soaked spots or lesions that spread across the leaves' surface. The lesions turn necrotic and brown, and are round to irregular in shape. In the advanced stage, the lesions occupy a large portion of the leaf surface

that leading to a premature defoliation and less photosynthesis, and a reduction of groundnut production.

(Sasmal et al., 2024)

Figure 3: Alternaria leaf spot images

1.2.3. Leaf spot (early and late)

The early and late leaf spot sample is displayed in Fig. 4. The majority of the groundnut crop is impacted by the fungal disease known as groundnut leaf spot (early and late) (Buddhadev et al., 2024). Early leaf spot is caused by the pathogen *Cercospora arachidicola*. If left unchecked, early leaf spot, a dangerous disease that infects groundnut plants, can drastically reduce crop yield. Groundnut early leaf spot symptoms often manifest as tiny, spherical, initially yellow or light brown spots that eventually develop dark brown or black on the plant's lower leaves. A yellow ring may encircle these spots, which may combine to produce larger lesions. As the illness worsens, the leaves may wither, turn necrotic, and finally fall off prematurely. Plants that are severely impacted may exhibit diminished pod development,

stunted growth, and significant production reductions. The late leaf spot is caused by the plant disease *Phaeoisariopsis*. It is one of the most dangerous groundnut diseases, and if left unchecked, it can result in large yield losses.

Similar to early leaf spot, the symptoms of groundnut late leaf spot usually show up on the lower leaves of the plant. On the other hand, late leaf spot lesions are frequently rounder and larger, with a diameter of 1 to 10 mm. The lesions begin as tiny, brown dots that progressively grow into necrotic lesions that range in color from reddish-brown to black and are either ringed with faint yellow or not. In severe situations, the lesions may combine to cover a significant area of the leaf. Significant yield losses can result from reduced photosynthesis, slowed growth, and poorer pod development caused by infected leaves that become chlorotic, wither, and prematurely defoliate.

(Sasmal et al., 2024)

Figure 4: Leaf spot (early and late) images

1.2.4. Rosette disease

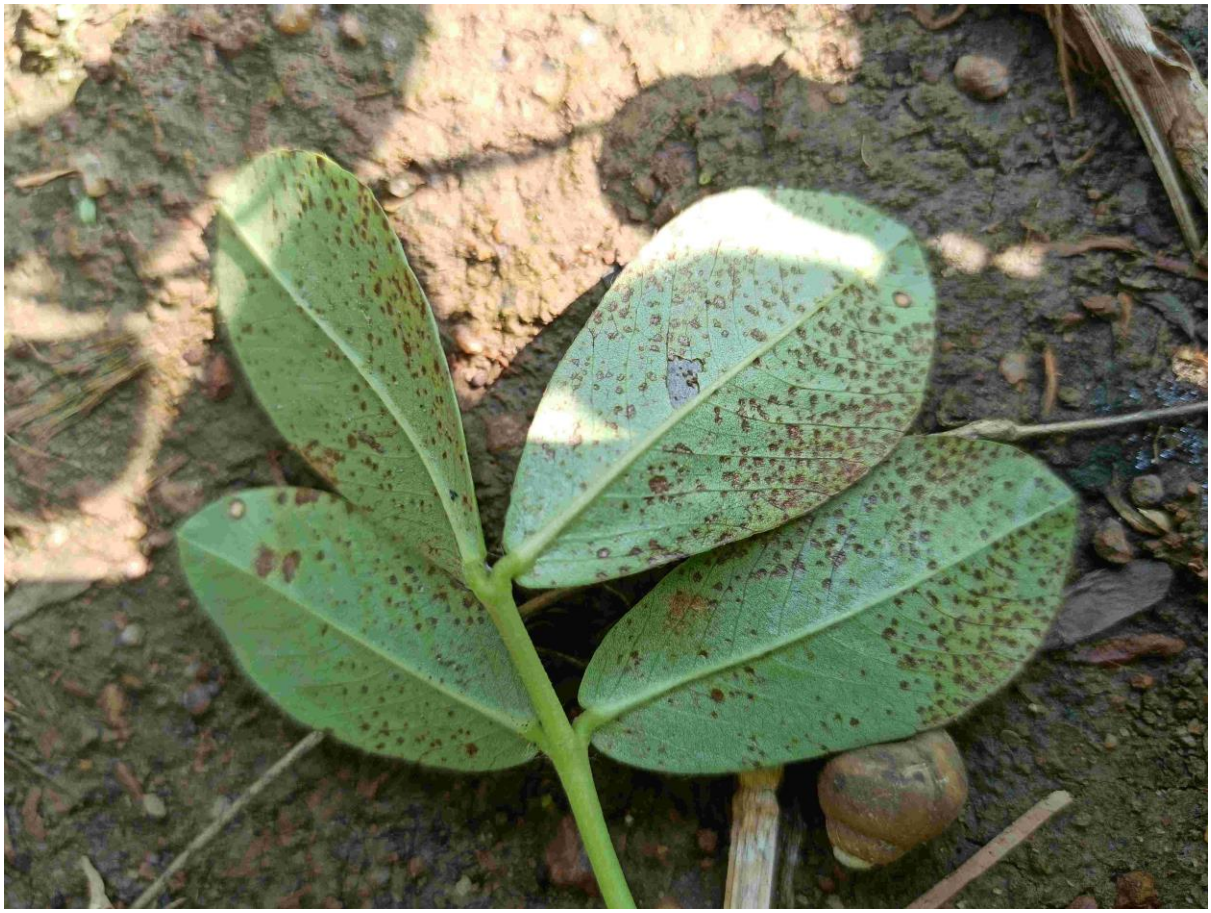
The groundnut rosette disease sample is displayed in Figure 5. The groundnut rosette virus is the cause of groundnut rosette disease also referred to as groundnut rosette viral disease, which infects groundnut leaves. It is a severe condition that can result in notable reductions in groundnut output yields (Kumar, 2016). Depending on the infection stage, groundnut rosette disease symptoms can change. Chlorosis and leaf curling, along with slowed plant growth, are possible early indications. With small, malformed leaves and reduced internodes, the diseased plants may have a distinctive "rosette" appearance that gives them a bushy, compact appearance. As the disease progresses, the leaves may show mottling, deformation, dieback, or necrosis. Reduced flower and pod development can result from the disease, and severely afflicted plants may not yield any viable pods at all, resulting in total crop loss.

(Sasmal et al., 2024)

Figure 5: Rosette disease images

1.2.5. Rust disease

The groundnut rust disease sample is displayed in Fig. 6, *Puccinia arachidis Speg.* , the scientific name for groundnut rust disease, is a fungal disease that primarily affects groundnut crops (Kumar, 2016). It is a prevalent disease that, if left unchecked, can significantly lower crop productivity. It spreads more easily in warm, humid weather. The disease targets every aerial part of the plant. When plants are around six weeks old, the illness is usually discovered. Dusty pustules that range in color from brown to chestnut appear on the underside of leaves. When the epidermis breaks, a large number of powdery uredospores are revealed. The sori are represented by tiny brown necrotic spots that show up on the upper surface of leaves. There are rust pustules on the stem and petiole. Brown teliospores show up as dark pustules among the necrotic areas toward the conclusion of the season. Lower leaves with serious infections shrivel and fall off too soon. The little, shrivelled seeds are the result of the severe illness.



(Sasmal et al., 2024)

Figure 6: Rust disease images

Table 1: Description of groundnut diseases

Disease selected	**Types**	**Symptoms**	**Disease attack duration**
Alternaria leaf spot	Fungal disease	On both leaf surfaces, tiny, water-soaked, chlorotic lesions are apparent. The leaves curl inward and become brittle.	5 –6 weeks after seed sowing
Leaf spot (Early and Late)	Fungal disease	Small, round, initially yellow or light brown specks that eventually turn dark black or brown and have a yellow ring around them are known as early leaf spots.	1 month after seed sowing
Rosette	Viral disease	The leaves may turn yellow and curl, and the plant's growth may be stunted until it takes on the distinctive "rosette" look.	2 –3 weeks after the seed sowing
Rust	Fungal disease	On the undersides of leaves, there are dusty pustules that range in color from brown to chestnut. In the necrotic areas, brown teliospores manifest as dark pustules.	After 6 weeks from the seed sowing

1.3. Impact of climate change on groundnut diseases spread and severity

Climate change is all about the long-term shifts we see in things like temperature, rainfall, wind patterns, and other aspects of our planet's climate system (Masson-Delmotte et al., 2021). It is drastically changing how plant diseases spread, which has big effects on farming around the world and the security of ecosystems.

Climate change significantly influences the spread and severity of groundnut diseases by altering environmental conditions that affect both pathogens and their vectors (Qureshi, 2024). Rising temperatures, increased humidity, and changes in rainfall patterns create favorable conditions for fungal and viral diseases, including early and late leaf spots (*Cercospora arachidicola* and *Cercosporidium personatum*) and rust (*Puccinia arachidis*) (Pratap et al., 2024). Higher temperatures and drought stress also predispose groundnut plants, which not only reduce yield but also pose serious food safety concerns (Pratap et al., 2024). Moreover, climate variability can expand the geographical range and prolong the lifecycle of disease vectors, such as aphids that transmit groundnut rosette disease, increasing the frequency and intensity of outbreaks (Chikezie et al., 2024). These climate-induced changes can result in more frequent epidemics, increased economic losses, and reduced effectiveness of traditional disease management practices (Chikezie et al., 2024). As a result, adapting groundnut production to climate change requires integrated approaches, including the development of climate-resilient and disease-resistant varieties, improved forecasting systems, and innovative management strategies (Qureshi, 2024).

The symptoms of the disease appear differently on both sides of the leaf. Example:

- **For Alternaria leaf spot**

Figure 7: The symptoms on both sides of the leaf

- **For leaf spot (early and late)**

(Nigam, 2014)

Figure 8: Symptom of leaf spot(early and late)

- **For rust**

(Nigam, 2014)

Figure 9: Symptom of rust on both sides

- **For rosette disease**

The disease is caused by a synergistic interaction between groundnut rosette assistor virus (GRAV), groundnut rosette virus (GRV), and satellite RNA (satRNA) associated with GRV(Nigam, 2014).

Groundnut rosette disease occurs in three main forms: chlorotic, green rosette, and mosaic rosette (Nigam, 2014). Each form presents distinct symptoms due to differences in causal factors such as various strains of satellite RNA (satRNA), the host plant's response, environmental conditions, and the possibility of co-infections (Nigam, 2014). Common visible

signs of the disease include stunted plant growth, shortened internodes, which lead to a bushy appearance, and smaller-than-normal leaves (Nigam, 2014).

(Nigam, 2014)

1.4. Machine learning (ML) techniques

Disease identification is a critical aspect of agriculture that requires significant attention. Although various methods have been developed and implemented to address this issue, the rapid and accurate detection of plant diseases remains a challenge (Kothari, 2018). The integration of machine learning has dramatically enhanced the ability to identify and classify plant diseases efficiently, offering a more effective solution to this ongoing problem.

1.4.1. K-nearest neighbors (KNN)

An approach for supervised machine learning pertaining to pattern identification and classification is called K-Nearest Neighbors (KNN) (Gurunathan et al., 2023). Despite being relatively easy to grasp, it stands out as one of the most straightforward machine learning algorithms, as it understands all the existing data points along with their corresponding class labels. When a new data point is added, KNN calculates its distance to other points in the training dataset using common distance metrics like Manhattan distance and Euclidean distance (Mensah et al., 2023). Using these measurements, KNN recognizes K neighbors that are closest to a new data point. In plant disease classification, KNN determines the result for a new plant sample by designating it with the class label most common among its K nearest neighbors.

If the majority are classified as diseased, the new sample will be considered diseased; if not, it will be deemed healthy (Mohaned, 2024)**.**

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

Using KNN, V et al. proposed a novel approach for automating the identification and classification of plant leaf diseases (Gurunathan et al., 2023). Their methodology integrates the KNN classifier with image processing techniques to enhance disease detection. The process involves several key steps: image preprocessing, contrast enhancement, RGB to grayscale conversion, feature extraction using the Gray Level Co-occurrence Matrix (GLCM), segmentation, and final classification using KNN (Mohaned, 2024). An accuracy of 96.76% was attained by the suggested model after it was trained on a dataset of plant leaf images that represented different disease kinds. The study introduced an innovative methodology for automating plant leaf disease detection and classification using KNN.

Amrita S. Tulshan and Nataasha Raul employed image processing and machine learning to identify several diseases from photos of plant leaves (Tulshan & Raul, 2019) by applying KNN classification and k-means clustering. They could predict plant leaf diseases with an astounding 98.56% accuracy rate. Additionally, their method offers extra data like the illness name, elapsed time, sensitivity, and afflicted region, which increases its usefulness for farmers.

1.4.2. Support vector machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm used for both regression and classification tasks. It is particularly effective for classification problems, especially in high-dimensional spaces and scenarios where the number of features exceeds the number of samples (Osisanwo et al., 2017). SVM works by identifying the optimal hyperplane that best separates different classes within the feature space. The primary objective is to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors.

SVM is categorized into two types: linear SVM and non-linear SVM.

- **Linear SVM** is used when the data is linearly separable, meaning a single straight line can divide the dataset into two distinct classes. In this case, SVM identifies the optimal hyperplane that maximizes the margin between the two classes (Shanmugam & Dharmar, 2024). This hyperplane is determined by the support vectors, which are the data points closest to the decision boundary.

- **Non-linear SVM** is applied when the data is not linearly separable, meaning a straight line cannot effectively classify the dataset (Shanmugam & Dharmar, 2024). In such cases, SVM utilizes the *kernel trick* to transform the original input space into a higher-dimensional feature space, where linear separation becomes possible.

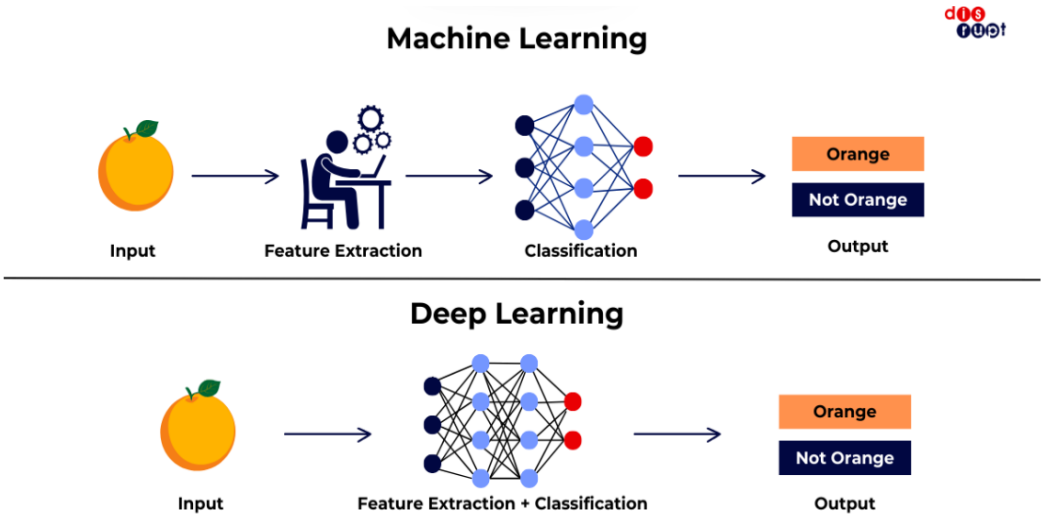
  A study by Dwi Ratna Sulistyaningrum, Alima Rasyida, and Budi Setiyono proposes an approach for identifying rice plant diseases using Support Vector Machine (SVM). The model classifies rice diseases into three categories: blight, blast, and brown spot (Sulistyaningrum et al., 2020). The methodology consists of several key stages, including image acquisition, preprocessing, segmentation using Fuzzy C-Means (FCM), and feature extraction based on color, texture, and shape attributes. The authors employed a multilevel SVM classification approach, specifically the one-against-all strategy, combined with different kernel functions. Among these, the linear kernel achieved the highest accuracy at 86.10%, outperforming both polynomial and radial basis function (RBF) kernels.

  An inventive strategy that combines sophisticated image processing techniques with Support Vector Machine (SVM) classifiers, enhanced by post-processing approaches to improve classification accuracy, is proposed in another study by Thyagaraj R, Satheesha T.Y, and Sathish Bhairannawar (Thyagaraj, R et al. 2023).

  The authors employed a multi-step procedure that started with image acquisition and proceeded to pre-processing stages like Otsu's segmentation and adaptive histogram equalization for efficient image enhancement and segmentation (Sahoo et al., 2024). Following feature extraction, important attributes necessary for illness classification are captured. Images of healthy and unhealthy plants were then separated using the SVM classifier. With an accuracy of 95.16%, the suggested model outperformed Vidyashree Kanabu's previous work, which used SVM, GLCM, and K-means clustering to obtain an accuracy of 90%.

1.5. Deep learning technique

A straightforward comparison of the two approaches is presented in Figure 7. Plant disease classification is one of the many machine learning tasks for which deep learning (DL) has emerged as a potent method. Unlike traditional machine learning methods, which require manual feature engineering where researchers must select and extract relevant features from raw data, deep learning models can automatically learn hierarchical representations directly from raw inputs. This capability eliminates the need for manual feature extraction, making the process more efficient and adaptable (Munaf & Karan, 2023).



(https://www.disruptignite.com/blog/deep-learning)

Figure 10: ML and DL comparison

1.5.1. Convolutional neural networks (CNN)

Plant disease classification is a particularly good application for Convolutional Neural Networks (CNNs), which have impressive performance in image-based applications (Jogin al. 2018). CNNs include several specialized layers, such as Convolutional, Activation, Pooling, Fully Connected, Batch Normalization, and Dropout layers, in contrast to conventional neural networks that are made up of only fully connected layers (Kim et al., 2022). Figure 8 illustrates this. CNNs can more reliably classify plant illnesses thanks to these layers, which allow them to extract intricate patterns and properties from photos automatically.

https://developersbreach.com/convolution-neural-network-deep-learning/

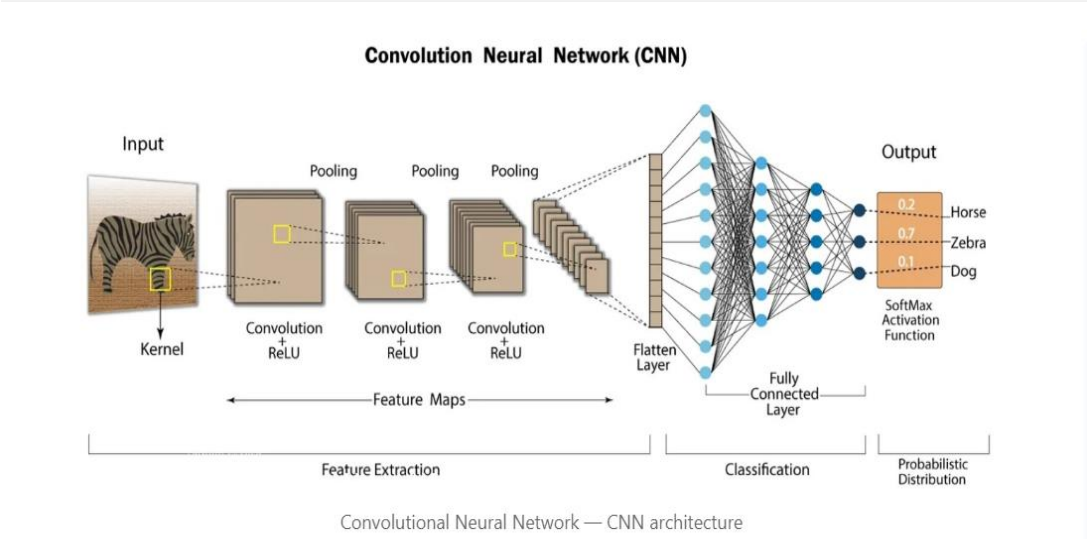


Figure 11: Convolutional neural network

- **Convolution layer:** The task of extracting features from the input image falls to the convolutional layer. It applies a filter or kernel to the input image to locate and extract particular characteristics, a process known as convolution.
- **Pooling layer:** The pooling layer decreases the spatial size of the feature maps produced by the convolutional layer. This down-sampling process helps decrease the feature map size, lowering computational complexity while retaining essential information.
- **Activation layer:** The activation layer applies a non-linear activation function, like the ReLU function, to the pooling layer's output. By adding non-linearity to the model, this function enables it to learn more intricate representations of the input data.
- **Fully Connected layer:** Every neuron in the fully connected layer, a typical neural network layer, is connected to every other neuron in the layers that come before and after it. Combining the traits that the pooling and convolutional layers have extracted is crucial since doing so finally enables the network to generate accurate predictions.
- **Normalization layer:** To make sure that each layer's activations are properly conditioned and avoid overfitting, the normalization layer carries out normalization procedures, including batch normalization and layer normalization.
- **Dropout layer:** By randomly deactivating a fraction of neurons during training, the dropout layer aids in preventing overfitting. The model performs better on unknown data thanks to this regularization strategy, which encourages the model to acquire more resilient and generic features rather than memorize the training data.

- **Dense layer:** Once the convolutional and pooling layers have extracted meaningful features from the input image, the dense layer integrates these features to generate the final prediction. The dense layer in a CNN is usually the last layer and is in charge of generating the output. The activations from previous layers are first flattened and then fed into the dense layer, where a weighted sum of the inputs is computed. An activation function is then applied to generate the final classification or regression output.

1.5.2. Transfer learning models

Transfer learning (TL) is a deep learning technique that enhances model performance by leveraging a pre-trained neural network for a new task (Shoaib et al., 2023). These models are initially trained on large datasets, often for tasks like image recognition or natural language processing, and are later fine-tuned using a smaller dataset specific to the new application (Gai & Wang, 2024). The core principle of TL is that a model trained to recognize patterns in one domain can be adapted to identify similar features in another, reducing the need for extensive labeled data and computational resources.

In recent years, various deep learning models, such as EfficientNet, MobileNet, VGG, DenseNet, Xception, and ResNet, have achieved top performance in ILSVRC competitions (Sahoo et al., 2024). These models are often trained on the ImageNet dataset, which consists of over 1.4 million images spanning 1,000 classes, making it one of the most widely used benchmarks in computer vision. This research focuses on combining two convolutional neural network (CNN) architectures, ResNet50 and VGG16, to enhance the classification of plant diseases (Islam, 2022).

1.5.3. VGG16 model

Figure 9 illustrates the architecture of the VGG-16 model. We initially selected the pre-trained VGG-16 convolutional neural network model in our experiments. To mitigate data overfitting, particularly due to the small size of our image dataset, we refined the model by freezing some of its layers.

In 2014, Andrew Zisserman and Karen Simonyan unveiled VGG-16, a deep neural network architecture with 16 convolutional layers. It processes input images with dimensions of (224 × 224 × 3) and utilizes fixed-size (3 × 3) convolutional filters. The network includes five max-pooling layers, each with a filter size of (2 × 2), distributed across the architecture.

At the top of the network, VGG-16 features two fully connected layers, followed by a softmax output layer. With approximately 138 million parameters, VGG-16 is a large and deep network,

leveraging multiple stacked convolutional layers to enhance feature learning and representation. (https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918).

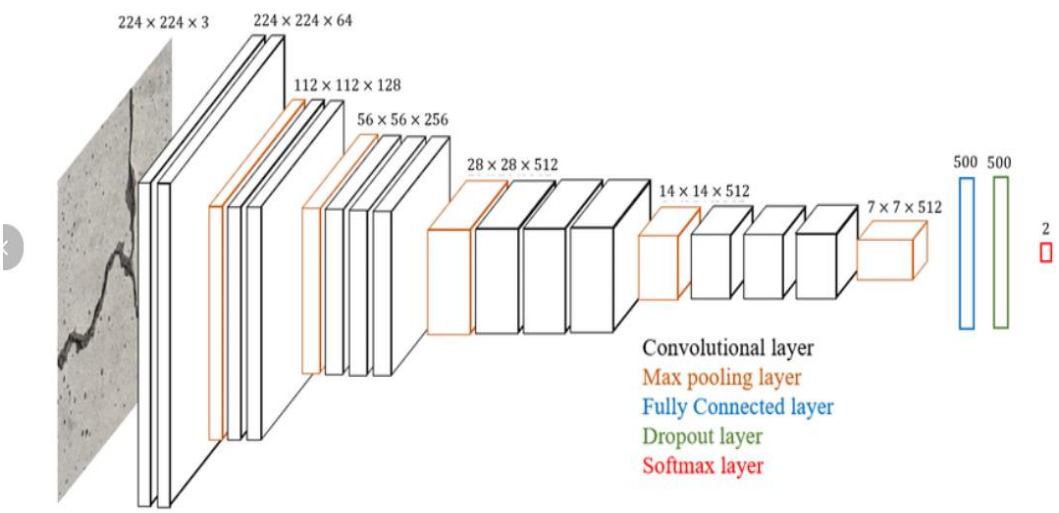
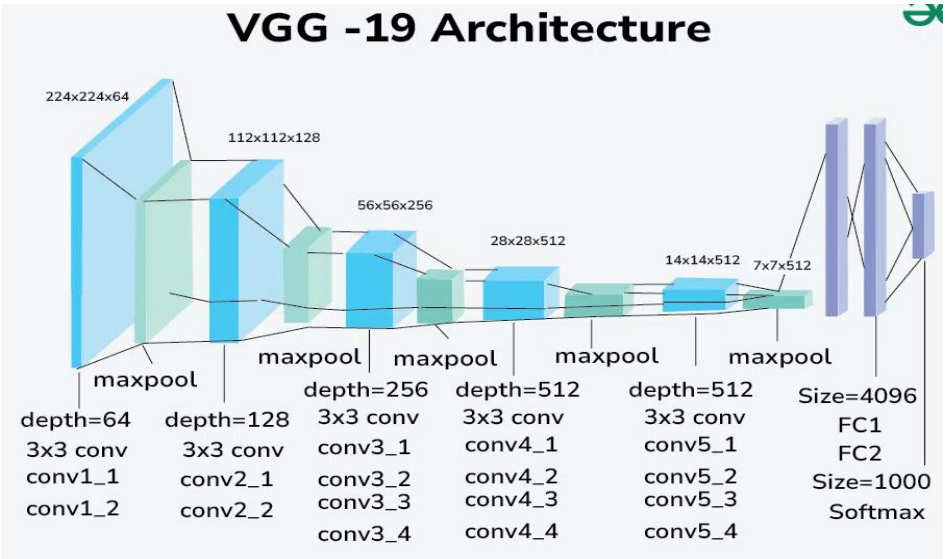


Figure 12: VGG16 model architecture

1.5.4. VGG19 model

Figure 13 show the VGG19 model architecture. The deep convolutional neural network VGG-19 has 19 weight layers, including 3 fully connected layers and 16 convolutional layers. The architecture follows a straightforward and repetitive pattern, making it easier to understand and implement (Chandra, B.R.K et al. 2023).

- **Convolutional Layers**: To maintain spatial resolution, use 3x3 filters with a stride of 1 and padding of 1.
- **Activation Function**: Each convolutional layer is followed by a Rectified Linear Unit (ReLU) to add non-linearity.
- **Pooling Layers**: To decrease the spatial dimensions, use max pooling with a stride of two and a 2x2 filter.
- **Fully Connected Layers**: The network's classification end has three completely connected layers.
- **Softmax Layer**: Class probabilities are output by the last layer.



(Chandra, B.R.K et al. 2023)

Figure 14: VGG19 model architecture

1.6. Plant disease detection software

PEAT develops the Plantix application in association with ICRISAT. This application comes with the tagline Your Crop Doctor, which is used for disease diagnosis and monitoring. This application provides a platform to global users with native information regarding best practices of disease and pest management, along with real-time diagnostics. Plantix functions on real-time diagnosis of the uploaded image of an infected plant and prescribes through image recognition technology. It also provides information about the weather as well as makes interactions between the farmers' community simple (Manish & Lalit, 2020). This program uses images to determine the diseases and then the plant health diagnosis is arrived at. It has entered the market first in the Indian regional languages of Telugu and Hindi. 30 crops and gives more prescriptions 120 plant diseases (Durga, 2020). The application uses plant images to differentiate diseases. A variety of such pictures is kept in a high-level cell and are coordinated with the picture in the worker for diagnosis. The most critical element of the Plantix application is the automated crop illness identification evidence. App analysis is based on images posted by ranchers of their toxic plants. Apart from known evidence of illness, app also recommends methods to minimize illness and offers useful information on the prevention of harvest illness in future seasons. The app also keeps a database of illnesses so farmers without internet network can also refer to it (Samal, I et al. 2023).

S. Shrimali is the author of an application called PlantifyAI, which aims to diagnose crop disease and give a treatment recommendation. The paper investigates and compares 16 convolutional neural network (CNN) architectures, evaluating various image processing filters for the PlantVillage Dataset, which includes 87,000 leaf images.

It belongs to 38 classes. MobileNetV2 with the Canny Edge Detection filter was chosen as the best model, with 95. 7% classification accuracy and an F1 score of 96.1. The application is very accurate and correctly diagnosed 46 out of 50 crop leaves during testing.

Ng et al. Mobile applications: For example, we introduced a mobile application for grape leaf disease detection and classification by employing deep learning object detection models (Ng et al., 2021).Faster R-CNN Object Detector with an efficient disease detection model based on the Inception-v2 backbone network. The application demonstrates promising results, achieving 97.9% accuracy in detecting grape leaf disease, entirely on a smartphone and without requiring a server connection.

Pan, et al. DenseNet. Proposes an intelligent diagnosis system for citrus disease. Developed in the WeChat applet, users can upload an image and get diagnostic output and comments (Pan et al., 2019). The model accuracy exceeded 88% in the following outcomes: detection of citrus diseases, and the time consumption has also been reduced by making the DenseNet structure even simpler. In addition, the paper also introduces dataset building, system architecture, as well as data augmentation. The proposed system can provide convenient services for citrus growers and bridge the gap between them and experts.

A study by Petrellis introduces a method for diagnosing plant diseases through mobile phone applications that operate without needing a connection to a remote server, making it ideal for use in the field (Petrellis, 2019). The focus of the study is on identifying citrus diseases by employing image processing techniques and feature extraction methods, including background separation and the identification of disease signatures. The research also utilized a classification method that employs a fuzzy-like approach to compare feature values with established disease signatures, extracting characteristics from images of plant parts such as leaves or fruits. These characteristics include color, relative area, and the number of lesion spots. Additionally, information like weather metadata is incorporated to help define the disease signatures. These signatures are developed using statistical analysis of representative training images and can be easily tailored by users according to their skill level. A key advantage of this approach is its resilience to variations such as orientation, scale, and image resolution, which improves its

applicability in various settings. The research thoroughly assesses the proposed method by utilizing images of citrus diseases taken in different environmental conditions. The report shows experimental accuracy between 70% and 99%, with acceptable accuracy often surpassing 90%. The methodology has been tested using standard smartphone cameras that have resolutions from 5 to 23 megapixels, highlighting its compatibility with typical smartphone hardware.

1.7. Software requirements

Requirement analysis involves identifying the necessary criteria that a developed system must fulfill, along with understanding user expectations for the proposed system (Mohaned, 2024). This includes functional and non-functional requirements, user requirements, and system requirements, all of which contribute to the overall system requirements.

1.7.1. Functional requirements

1.7.1.1. User authentication

Users will be able to create accounts and log in securely by entering their email and password, which will be sent to the application's server (Mohaned, 2024). The system will check the validity of the user's email and password before forwarding them to the server. After that, the server will confirm the user's credentials (Mohaned, 2024). If the credentials are valid, the system will direct the user to the main page; if not, it will inform the user that the credentials are incorrect (Mohaned, 2024).

1.7.1.2. Image upload

Users can upload images of plant leaves by either choosing them from their device's gallery or taking photos in real-time with the device's camera, providing flexibility and convenience for image acquisition (Sasmal et al., 2024).

1.7.1.3. Deep learning model integration

The system will incorporate a deep learning model specifically designed for detecting and diagnosing plant diseases in the uploaded samples (Syed et al., 2021).

1.7.1.4. Disease classification

The system will identify the diseases and provide the user with pertinent information, such as typical symptoms, causes, and suggested treatments (Syed et al., 2021).

1.7.1.5. Plant chatbot

Users can interact with a plant chatbot to get tailored advice and information on how to care for their plants (Syed et al., 2021). The chatbot should be able to respond to frequently asked questions, share helpful tips, and provide guidance on treating plants.

1.7.2. Non-functional requirements

1.7.2.1. Performance

The application needs to provide quick response times for image processing and disease detection to guarantee a smooth user experience (Dutta & Kumari, 2023). Both the mobile app and server should be operational at all times.

1.7.2.2. Reliability

The application must be extremely reliable, ensuring minimal downtime or service interruptions (Syed et al., 2021).

1.7.2.3. Scalability

The application needs to be scalable to handle a growing number of users and the images they upload over time (Syed et al., 2021).

1.7.2.4. Usability

The system should feature a user-friendly interface with clear navigation bars and clickable buttons, designed to visually indicate their interactivity to users (Syed et al., 2021).

1.7.2.5. Security

The system will implement strong authentication and authorization measures to safeguard sensitive information from unauthorized access (Mohaned, 2024). User credentials, such as passwords, will be securely hashed using industry-standard encryption methods before being stored in the database (Mohaned, 2024). Furthermore, access controls will be put in place to ensure that only authorized users can view or modify sensitive data.

1.7.3. User story

As a user, I should be able to securely create an account using my email and password, with the system ensuring my credentials are protected. Once logged in, I should be able to upload images of plant leaves either by selecting them from my device’s gallery or by capturing new photos using the device's camera. Upon submitting the image, the system should accurately detect and classify any diseases present in the plant leaves. The system should provide comprehensive details for each diagnosed condition, including common symptoms, causes, and recommended treatments, to help me address the issue effectively.

To further assist in plant care, the system should offer a plant chatbot that delivers personalized advice, answers to common plant-related queries, and tips for maintaining healthy plants. This chatbot will serve as a practical tool for getting real-time recommendations, fostering a deeper understanding of plant health and management.

1.7.4. Design

1.7.4.1. Use Case diagram

Figure 11 represents the use case diagram. Use Case Diagrams represent the system's functional requirements in terms of users (Mohaned, 2024). They represent interactions between outside actors (such as users and federations) and the system, explaining how users interact to achieve certain outcomes or perform certain operations (Mohaned, 2024). They work as a guide for visual prediction of how the system is going to act or behave and find its limits (Mohaned, 2024).

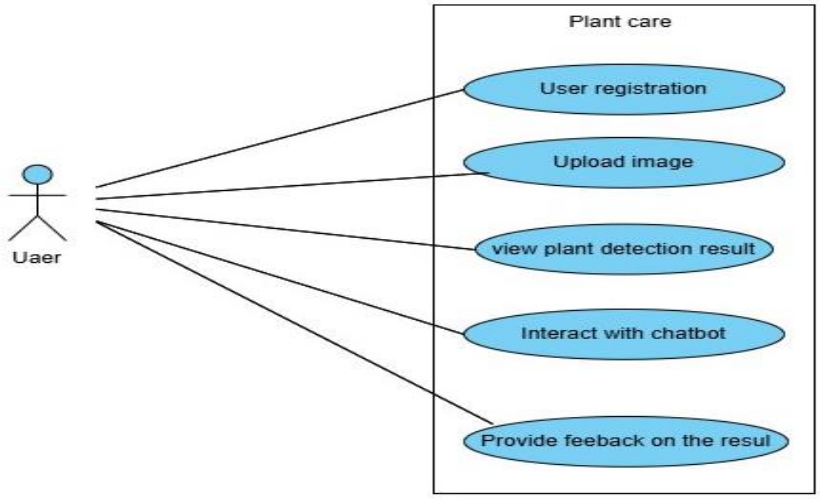


Figure 15: Use case diagram

1.7.4.2. Sequence diagram

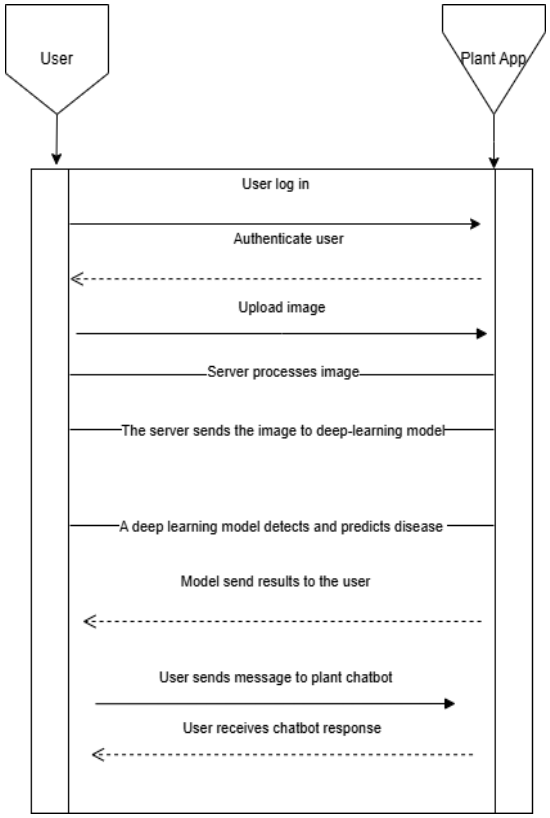


Figure 16: Sequence diagram

1.7.5. Database design

Figure 13 illustrates the database design. To ensure reliability and offline functionality, many mobile applications integrate local databases such as SQLite. SQLite is a serverless, transactional, zero-configuration, self-contained SQL database engine. It is embedded into mobile devices and widely adopted due to its low overhead and high performance (Junyan, Lv et al. 2009). In the context of Flutter applications, the sqflite plugin allows developers to leverage SQLite for persistent local storage (Junyan et al., 2009). This is particularly useful for storing user credentials, application settings, and historical data generated by the user.

Local databases are essential for managing both user data (login credentials, user preferences) and application data (analysis results, images) (Junyan et al., 2009). Local storage ensures

continuity of service when connectivity is absent, which is a common scenario in rural agricultural regions (Junyan et al., 2009).

User authentication and data tracking features enhance the personalization and reliability of agricultural applications (Junyan et al., 2009). By implementing a structured database schema, including users and analysis_history tables, developers can ensure that each user's data is securely stored and accessible for future analysis (Sjaak et al., 2017). This design supports accountability and traceability, as users can revisit previous diagnoses and monitor trends over time. Such features align with principles of precision agriculture and digital farming (Sjaak et al., 2017)

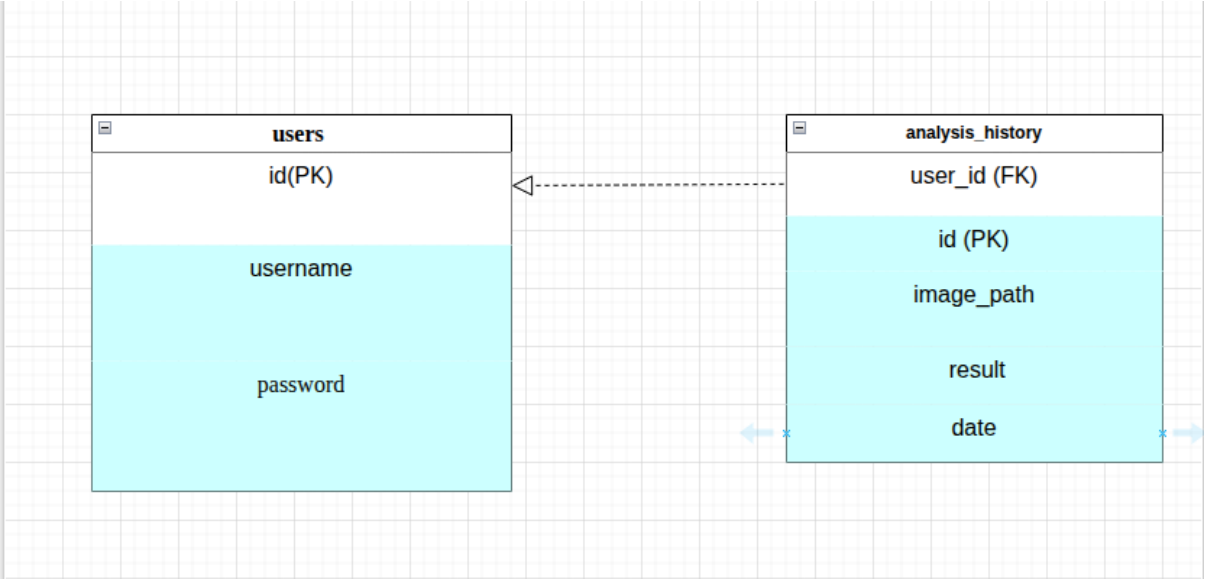


Figure 17: Database design

Chapter 2 Materials and methods

2.1. Materials

2.1.1. Study area

The ICRISAT research station (12°5'W, 8°5'N and 331 m altitude) is located 25 km southwest of Bamako, along the Kangaba road (Mohamed, 2024). The station lies within the geographical area of the village of Samanko (Mohamed, 2024). Samanko is a village in the rural commune of Mandé, in the sub-prefecture of Kati, in the Koulikoro region (Mohamed, 2024). The Samanko agronomic research station covers an area of 126 hectares.

The climate is Sudanian-Sahelian and has two seasons: a rainy season from May to October, and a dry season that is divided into two periods a cool period from November to January, and a hot period from February to April (Mohamed, 2024). The dominant wind is the Harmattan, which blows from east to west. Average annual rainfall ranges from 800 to 1100 mm (Mohamed, 2024).

The vegetation is characterized by the presence of large trees such as Néré (*Parkia biglobosa*), Shea tree (*Vitellaria paradoxa*), Tamarind (*Tamarindus indica*), N'golobè (*Combretum micrantum*), Mahogany (*Khaya senegalensis*).

Two types of soils are observed: leached tropical ferruginous soils, with a red profile and a silty-sandy texture on the surface and leached tropical ferruginous soils, with concretions and a silty-clayey texture in depth (Mohamed, 2024).

The local population is composed of Malinké, Bambara, Bobo, Senoufo, Dogon, and Fulani (Peulh) communities who live off agriculture, livestock, fishing, trade, and crafts (Mohamed, 2024).

So, after developing the application, it was tested in the groundnut field of ICRISAT.

2.1.2. Python programming language

Python is a computer programming language that was thought up and developed as a result of ABC. In the early 1980s, Van Rossum was employed at CWI (Centrum voor Wiskunde en Informatica), where he contributed to the development of the ABC programming language. In the late 1980s, while developing a distributed operating system called AMOEBA at CWI, Van Rossum began searching for a scripting language with ABC-like syntax that could also interact with Amoeba system calls. In order to overcome the limitations of ABC, Van Rossum himself started working on a new and minimal scripting language. Van Rossum began working on the

new scripting language in the late 1980s, and he released its first version in 1991. The module system implemented in this first release goes by the name Modula-Later on. That programming language had the name 'Python'(http://www.trytoprogram.com/python-programming/history-of-python/). Before we use Python, we need to make sure that it is not already installed in the computer because most Linux and UNIX distributions (and even some Windows systems) have a newer Python in their installation disk. We can make a quick check by simply typing Python on an open command line window an echo from Python interpreter. We will use Python 3.x in this research. to https://www.python.org/downloads/ Anaconda (https://www.anaconda.com/products/individual), an open-source programming language distributor supporting Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN.Anaconda is an open-source package management and environment that works with Windows, Linux, and macOS. Install Python upon downloading. A virtual environment is a standalone copy of Python that includes its own files, directories, and paths. This allows you to work on specific versions of Python and libraries without ruining other projects. To create environments and activate them, use the terminal commands below: conda create-- name myenv. Examples: conda create-- name eagles python. To activate your conda environment, use conda activate eagles-python. Jupyter Notebook (IPython): Jupyter Notebook is an open web application that makes it easy to create and share documents. With Jupyter Notebook, you can well combine live code, equations, visualizations, and narrative text. You can install jupyter notebook using command: conda install -c anaconda jupyter Other IDE's If you want to use a different IDE (Integrated Development Environment) than IPython you can use one of the numerous other Python IDE's, which give you an easy means to enter your code and display your data: Common IDE's:

Rodeo http://rodeo.yhat.com/

PyCharm https://www.jetbrains.com/pycharm/

Mendeley for References, Visual Studio Code for mobile application implementation.

2.2. Methods

2.2.1. Data collection

Collecting images of plant leaves is a challenging task due to the varying life stages that plant leaves undergo, which depend on the species and the characteristics of each plant type. Thus, the model is trained, validated, and tested using Mendeley data, an open-source dataset on plant diseases. Ten percent of the dataset was used for testing, twenty percent for validation, and seventy percent for training. A total of 1,720 4624 × 3472 pixel JPEG images comprise the

groundnut dataset. Each image is labeled with its name and image number. Five different folders or classes of data were uploaded to the repository: one for healthy data, one for early and late leaf spot, one for Alternaria leaf spot, one for rust, and one for Rosette diseases. Additionally, the name of each folder matched the image class that it belonged to. A collection of photos showing groundnut leaves in good health can be found in the HEALTHY folder. Images of early and late-stage leaf spots on groundnut leaves can be found in the LEAF SPOT (EARLY AND LATE) folder. Images of groundnut leaves infected with Alternaria leaf spot can be found in the folder named ALTERNARIA LEAF SPOT. Images of rust-infected groundnut leaves can be found in the RUST folder. The ROSETTE section contains images of groundnut leaves with Rosette disease. Since this dataset displays differences in plant traits like leaf shape, color, and texture, it can be used to identify and diagnose plant diseases. https://data.mendeley.com/ . Figure 15 illustrates the images of which classes.

Table 2: Number of images for each groundnut group.

Class name	**Number of images**
Healthy	600
Alternaria leaf spot	450
Leaf spot (Early and Late)	450
Rosette	100
Rust	120
Total	**1720**

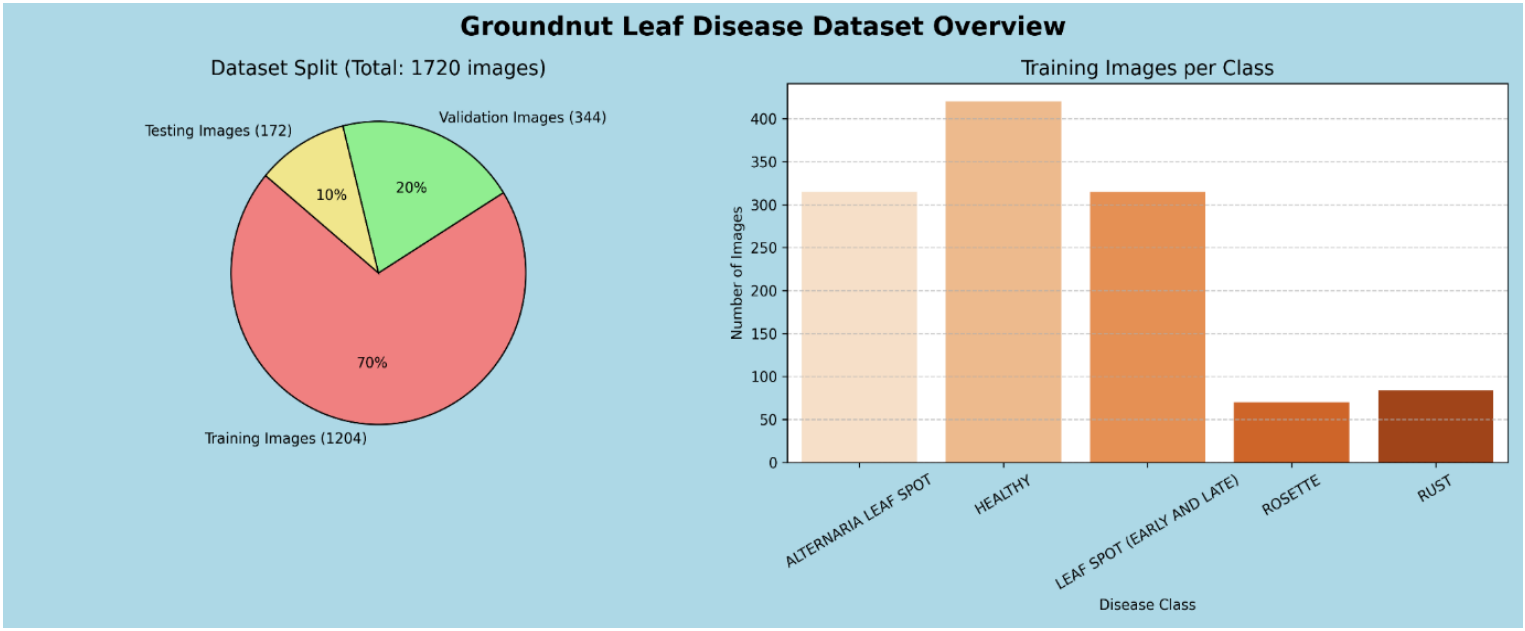


Figure 18: Overview of the dataset used for training

(Sasmal et al., 2024)

Figure 19: Groundnut class images

2.2.2. Image processing and analysis

To create a more accurate model, data processing and augmentation are required. The Keras in TensorFlow is utilized to implement image augmentation and preprocessing. The pictures are

resized to 224 x 224 as part of the preprocessing step. These images are then subjected to augmentation techniques, such as rotation, scaling, and flipping. The random rotation (0.2) function is used to apply rotation, rotating the images at a 20% angle in both clockwise and counterclockwise directions. The Rescaling (1/255) function is then used to apply scaling, guaranteeing uniformity in the input data scale to the input layer by normalizing pixel values between 0 and 1. The input images are then randomly flipped horizontally using RandomFlip, which adds variation by showing the model images from various angles without altering the image's semantic content. The model becomes more flexible to different image orientations, scales, and perspectives by directly incorporating these augmentation techniques into the model architecture, which improves the model's capacity for generalization in practical situations.

2.2.3. Different models used for training

Using some of the pre-trained convolutional neural network (CNN) architectures discussed in the literature review (VGG19, VGG16), apply transfer learning to develop deep learning models in this setting.

2.2.3.1. Model architecture

Figures 17,18, and 19 show the simple implementation of each model architecture. The CNN-based model used in this investigation was created especially to extract hierarchical features from plant disease. This allows it to efficiently record and examine trends and characteristics indicative of various crop diseases.

To adapt the CNN, VGG16, and VGG19 architectures for plant disease classification, the top layers used for ImageNet classification were removed. The output from the base network is then passed through fully connected (Dense) layers, each using the Rectified Linear Unit (ReLU) activation function. Each successive Dense layer has fewer nodes than the previous one, gradually reducing dimensionality. To prevent overfitting, a dropout layer follows each Dense layer, randomly dropping connections during training to help the model develop task-specific representations and enhance classification performance. The final Dense layer contains five nodes, corresponding to the four groundnut diseases and unhealthy classes in the Mendeley dataset. It utilizes a softmax activation function to generate a probability distribution across these classes. The model is implemented using the Keras API from TensorFlow and is designed to process input images of size (224, 224, 3), outputting a probability distribution for accurate multi-class classification.

```
model = Sequential()

model.add(Conv2D(filters=32,kernel_size=3,padding='same',activation='relu',input_shape=[224,224,3]))
model.add(Conv2D(filters=32,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))

model.add(Conv2D(filters=64,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=64,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))

model.add(Conv2D(filters=224,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=224,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))

model.add(Conv2D(filters=256,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=256,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))

model.add(Conv2D(filters=512,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=512,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))

model.add(Dropout(0.25))
model.add(Flatten())

model.add(Dense(units=1500,activation='relu'))
model.add(Dropout(0.4))
```

Figure 20 : CNN simple implementation

```
# Add preprocessing layer to the front of VGG
vgg = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)

# Freeze the pre-trained weights
for layer in vgg.layers:
    layer.trainable = False
```

[3] Python

```
2025-02-03 22:11:26.096504: E external/local_xla/xla/stream_executor/cuda/cuda_driver.cc:152] failed call to cuInit: I
```

```
# Useful for getting the number of classes
folders = glob('/home/aissata/Downloads/ICRISAT/groundnut-leaf-diseases/image_plant/train/*')
```

[4] Python

```
# Our custom layers - you can add more if desired
x = Flatten()(vgg.output)
prediction = Dense(len(folders), activation='softmax')(x)
```

[5] Python

```
# Create the model object
model = Model(inputs=vgg.input, outputs=prediction)
```

Figure 21: VGG16 implementation

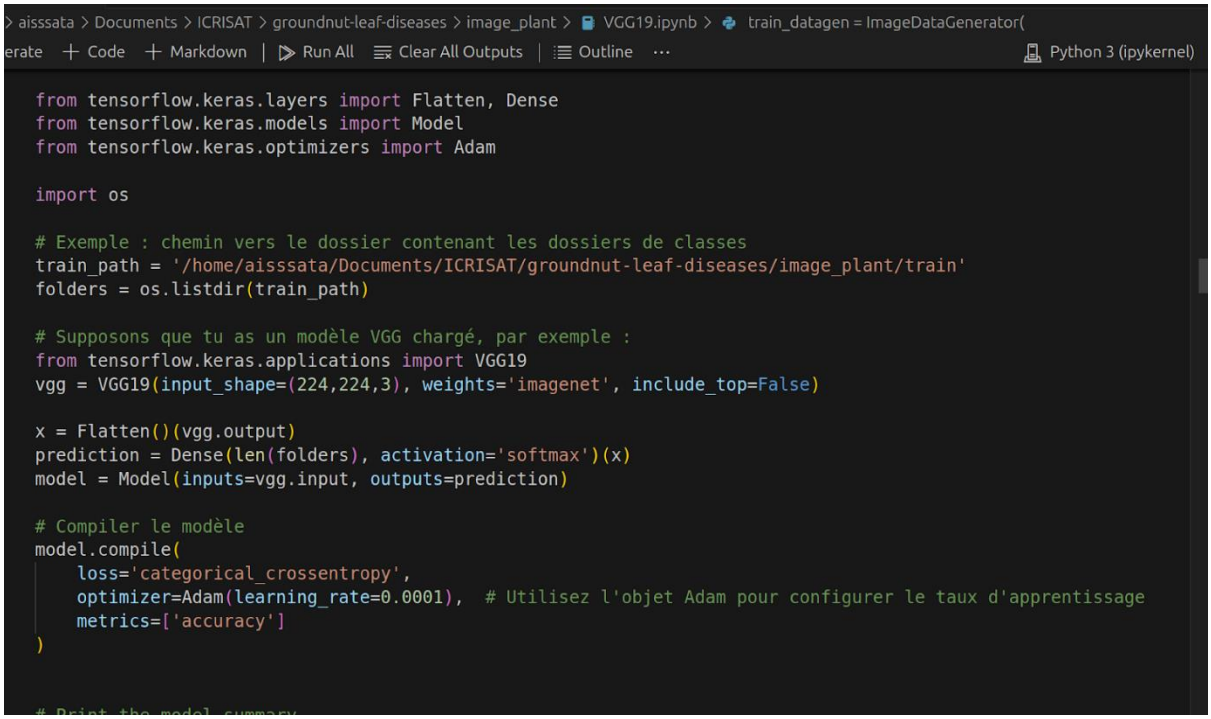


Figure 22: VGG19 implementation

2.2.3.2. Model training and hyperparameters

The Adam optimizer was used to train each architecture for 50 epochs with a batch size of 32 at a learning rate of 0.001. The next step was fine-tuning, which involved unfreezing every layer in the basic model so that all weights, including those frozen during feature extraction, could be changed. All layers, except the final five, were then frozen to prevent overfitting and preserve important learned properties. This made sure that just the last layers, which are in charge of identifying patterns unique to a given task, were modified. After adjusting the model structure, each architecture was recompiled using the Adam optimizer with a reduced learning rate of 0.0001 to promote stable training. Fine-tuning was then conducted for five more epochs, leveraging the knowledge gained during the initial training phase. The table 3 provides a detailed summary of the hyperparameters used.

Table 3: Hyperparameters

Parameters	**Description**
Classes	5
Batch Size	32
Image Dimensions	224 x 224 pixels
Activation Function	ReLu
Epochs	50
Loss Function	Categorical Cross Entropy
Optimizer	Aam
Learning rate	0.001 in the initial complication, 0.0001 in fine-tuning

2.2.3.3. Model evaluation

The effectiveness of plant disease detection and diagnosis models is usually measured using a variety of performance metrics. The mathematical formulas used to assess the model's performance are introduced in this section.

2.2.3.4. Confusion matrix

A key evaluation metric in machine learning, the confusion matrix offers a thorough analysis of a classification algorithm's performance. Each row in the matrix represents a predicted class, whereas each row corresponds to a predicted class. The number of samples that the model correctly or incorrectly classified is shown in the matrix's cell (Mohaned, 2024). The confusion matrix was used to calculate the following:

Accuracy: Measures the proportion of correctly classified instances out of the total number of samples. In terms of mathematics, it is expressed as:

$$\textbf{Accuracy} = \frac{\textbf{True Positive} + \textbf{True Negative}}{\textbf{True Positive} + \textbf{True Negative} + \textbf{False Positive} + \textbf{False Negative}}$$

Precision: The percentage of correctly classified positive instances relative to all predicted positive instances is measured. In terms of mathematics, it is expressed as:

$$\textbf{Precision} = \frac{\textbf{True Positive}}{\textbf{True Positive} + \textbf{False Positive}}$$

Recall: Calculates the percentage of accurately identified positive cases relative to all positive cases. In terms of mathematics, it is expressed as:

$$\textbf{Recall} = \frac{\textbf{True Positive}}{\textbf{True Positive} + \textbf{False Negative}}$$

F1-Score: Is the precision and recall harmonic mean. In terms of mathematics, it is expressed as:

$$\textbf{F1-Score} = \frac{2(\textbf{Recall} * \textbf{Precision})}{\textbf{Recall} + \textbf{Precision}}$$

2.2.4. Software design architecture

Figure 19 show the design architecture. The proposed system is designed as a mobile-based application for plant disease detection using deep learning. It adopts a modular architecture comprising four key components: the Frontend User Interface (UI), the Backend API, the Machine Learning (ML) Model, and a Database.

The Frontend, built with Flutter, allows users to submit plant photos and access diagnostic outcomes. The Backend API, ideally designed with Flutter for compatibility with ML

frameworks, acts as the main controller. It gathers user inputs from the frontend, sends them to the ML model for evaluation, and sends the predictions back to the frontend. The ML Model, designed to identify plant diseases from pictures, handles incoming data and produces classification outcomes. The Database archives user interactions, image metadata, and prediction logs, enabling future analysis and enhancements to the system.

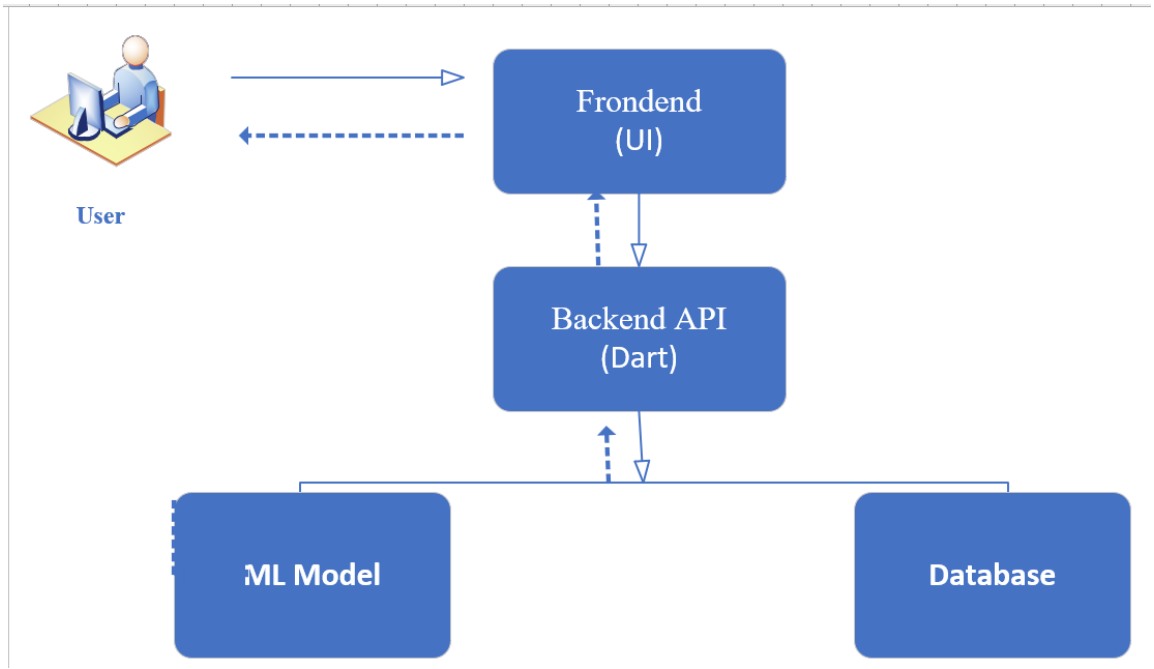


Figure 23: Software design architecture

2.2.5. Application development

2.2.5.1. Frontend development

The Flutter Dart framework was used for building the mobile application. As illustrated in the figure 20 , it has five primary pages and elements: user authentication, analysis, history, chatbot, and the main page.

To begin using the application's functions, the user can either capture an image of a plant or upload one immediately from the home page. As an alternative, the user can decide to register or log in by going straight to the authentication page. During the registration procedure, a username and password are entered to create an account. Following authentication, the user can access personalized features like recommendation tracking and diagnostic history, as well as continue with picture analysis.

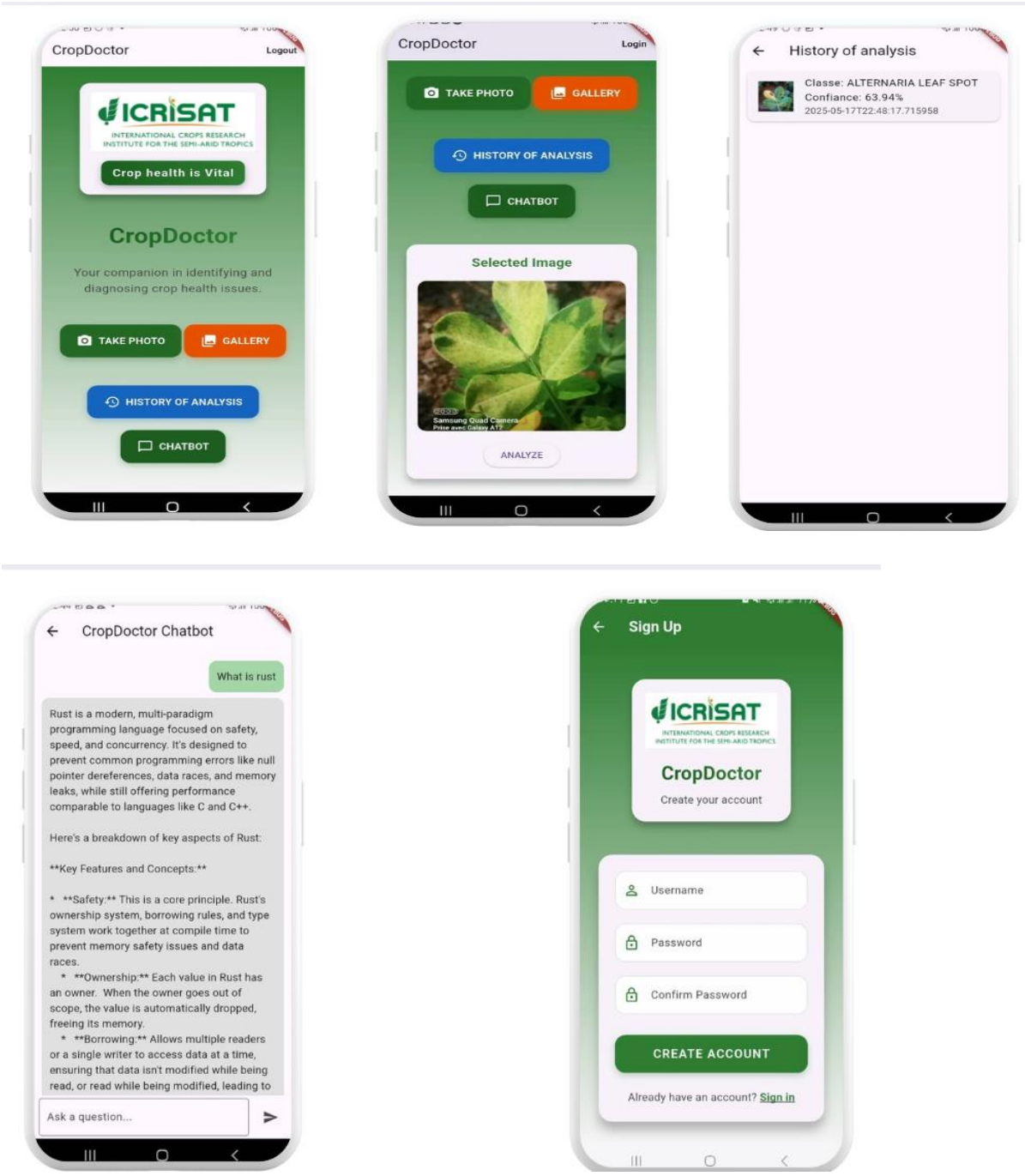


Figure 24: App main page

Once the user takes or uploads a photo on the main page, they are directed to the Diagnosis and Recommendation section. On this page, the system examines the image and offers a diagnosis of the identified plant disease. According to the diagnosis, tailored suggestions are provided to assist the user in handling or addressing the problem.

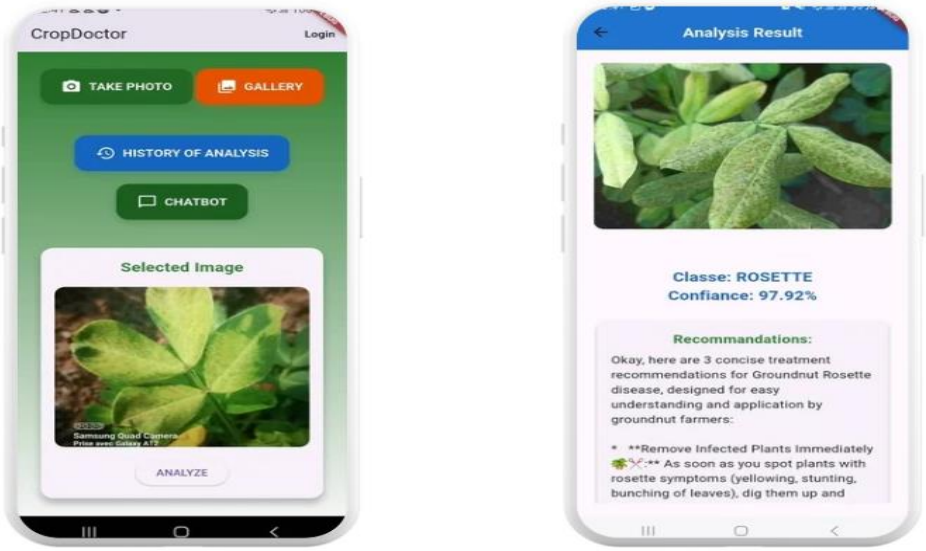


Figure 25: Snap leaf page (A: The submitted image, B: Analysis of diseases)

To view past diagnoses and associated recommendations (analysis history), the user is required to create an account or sign in. This guarantees that all diagnostic information and suggestions are safely kept and available to the user over time, allowing improved supervision and decision-making for their crops.

The Chatbot section provides users with a specific platform to engage with a plant-oriented virtual assistant directly through the mobile app. This function allows users to obtain information about plants, pose inquiries, and get immediate help on subjects like disease signs, treatment recommendations, and optimal farming methods.

The chatbot's implementation and features, along with its architecture and integration, will be elaborated upon in the Back-End Development section.

2.2.5.2. Back-End development

Figure 23 shows the user authentication code snippet. The application's backend is developed locally utilizing an SQLite database, which is managed via the lib/db_helper.dart file. This file outlines the database structure, offers methods for data access, and contains the business logic needed for user authentication and the preservation of analysis history. Importantly, there is no distinct remote backend server or external API utilized; all backend features are entirely integrated within the application.

The Flutter framework and the Dart programming language were used to create the user registration interface, making it dynamic and interactive. Because of its stateful widget design, the user interface can be updated in real time in response to asynchronous processes and user

interaction. Each of the form's input fields password, username, and password confirmation has validation logic to guarantee data integrity**.**

The system checks for consistency in the password and validates the inputs after submission. A helper class called DBHelper isolates the database operations and stores user credentials in a local database if they are valid. To improve the user experience, the program offers feedback features including loading indicators and success and problem notifications. Within the Flutter environment, this architecture encourages a clear division between data persistence, logic handling, and interface design.

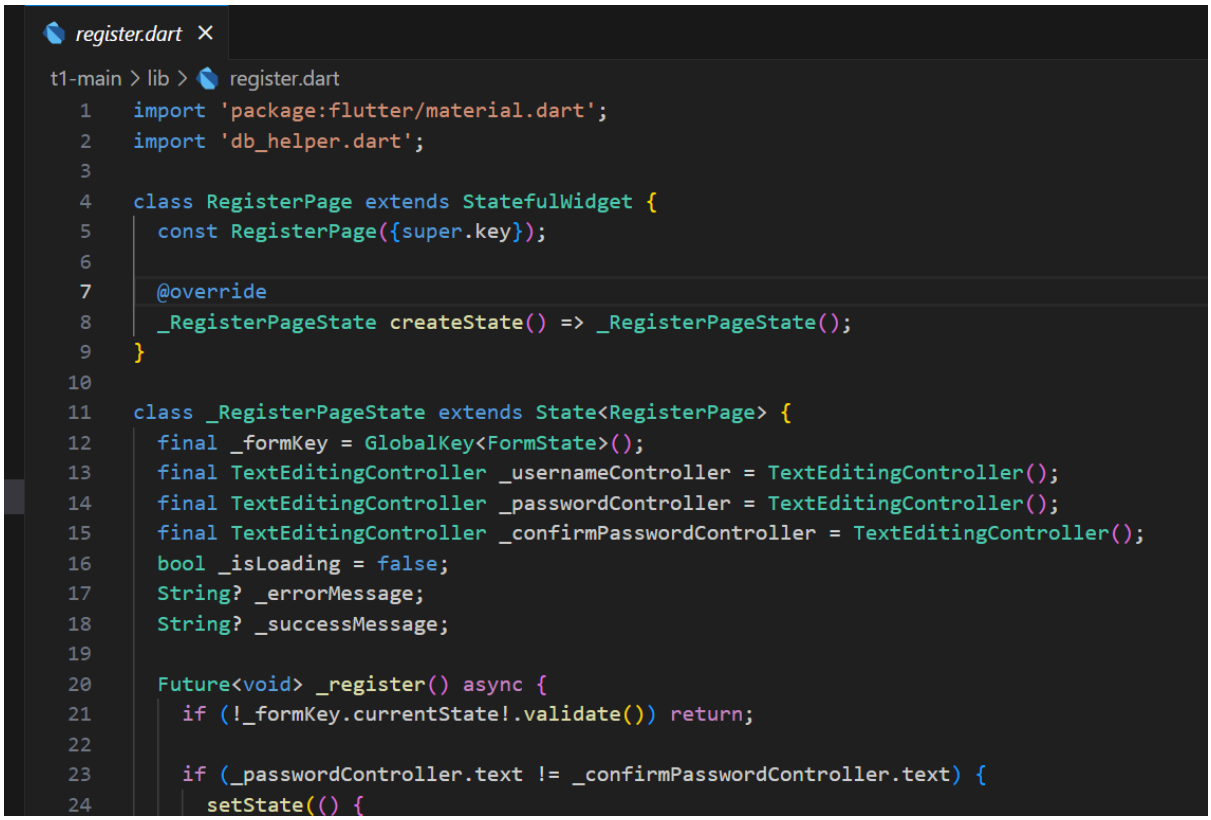


Figure 26: User authentication code snippet.

2.2.5.3. Chatbot implementation

Figure 24 shows a Chatbot code snippet. The mobile application's chatbot module was created using the Dart programming language and the Flutter framework, emphasizing the incorporation of smart conversational features into a cross-platform setting. The execution utilizes Google's Gemini API (gemini-2.0-flash) through the google_generative_ai Dart package, facilitating natural language comprehension and creation. To provide secure and adaptable access to API credentials, the flutter_dotenv package was employed for managing environment variables.

The chatbot interface is designed as a stateful widget, enabling dynamic updates to the UI based on user interactions and asynchronous communication with the backend model. The layout features a vertically scrollable list that shows messages in a dialogue style, clearly differentiating between user entries and AI replies. A text entry field, along with a submit button, collects user inquiries, which are subsequently processed and sent to the Gemini model using the generateContent() method. The AI-generated content appears in real time in the chat window. Internally, the dialogue is preserved as a series of organized message objects, contained within a unique ChatMessage class, which monitors both the content and its origin (user or bot). This method guarantees modularity and facilitates future expansion. A loading indicator appears during inference to enhance user feedback and ensure interactivity.

We can say that this implementation demonstrates how large language model (LLM) APIs can be successfully integrated into mobile environments using Flutter, enabling the development of intelligent, context-sensitive virtual assistants that can help users make decisions in fields like education, healthcare, and agriculture.

chatbot.dart

t1-main > lib > chatbot.dart

```dart
import 'package:flutter/material.dart';
import 'package:flutter_dotenv/flutter_dotenv.dart';
import 'package:google_generative_ai/google_generative_ai.dart';

class ChatbotPage extends StatefulWidget {
  const ChatbotPage({super.key});

  @override
  _ChatbotPageState createState() => _ChatbotPageState();
}

class _ChatbotPageState extends State<ChatbotPage> {
  final TextEditingController _controller = TextEditingController();
  final List<_ChatMessage> _messages = [];
  bool _isLoading = false;
  late GenerativeModel _model;

  @override
  void initState() {
    super.initState();
    _initModel();
  }

  Future<void> _initModel() async {
```

Figure 27: Chatbot code snippet.

Chapter 3 Results and discussion

3.1. Results

This section contrasts the outcomes of employing three distinct models, specifically two pre-trained transfer learning CNN architectures, to categorize and predict plant diseases: CNN Simple, VGG16, and VGG19. These networks, along with test scenarios for the server and mobile application, were analyzed and contrasted using the metrics discussed in the Model Evaluation chapter.

3.1.1. CNN model training

In our proposed learning classifier, the first deep learning model used is a CNN model. With a batch size of 32, the desired image size is set to (224,224). The softmax activation function for model training was employed, and the Adam optimizer was used to optimize the categorical cross-entropy loss function. After 50 epochs of training, the model's accuracy was 85%. Figure 24 displays the training and validation loss and accuracy graphs.

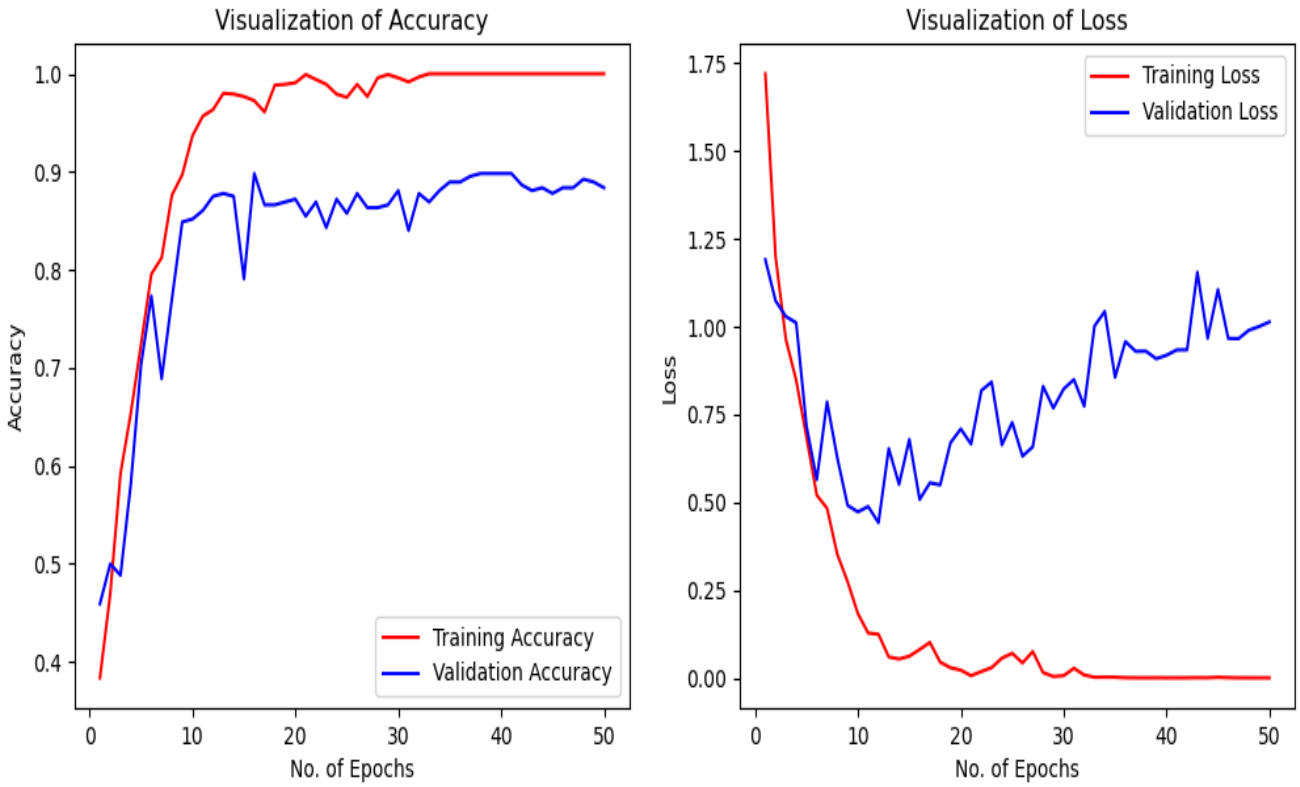


Figure 28: CNN accuracy and loss

3.1.2. Confusion matrix of CNN

The confusion matrix (Figure 25) offers an understanding of the predictive effectiveness of the suggested plant disease classification model across five categories: Alternaria Leaf Spot, Healthy, Leaf Spot (Early and Late), Rosette, and Rust. The evaluation employs standard classification metrics such as True Positives (TP), False Positives (FP), False Negatives (FN), and consequently, True Negatives (TN), to assess the model's distinguishing capability. In the Healthy class, the model records a significant number of True Positives (TP = 57), with a small count of False Negatives (FN = 3) and False Positives (FP = 11). This suggests the model successfully recognizes healthy plant samples while keeping a low error rate for diseased ones. The Alternaria Leaf Spot category demonstrates robust performance with TP = 40, FN = 5 (primarily misclassified as Leaf Spot), and low FP, indicating good sensitivity and specificity for this category.

Conversely, the Leaf Spot (Early and Late) category shows an increased count of False Negatives (FN = 11), mainly because of being incorrectly classified as Healthy (8 occurrences). This suggests that the model sometimes does not recognize the disease even when it is truly there. The TP for this class is 34, indicating a moderate level of effectiveness, yet there is potential for enhancement in differentiating early-stage disease signs from healthy tissue.

In the Rosette class, the model produces a lower TP (7) and a higher FN (3), along with some FP, indicating difficulties in accurately recognizing this condition. Likewise, the Rust class shows moderate performance with TP = 8, FN = 4, and FP arising from confusion with Rosette and Leaf Spot.

In general, the model attains excellent classification accuracy for clearly defined categories such as Healthy and Alternaria Leaf Spot, but demonstrates less sensitivity toward visually comparable diseases like Leaf Spot and Rosette. These results indicate a requirement for additional refinement, especially in feature extraction and class distinction, to diminish FN and FP rates in the more unclear categories.

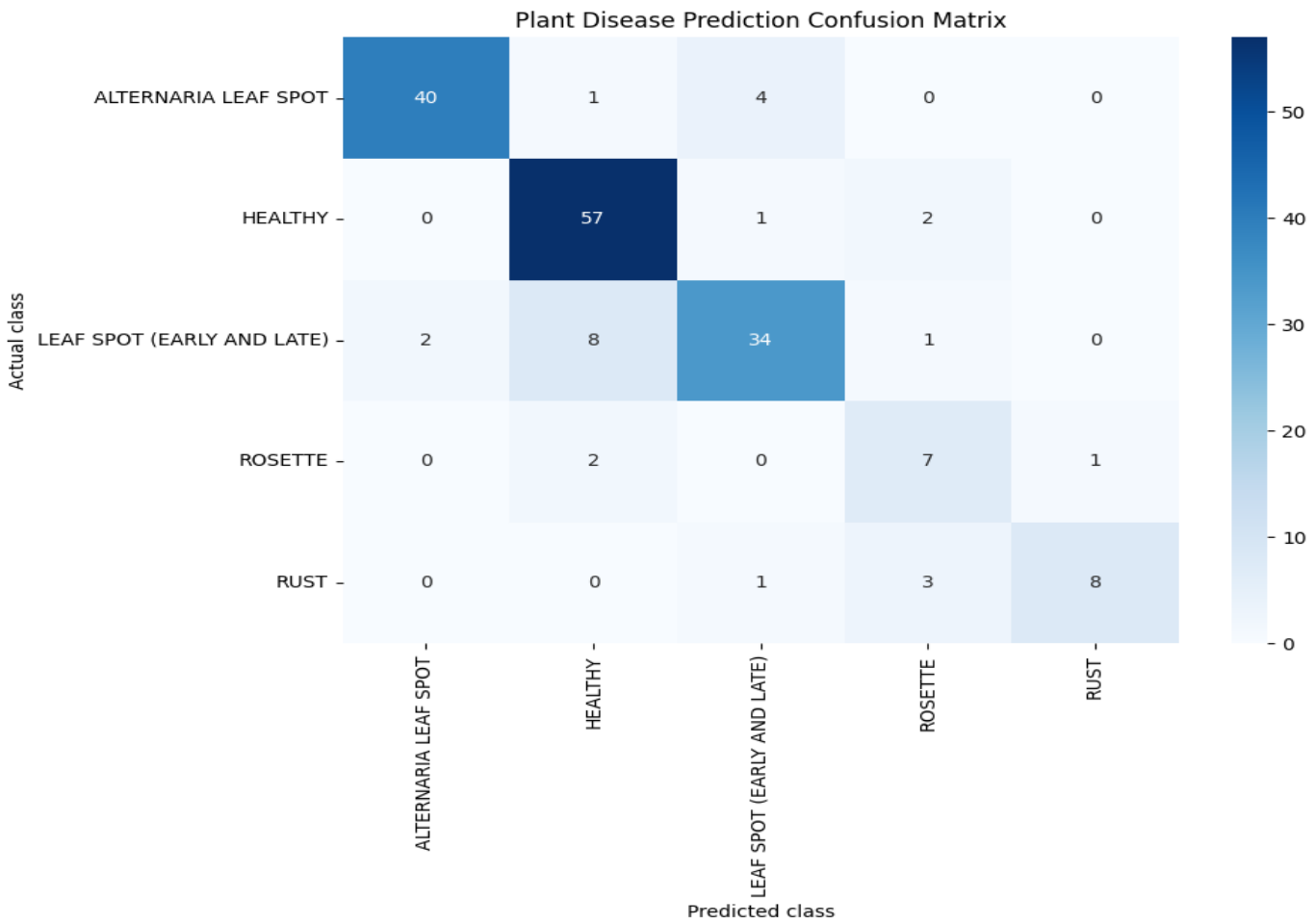


Figure 29: CNN matrix

3.1.3. VGG16 model training

Convolutional neural networks (CNNs) based on VGG16, like the second model. It was trained on 224 x 224 images with a batch size of 32. This model features an Adam optimizer with

cross-entropy loss and is implemented with a softmax activation. The model achieved over 93% accuracy with very low loss after 50 epochs, as shown in Figure 26.

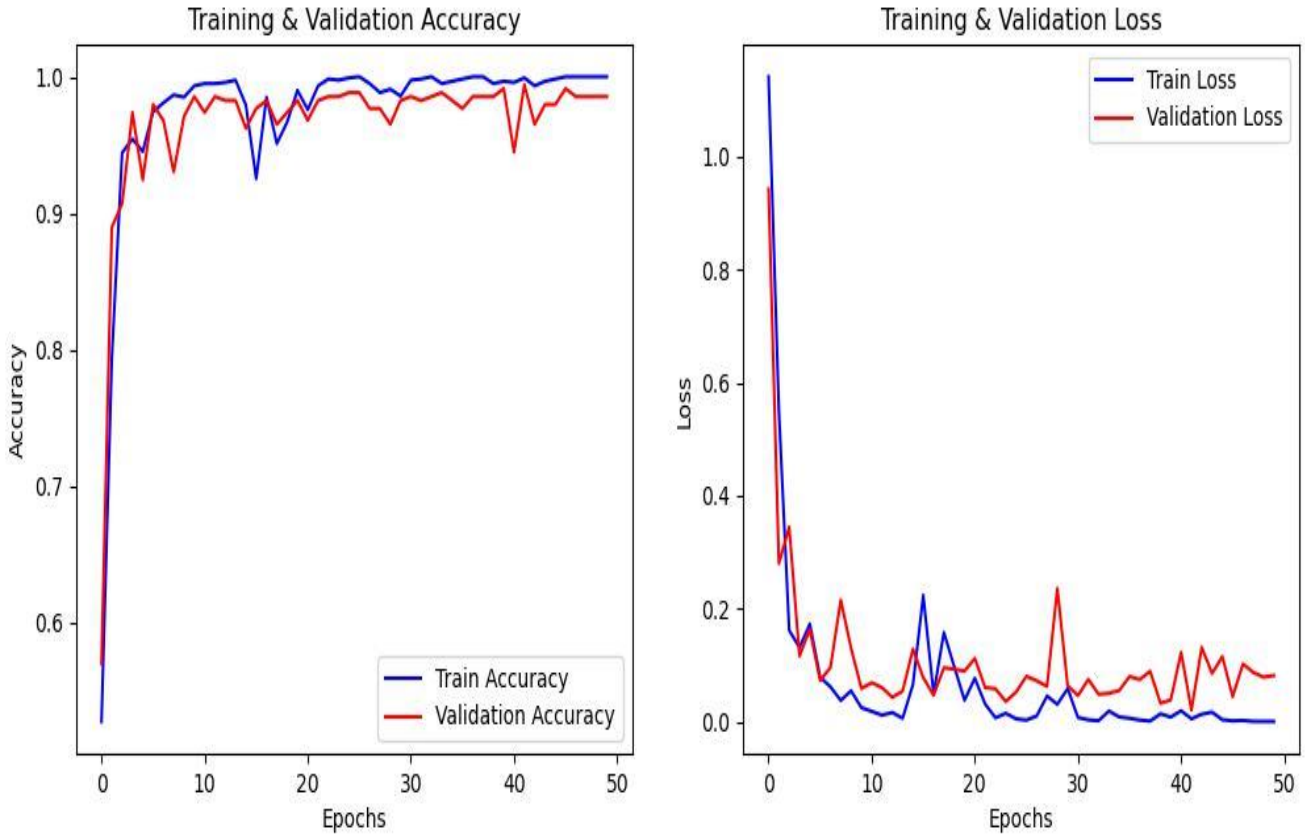


Figure 30: VGG16 accuracy and loss

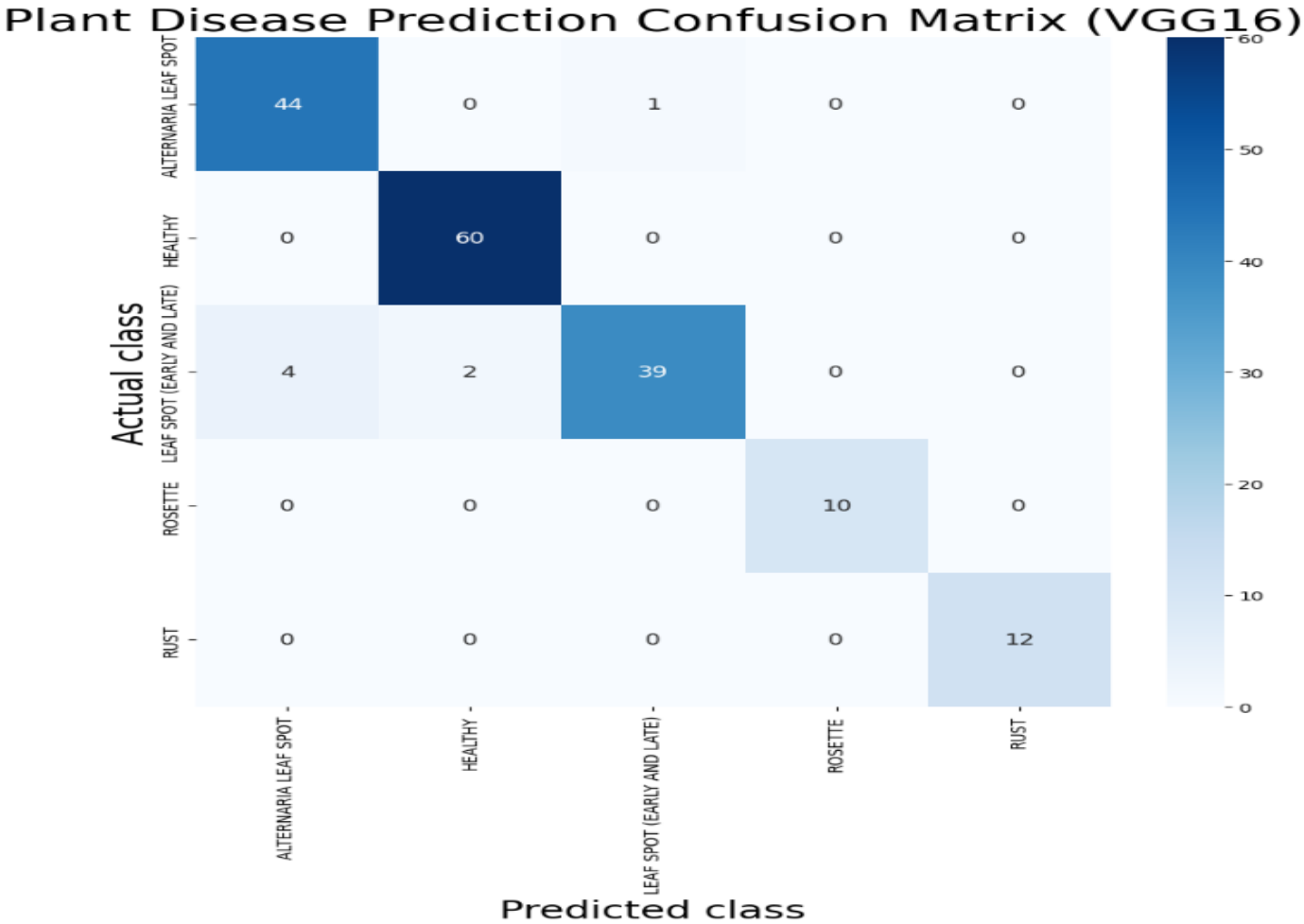
3.1.4. Confusion matrix of VGG16

The confusion matrix (Figure 27) gives a detailed overview of how well the suggested plant disease classifier is working on a wide variety of illnesses: Alternaria Leaf Spot, Healthy, Leaf Spot (Early and Late), Rosette, Rust. The evaluation is based on the usual classification metrics, True Positives (TP), False Positives (FP), False Negatives (FN), and, similarly, True Negatives (TN) to measure how diagnosable the model is and to understand the model's capability to separate different classes.

The Healthy Class predicts 57 healthy plants, with 3 false negatives and 11 false positives. The model is quite good at spotting healthy leaves, but sometimes it confuses early signs of disease for healthy tissue. Overall, it performs well, though some early disease symptoms can look similar to healthy parts. For the Alternaria Leaf Spot, it correctly identified 40 cases but missed

5, mostly confusing them with other leaf spot diseases. It rarely misclassified non-affected leaves, showing it's quite reliable. The visual cues for Alternaria seem well captured by the model, although it still has a bit of trouble distinguishing it from similar diseases like Leaf Spot.

When it comes to Leaf Spot, the model correctly identified 34 cases but missed 11, with 8 of those being incorrectly labeled as Healthy. It struggles to tell the difference between early-stage Leaf Spot and healthy leaves, especially when the symptoms are barely visible. The overall performance is okay but could use improvement in sensitivity and feature extraction to better catch faint or tiny lesions. For Rosette, it got 7 cases right but missed 3, with some false positives. It seems to have a hard time accurately spotting Rosette, likely because the symptoms look a lot like other diseases or vary quite a bit. Its accuracy isn't great, and it would benefit from more varied training data or some fine-tuning. When detecting Rust, the model identified 8 true cases but missed 4, and it also confused some cases with Rosette and Leaf Spot, leading to false positives. It can detect Rust reasonably well, but because some symptoms overlap, it



sometimes gets confused. Overall, it's doing okay, but improving the features or balancing the classes could help make it more reliable.

Figure 31: Matrix of VGG16

3.1.5. VGG19 model training

The third model is the CNN model based on VGG19. A batch size of 32 and an input image size of 224,224 are used to train the model. Utilizing the Adam optimizer, we optimized the VGG19 model with a softmax activation function and softmax cross-entropy loss. The model was trained for 50 epochs and attained an accuracy of 96% with a minimum loss.

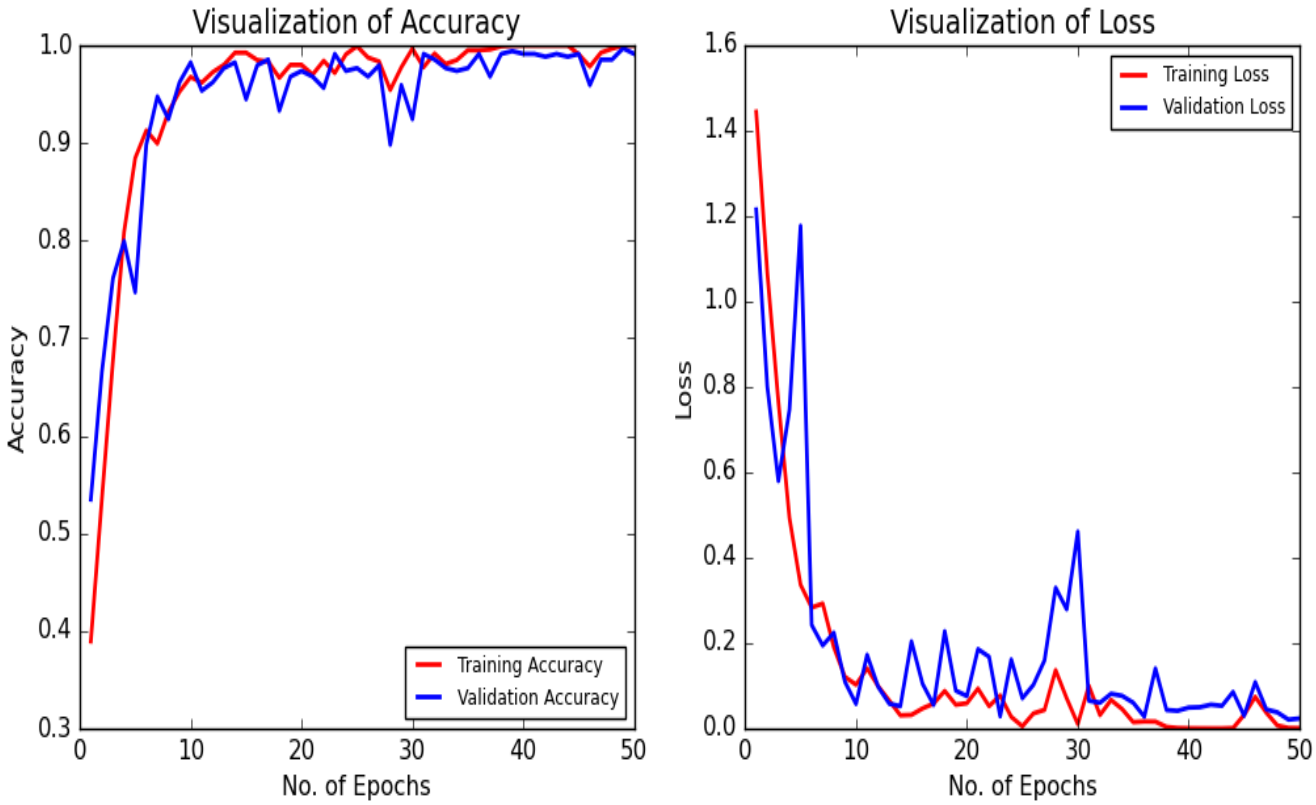


Figure 32: VGG19 accuracy and loss

3.1.6. Confusion matrix of VGG19

The confusion matrix (Figure 29) offers a clear look at how well the VGG19-based plant disease detection model performs across five different categories: Alternaria Leaf Spot, Healthy, Leaf Spot (both Early and Late), Rosette, and Rust. This matrix allows for a thorough

assessment using standard classification metrics, True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN), helping us grasp the model's precision and recall for each category.

For Alternaria Leaf Spot, the model is super good here. It has quite a good amount of True Positives (TP = 42). Only 3 False Negatives (FN) are there: 1 was labeled as Healthy, although it was not, and 2, which were marked as Leaf Spot (both, Early and Late). There is only a very small number of False Positives (FP), which shows that this model has a great sensitivity and specificity in finding this disease. In the Healthy class, we have an awesome performance here. The highest amount of True Positives (TP = 59) and only 2 False Negatives (FN), 1, which was labeled as Leaf Spot, and 1, which was labeled as Alternaria Leaf Spot. On top of that only 1 False Positive (FP), which shows that the model is very good in recognizing healthy plants and does a good job in not being easily duped into believing something that isn't a healthy plant.

Leaf Spot (Early and Late) shows a moderate performance with a true positive (TP) rate of 40, but it also has 5 false negatives (FN) 3 samples were misidentified as Alternaria Leaf Spot and 2 as Healthy. This confusion with similar leaf diseases underscores the challenge of telling apart overlapping symptoms, especially in the early stages.

For Rosette, the model records a TP of 8, but it has 2 FNs (both misclassified as Healthy) and 2 false positives (FPs), where Healthy samples were incorrectly labeled as Rosette. This suggests a lower accuracy, likely due to the similarity of symptoms with other diseases or perhaps a lack of representation in the training data.

On the other hand, the Rust class shows perfect classification, with a TP of 12 and no FNs or FPs. The model effectively identifies Rust symptoms, indicating a strong ability to distinguish features for this particular class.

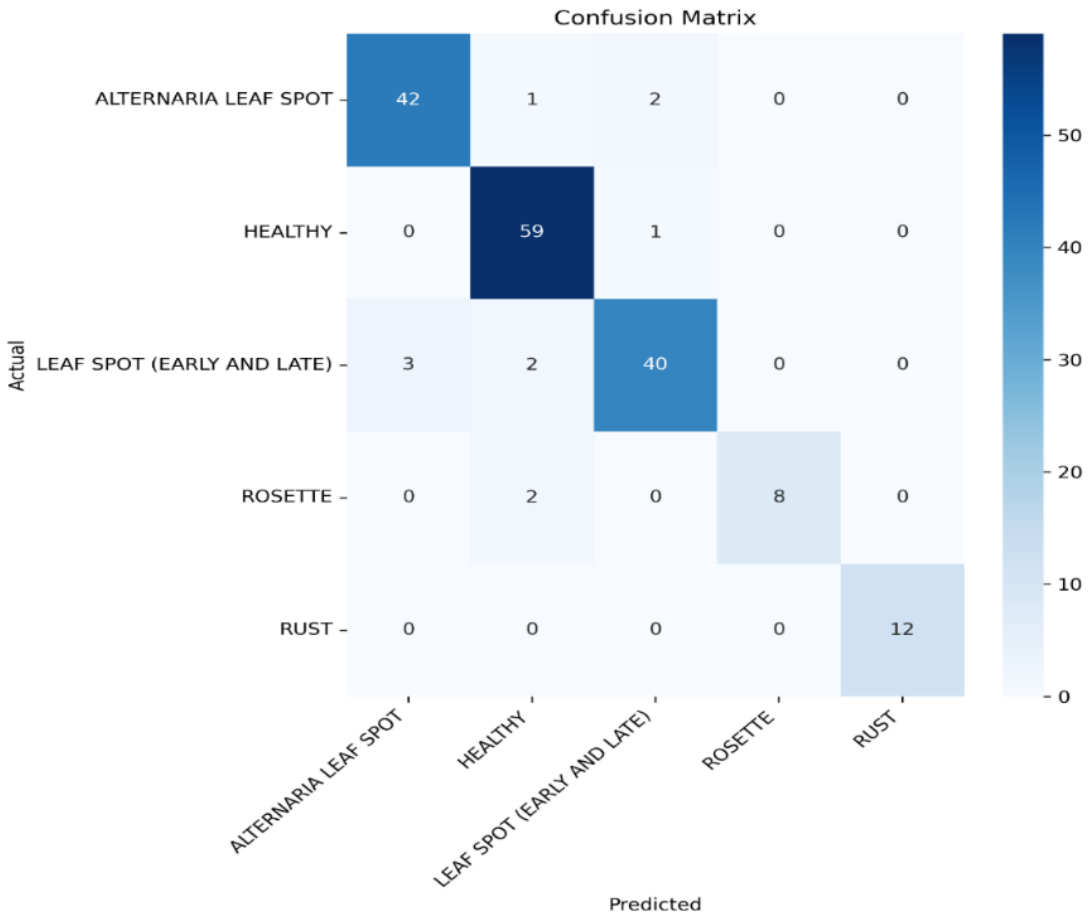


Figure 33: VGG19 matrix

3.2. Discussion

Among the three deep learning architectures assessed, VGG19 stood out as the most precise model, attaining exceptional results regarding accuracy, recall, and F1-score, highlighting its strength in differentiating subtle visual symptoms across the disease categories. Because of its deeper analysis, consistent design, and utilization of pre-trained ImageNet weights, VGG19 performed better than the other models. Thanks to these aspects, it was able to extract more abstract and detailed information from the plant images, which increased the classification accuracy. The performance of each model in training and validation was monitored, as shown by the graphs depicting loss and accuracy. The custom CNN model had the lowest accuracy among the three, while VGG19 and VGG16 achiev21ed the highest accuracy. These findings emphasize the enhanced effectiveness of deeper pre-trained models such as VGG19 for the classification task, presumably because they can extract more complex features from the input images.

VGG19 achieved the highest classification accuracy among the tested models due to several key factors, such as being a more complex model than VGG16 and the custom CNN. It consists of 19 layers (16 convolutional + 3 fully connected), enabling it to acquire more complex and abstract features from the input images. This depth allows VGG19 to detect slight variations in texture, shape, and color patterns essential for tasks such as identifying plant diseases, where visual symptoms may closely resemble one another.

VGG19 (similar to VGG16) is usually initialized using pre-trained weights obtained from ImageNet, a vast and varied dataset. These weights offer a solid foundation for feature extraction, particularly when training data are scarce. Adjusting a previously trained model results in quicker convergence and improved generalization than starting from the beginning.

VGG19 employs a highly uniform architecture featuring small 3×3 convolution filters consistently across the network. This enables it to maintain spatial details efficiently while progressively enhancing feature complexity. Performance on image classification tasks is improved and training is stabilized thanks to this consistency. The deeper layers of VGG19 acquire more abstract, higher-level features, enhancing those from the lower layers. This hierarchical model is especially advantageous for differentiating classes that possess comparable low-level characteristics (such as leaf discoloration or spots).

The application incorporates this model to provide accurate, real-time forecasts directly to users via a user-friendly interface. This guarantees that farmers get precise and practical suggestions with minimal delay, enabling prompt and efficient reactions to disease outbreaks.

Many research efforts have investigated the application of artificial intelligence and deep learning for identifying crop diseases. For instance, Ng et al., 2021 created a mobile app designed for detecting grape leaf diseases through a deep learning object detection method. Similarly, the research by Shrimali, 2021. presented PlantifyAI, a mobile app aimed at diagnosing crop diseases and offering treatment methods. Similarly, suggestions highlighting the importance of deep learning in agriculture. In research conducted by Pan et al., 2019 a system for diagnosing citrus diseases was created, reaching high accuracy in identifying different citrus diseases using a deep learning approach. A study by Dwi Ratna Sulistyaningrum, Alima Rasyida, and Budi Setiyono proposes an approach for identifying rice plant diseases using Support Vector Machine (SVM). A study by Petrellis introduces a method for diagnosing plant diseases through mobile phone applications that operate without needing a connection to a remote server, making it ideal for use in the field Petrellis, 2019.

The results of the current study are consistent with previous research in emphasizing the effectiveness of convolutional architectures in diagnosing plant diseases through images. Nevertheless, this research sets itself apart by utilizing and contrasting three unique architectures, namely a lightweight custom CNN, VGG16, and VGG19, particularly focused on groundnut diseases, which are often underrepresented in numerous worldwide datasets. The impressive accuracy obtained with VGG19 confirms its appropriateness for these applications and adds a new facet to the existing literature centered on disease detection specific to crops.

Plantix, a prominent agricultural application created through a partnership between ICRISAT and the German agri-tech firm PEAT GmbH, is well known for utilizing machine learning to identify plant diseases and pests. The app encompasses more than 30 crops and 120 diseases, utilizing an extensive dataset of over 500,000 images (Samal et al., 2023). It offers support for multiple languages, geo-tagging features, and weather integration, and has attained a user base of over 12 million worldwide. It has taken a leading role in integrating AI into the monitoring of plant health. Nonetheless, the present research uncovers various shortcomings of the Plantix app that the newly created application aims to tackle: Plantix usually offers three possible diagnoses or suggestions for one image, which can create confusion and uncertainty for users, especially in scenarios where farmers must act quickly and do not have access to expert advice. Conversely, the application created in this research provides a single, accurate prediction with great confidence, thereby simplifying decision-making and diminishing cognitive burden for users.

Although Plantix functions with a general-purpose model trained on a wide range of crops, the model utilized in this research is specifically designed for detecting groundnut diseases. This specialization leads to improved predictive accuracy and dependability for groundnut farmers, who are frequently overlooked in generalist platforms. Plantix indicates an overall accuracy of about 85%; the VGG19-based model utilized in this research attained 96% accuracy on the particular dataset, implying that specialized models can surpass generalized ones in crop-specific scenarios.

The findings of this research highlight the practical promise of incorporating specialized deep learning models into mobile apps for application in smallholder farming. The app fosters user confidence and quicker adoption by delivering highly precise and straightforward outcomes. Additionally, the capability to retrain or adapt the application to different crops or diseases offers a level of flexibility that is greatly valued by agricultural extension services. In reality,

these tools can enhance the efforts of agronomists and extension workers, particularly in areas with restricted access to experts. They can additionally function as a quick monitoring tool during outbreak seasons, aiding in reducing crop losses and boosting food security.

Although the created app showed impressive performance, some limitations need to be recognized. The dataset, while representative, was restricted to four disease and healthy categories in groundnuts. Increasing the training data to incorporate greater variability, environmental conditions, and stages of disease would enhance the model's resilience. Moreover, the app at present offers feedback through text; upcoming versions might incorporate voice explanations and support for multiple languages to enhance accessibility.

Future efforts should prioritize practical field testing, analysis of user feedback, and the integration with various agricultural databases and advisory systems. Investigating lightweight architectures for deployment on low-end smartphones and edge computing devices would improve usability in rural regions with poor connectivity.

Conclusion and perspectives

The study confirms that deep learning algorithms, specifically the VGG19 model, are highly effective in accurately detecting diseases in groundnut leaves. It addresses important questions about the use of deep learning in plant disease diagnosis, comparing it to traditional methods and exploring its potential for detecting multiple diseases across various plant species. The findings support the initial hypothesis that models like VGG19 can achieve a high level of accuracy in identifying groundnut diseases. Importantly, the study shows that these algorithms can detect early symptoms of diseases such as anthracnose and leaf spot more efficiently than traditional visual inspection methods. Minimizing crop losses and ensuring timely action depend on this early diagnosis. Additionally, the app was able to offer real-time diagnoses, confirming the theory that farmers may be empowered to make knowledgeable decisions about disease management by using digital tools. Given enough training data, the model's performance also points to its scalability and potential for adaptation for the diagnosis of numerous diseases in various plant species.

The study effectively achieved its intended goals. Deep learning models, including VGG19, VGG16, and custom-designed CNN architectures, were successfully trained and assessed, demonstrating strong accuracy in classifying groundnut leaf diseases. The application's diagnostic capabilities were thoroughly tested and confirmed using a diverse set of groundnut disease images. An intuitive and accessible user interface was created, allowing farmers to easily upload plant images, obtain immediate diagnostic results, and receive tailored recommendations for disease management.

This study shows how mobile technology and deep learning may be used to effectively address important issues in agriculture. It offers a novel solution for smallholder farmers by concentrating on groundnut disease diagnosis by integrating CNN-based models into a user-centric application. VGG19 fared better than the other models under evaluation, demonstrating the advantages of transfer learning and deeper architectures. Through personalized recommendations, the app improves decision-making in addition to making disease detection easier. This specific method works better and is crop-specific than the current options, like Plantix. The app's usefulness for farmers with spotty internet connectivity is increased by its offline capabilities, user-friendly UI, and integrated chatbot. By facilitating prompt, well-informed interventions that maximize resource utilization and reduce production losses, it supports digital agriculture. Its accessibility and influence could be further increased in the

future by adding voice interaction, multilingual assistance, and support for additional crops, using local data for more precision.

All things considered, this study emphasizes how AI and mobile technology are revolutionizing agriculture and enhancing resistance to climate-related risks.

Bibliography references

Abady, S., Shimelis, H., Janila, P., & Mashilo, J. (2019). Groundnut (Arachis hypogaea L.) improvement in sub-Saharan Africa: A review. *Acta Agriculturae Scandinavica Section B: Soil and Plant Science*, *69*(6), 528–545. https://doi.org/10.1080/09064710.2019.1601252;WGROUP:STRING:PUBLICATION

Akram, N. A., Shafiq, F., & Ashraf, M. (2018). Peanut (Arachis hypogaea L.): A Prospective Legume Crop to Offer Multiple Health Benefits Under Changing Climate. *Comprehensive Reviews in Food Science and Food Safety*, *17*(5), 1325–1338. https://doi.org/10.1111/1541-4337.12383

Alagirisamy, M. (2016). Groundnut. *Breeding Oilseed Crops for Sustainable Production: Opportunities and Constraints*, 89–134. https://doi.org/10.1016/B978-0-12-801309-0.00005-7

Appiah, O., Hackman, K. O., Diallo, B. A. A., Ogunjobi, K. O., Diakalia, S., Valentin, O., Abdoul-Karim, D., & Dabire, G. (2024). PlanteSaine: An artificial intelligence empowered mobile application for pests and disease management for maize, tomato, and onion farmers in Burkina Faso. *Agriculture (Switzerland)*, *14*(8). https://doi.org/10.3390/agriculture14081252

Banda, P. C. (2022). Undergraduates' views and attitudes towards the use of computer-based tests for examination at the University of Lagos, Nigeria. *International Journal of Advanced Research in Social Sciences, Environmental Studies & Technology*, *7*(1), 58–61. https://doi.org/10.48028/iiprds/ijarssest.v7.i1.05

Barkley, N. A., Upadhyaya, H. D., Liao, B., & Holbrook Corley, C. (2016). Global Resources of Genetic Diversity in Peanut. *Peanuts: Genetics, Processing, and Utilization*, 67–109. https://doi.org/10.1016/B978-1-63067-038-2.00003-4

Basuchaudhuri, P. (2022). Physiology of the Peanut Plant. *Physiology of the Peanut Plant*. https://doi.org/10.1201/9781003262220

Bertioli, D. J., Cannon, S. B., Froenicke, L., Huang, G., Farmer, A. D., Cannon, E. K. S., Liu, X., Gao, D., Clevenger, J., Dash, S., Ren, L., Moretzsohn, M. C., Shirasawa, K., Huang, W., Vidigal, B., Abernathy, B., Chu, Y., Niederhuth, C. E., Umale, P., … Ozias-Akins, P. (2016). The genome sequences of Arachis duranensis and Arachis ipaensis , the diploid ancestors of cultivated peanut. *Nature Genetics*, *48*(4), 438–446. https://doi.org/10.1038/ng.3517

Boubacar, S., Desmae, H., Miningou, A., Nebie, B., Eleblu, J., Kwadwo, O., Traore, A., Zagre, B., & Tabo, R. (2020). *Participatory analysis of groundnut (Arachis hypogaea L) cropping system and production constraints in Burkina Faso*. https://doi.org/10.21203/RS.3.RS-57960/V1

Buddhadev, S., Das, A., Dhal, K. G., Saheb, S. B., Khurma, R. A., & Castillo, P. A. (2024). A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases. *Data in Brief*, *55*, 110763. https://doi.org/10.1016/J.DIB.2024.110763

Chandra, B. R. K. et al., Balakrishnan, N., & Rao, C. S. (2023). Deep fake image classification using VGG-19 model. *Ingenierie Des Systemes d'Information*, *28*(2), 509–515. https://doi.org/10.18280/isi.280228

Chikezie, F. M., Opara, K. N., & Ubulom, P. M. E. (2024). Impacts of Changing Climate on Arthropod Vectors and Diseases Transmission. *Nigerian Journal of Entomology*, *40*(1), 179–192. https://doi.org/10.36108/nje/4202/04.0161

Chohan, M., Khan, A., Chohan, R., Katpar, S. H., & Mahar, M. S. (2020). Plant disease detection using deep learning. *International Journal of Recent Technology and Engineering (IJRTE)*, *9*(1), 909–914. https://doi.org/10.35940/ijrte.A2139.059120

Durga, C. (2020). Artificial intelligence in agriculture. *Artificial Intelligence in Agriculture. Research Today*, *2*(7), 578–579. www.bioticainternational.com

Dutta, P., & Kumari, A. (2023). Diseases of Groundnut (Arachis hypogaea L.) and their integrated management. *Diseases of Oil Crops and Their Integrated Management*, 26–44. https://doi.org/10.1201/9781032627960-2/DISEASES-GROUNDNUT-ARACHIS-HYPOGAEA-INTEGRATED-MANAGEMENT-PRANAB-DUTTA-ARTI-KUMARI

Gai, Y., & Wang, H. (2024). Plant Disease: A growing threat to global food security. In *Agronomy* (Vol. 14, Issue 8). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/agronomy14081615

Ghafoor, A. Z., Javed, H. H., Karim, H., Studnicki, M., Ali, I., Yue, H., Xiao, P., Asghar, M. A., Brock, C., & Wu, Y. (2024). Biological Nitrogen Fixation for Sustainable Agriculture Development Under Climate Change–New Insights From a Meta-Analysis. *Journal of Agronomy and Crop Science*, *210*(5). https://doi.org/10.1111/JAC.12754

Gimenez, E., Maria, S., & Francisco, M.-A. (2018). Worldwide research on plant defense against biotic stresses as improvement for sustainable agriculture. *Sustainability 2018, Vol. 10, Page 391*, *10*(2), 391. https://doi.org/10.3390/SU10020391

Guo, S. (2023). Leaf disease detection by convolutional neural network (CNN). In *Highlights in Science, Engineering and Technology TPCEE* (Vol. 2023).

Gurunathan, V., Sathiya Priya, T., Dhanasekar, J., Ishwarya Niranjana, M., & Suganya, S. (2023). Plant leaf diseases detection using KNN classifier. *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, 2157–2162. https://doi.org/10.1109/ICACCS57279.2023.10112901

Huang, R., Li, H., Gao, C., Yu, W., & Zhang, S. (2023). Advances in omics research on peanut response to biotic stresses. *Frontiers in Plant Science*, *14*, 1101994. https://doi.org/10.3389/FPLS.2023.1101994/XML/NLM

Hunjan, M. S., & Lore, J. S. (2020). Climate Change: Impact on Plant Pathogens, Diseases, and Their Management. *Crop Protection Under Changing Climate*, 85–100. https://doi.org/10.1007/978-3-030-46111-9_4

Islam, M. T. (2022). *Plant disease detection using CNN model and image processing plant disease detection using CNN model and image processing*. https://www.researchgate.net/publication/358768246

James, M., Patrick, O., Wills, M., & Thomas, O. (2023). The Groundnut Rosette Disease at a Glance: Basics, Management and the Future. *Journal of Plant Sciences*. https://doi.org/10.11648/J.JPS.20231105.11

Jogin, M., Mohana, Madhulika, M. S., Divya, G. D., Meghana, R. K., & Apoorva, S. (2018). Feature extraction using convolution neural networks (CNN) and deep learning. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication*

Technology, RTEICT 2018 - Proceedings, 2319–2323. https://doi.org/10.1109/RTEICT42901.2018.9012507

Jonathan, F. T., & Mahendranathan, C. (2024). Impact of Climate Change on Plant Diseases. *AGRIEAST: Journal of Agricultural Sciences*, *18*(2), 1–17. https://doi.org/10.4038/agrieast.v18i2.133

Junyan, L., Xu, S., & Li, Y. (2009). Application research of embedded database SQLite. *Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009*, *2*, 539–543. https://doi.org/10.1109/IFITA.2009.408

Kankam, F., Akpatsu, I. B., & Tengey, T. K. (2022). Leaf spot disease of groundnut: A review of existing research on management strategies. In *Cogent Food and Agriculture* (Vol. 8, Issue 1). Informa Healthcare. https://doi.org/10.1080/23311932.2022.2118650

Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC Medical Imaging 2022 22:1*, *22*(1), 1–13. https://doi.org/10.1186/S12880-022-00793-7

Konate, M., Sanou, J., Miningou, A., Okello, D. K., Desmae, H., Janila, P., & Mumm, R. H. (2020). Past, present and future perspectives on groundnut breeding in Burkina Faso. *Agronomy*, *10*(5). https://doi.org/10.3390/AGRONOMY10050704

Kothari, J. D. (2018). *Plant disease identification using Artificial intelligence: Machine learning approach*. https://ssrn.com/abstract=3729753

Kumar, V. (2016). *Diseases of groundnut thirumalaisamy polavakkalipalayam palanisamy Indian Council of agricultural research*. https://www.researchgate.net/publication/305392088

Leal-Bertioli, S. C. M., de Blas, F. J., Carolina Chavarro, M., Simpson, C. E., Valls, J. F. M., Tallury, S. P., Moretzsohn, M. C., Custodio, A. R., Thomas Stalker, H., Seijo, G., & Bertioli, D. J. (2024). Relationships of the wild peanut species, section Arachis: A resource for botanical classification, crop improvement, and germplasm management. *American Journal of Botany*, *111*(6), e16357. https://doi.org/10.1002/AJB2.16357;WEBSITE:WEBSITE:BSAPUBS;WGROUP:STRING:PUBLICATION

Manish, K., & Lalit, A. (2020). Empowering farming community through mobile applications: changing scenarios. *International Journal of Scientific & Technology Research*. www.ijstr.org

Masson-Delmotte, V., Zhai, P., Chen, Y., Goldfarb, L., Gomis, M. I., Matthews, J. B. R., Berger, S., Huang, M., Yelekçi, O., Yu, R., Zhou, B., Lonnoy, E., Maycock, T. K., Waterfield, T., Leitzell, K., & Caud, N. (2021). *Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change Edited by*. www.ipcc.ch

Melesse, M. B., Miriti, P., Muricho, G., Ojiewo, C. O., & Afari-Sefa, V. (2023). Adoption and impact of improved groundnut varieties on household food security in Nigeria. *Journal of Agriculture and Food Research*, *14*, 100817. https://doi.org/10.1016/J.JAFR.2023.100817

Mensah, P. K. et al., Akoto-Adjepong, V., Adu, K., Ayidzoe, M. A., Bediako, E. A., Nyarko-Boateng, O., Boateng, S., Donkor, E. F., Bawah, F. U., Awarayi, N. S., Nimbe, P., Nti, I. K., Abdulai, M., Adjei, R. R., Opoku, M., Abdulai, S., & Amu-Mensah, F. (2023). CCMT: Dataset for crop pest and disease detection. *Data in Brief*, *49*, 109306. https://doi.org/10.1016/J.DIB.2023.109306

Mohamed. (2024). *Evaluation des hybrides de sorgho (Sorghum bicolor) pour la résistance au Striga hermonthica dans la zone soudanienne du Mali*.

Mohaned, M. (2024). *Mobile application for detecting and diagnosing plant diseases using deep learning*. https://doi.org/10.13140/RG.2.2.12956.71041

Munaf, M., & Karan, O. (2023). Deep learning for plant disease detection. *International Journal of Mathematics, Statistics, and Computer Science*, *2*, 75–84. https://doi.org/10.59543/ijmscs.v2i.8343

Nehemia, K., Rotich, L. K., & Riungu, G. K. (2015). Agriculture, climate change and food security. *OALib*, *02*(05), 1–7. https://doi.org/10.4236/oalib.1101472

Ng, H. F., Lin, C. Y., Chuah, J. H., Tan, H. K., & Leung, K. H. (2021). Plant Disease Detection Mobile Application Development using Deep Learning. *Proceedings - International Conference on Computer and Information Sciences: Sustaining Tomorrow with Digital Innovation, ICCOINS 2021*, 34–38. https://doi.org/10.1109/ICCOINS49721.2021.9497190

Nigam, S. N. (2014). *Groundnut at a glance, pp.121*.

Osisanwo, F. Y., J.E.T, A., O, A., J. O, H., O, O., & J, A. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology*, *48*(3), 128–138. https://doi.org/10.14445/22312803/IJCTT-V48P126

Pan, W., Qin, J., Xiang, X., Wu, Y., Tan, Y., & Xiang, L. (2019). A Smart Mobile Diagnosis System for Citrus Diseases Based on Densely Connected Convolutional Networks. *IEEE Access*, *7*, 87534–87542. https://doi.org/10.1109/ACCESS.2019.2924973

Petrellis, N. (2019). Plant disease diagnosis for smart phone applications with extensible set of diseases. *Applied Sciences (Switzerland)*, *9*(9). https://doi.org/10.3390/app9091952

Pokhrel, A. (2021). Role of individual components of disease triangle in disease development: A review. *Review Article in Journal of Plant Pathology & Microbiology*, *12*(9), 573. https://doi.org/10.35248/2157-7471.21.12.573

Pratap, D., Tamuly, G., R, G. N., Anbarasan, S., Pandey, A. K., Singh, A., P, P., Debnath, A., Asmatullah, & Iberaheem, M. (2024). Climate Change and Global Agriculture: Addressing Challenges and Adaptation Strategies. *Journal of Experimental Agriculture International*, *46*(6), 799–806. https://doi.org/10.9734/JEAI/2024/V46I62533

Qureshi, I. (2024). Integrating Few-Shot Learning and Multimodal Image Enhancement in GNut: A Novel Approach to Groundnut Leaf Disease Detection. *Computers 2024, Vol. 13, Page 306*, *13*(12), 306. https://doi.org/10.3390/COMPUTERS13120306

Sahoo, L., Mohapatra, D., Raghuvanshi, H. R., Kumar, S., Kaur, R., Anshika, ., Sapna, ., Chawla, R., & Afreen, N. (2024). Transforming Agriculture through Artificial Intelligence: Advancements in Plant Disease Detection, Applications, and Challenges. *Journal of Advances in Biology & Biotechnology*, *27*(5), 381–388. https://doi.org/10.9734/jabb/2024/v27i5796

Saif, M. F. M. A., & Nureize, A. (2023). *Plant disease detection application using deep learning(PLANTSCARE)*. https://doi.org/10.30880/aitcs.2023.04.02.062

Samal, I., Bhoi, T. K., & Kumar Mahanta, D. (2023). *Plantix app: A success story of artificial intelligence in plant protection*. https://www.researchgate.net/publication/370650382

Sasmal, B., Das, A., Dhal, K. G., Saheb, B., Abu Khurma, R., & Castillo-Valdivieso, P. A. (2024). *A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases*. *2*. https://doi.org/10.17632/X6X5JKK873.2

Shanmugam, S., & Dharmar, S. (2024). Implementation of a non-linear SVM classification for seizure EEG signal analysis on FPGA. *Engineering Applications of Artificial Intelligence*, *131*, 107826. https://doi.org/10.1016/J.ENGAPPAI.2023.107826

Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. In *Frontiers in Plant Science* (Vol. 14). Frontiers Media SA. https://doi.org/10.3389/fpls.2023.1158933

Shrimali, S. (2021). PlantifyAI: A novel convolutional neural network based mobile application for efficient crop disease detection and treatment. *Procedia Computer Science*, *191*, 469–474. https://doi.org/10.1016/j.procs.2021.07.059

Sivaganesan, D. (2023). Noncommercial International (CC BY-NC 4.0) License Deep Learning Approaches for Disease Detection in Groundnut Crops using CNN Models. *Journal of Soft Computing Paradigm*, *5*(4), 404–416. https://doi.org/10.36548/jscp.2023.4.006

Sjaak, W., LanGe, Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming – A review. *Agricultural Systems*, *153*, 69–80. https://doi.org/10.1016/J.AGSY.2017.01.023

Sonali, B., Bamane, A., Bhagat, S., & Chande, A. (2024). A deep learning based framework for accurate prediction of disease of crops and nutrients in smart agriculture. In *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* (Vol. 12). www.ijraset.com

Sulistyaningrum, D. R., Hasan, H., Shafri, H. Z. M., Habshi, M., Zulvia, F. E., Kuo, R. J., Rasyida, A., & Setiyono, B. (2020). Rice disease classification based on leaf image using multilevel Support Vector Machine (SVM). *Journal of Physics: Conference Series*, *1490*(1), 012053. https://doi.org/10.1088/1742-6596/1490/1/012053

Suza, W. P., & Lamkey Eds, K. (2025). *Crop improvement*. https://LibreTexts.org

Syed, F., Arif, S., Ahmed, I., & Khalid, N. (2021). Groundnut (Peanut) (Arachis hypogaea). *Oilseeds: Health Attributes and Food Applications*, 93–122. https://doi.org/10.1007/978-981-15-4194-0_4

Thyagaraj, R., Satheesha, T. Y., & Bhairannawar, S. (2023). Plant leaf disease classification using modified SVM with post processing techniques. *International Conference on Applied Intelligence and Sustainable Computing, ICAISC 2023*. https://doi.org/10.1109/ICAISC58445.2023.10201001

Tulshan, Amrita. S., & Raul, N. (2019). Plant leaf disease detection using machine learning. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*. https://doi.org/10.1109/ICCCNT45670.2019.8944556

Türkoğlu, M., & Davut, H. (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(3), 1636–1651. https://doi.org/10.3906/ELK-1809-181

Zhang, H., Mascher, M., Abbo, S., & Jayakodi, M. (2022). Advancing grain legumes domestication and evolution studies with genomics. *Plant and Cell Physiology*, *63*(11), 1540–1553. https://doi.org/10.1093/PCP/PCAC062

Appendix A: Code VGG19

```
# -------------------- Import Libraries --------------------
from tensorflow.keras.applications import VGG19
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import Flatten, Dense, Input, GlobalAveragePooling2D, Rescaling
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import Precision, Recall
from sklearn.metrics import confusion_matrix
import tensorflow as tf
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from glob import glob
# -------------------- Data Preprocessing --------------------
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
valid_datagen = ImageDataGenerator(rescale=1./255)

training_set = train_datagen.flow_from_directory(
    "/home/aisssata/Documents/ICRISAT/Groundnut-Leaf-Diseases/Image_Plant/train",
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)
```

```
validation_set = valid_datagen.flow_from_directory(
    "/home/aisssata/Documents/ICRISAT/Groundnut-Leaf-Diseases/Image_Plant/val",
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
# ---------- Model Construction ----------
vgg = VGG19(input_shape=(224, 224, 3), weights='imagenet', include_top=False)
x = Flatten()(vgg.output)
folders = glob("/home/aisssata/Documents/ICRISAT/Groundnut-Leaf-Diseases/Image_Plant/train/*")
prediction = Dense(len(folders), activation='softmax')(x)
model = Model(inputs=vgg.input, outputs=prediction)

# ---------- Model Compilation ----------
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(learning_rate=0.0001),
    metrics=['accuracy']
)x = Flatten()(vgg.output)
folders = glob("/home/aisssata/Documents/ICRISAT/Groundnut-Leaf-Diseases/Image_Plant/train/*")
prediction = Dense(len(folders), activation='softmax')(x)
model = Model(inputs=vgg.input, outputs=prediction)

# ---------- Training the Model ----------
r = model.fit(
    training_set,
    validation_data=validation_set,
    epochs=50,
    steps_per_epoch=len(training_set),
    validation_steps=len(validation_set)
)    metrics=['accuracy']
```

```
# ---------- Plot Accuracy and Loss ----------
plt.subplot(1, 2, 1)
plt.plot(r.history['accuracy'], color='red', label='Training Accuracy')
plt.plot(r.history['val_accuracy'], color='blue', label='Validation Accuracy')
plt.xlabel("Number of Epochs")
plt.ylabel("Accuracy")
plt.title("Model Accuracy")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(r.history['loss'], color='red', label='Training Loss')
plt.plot(r.history['val_loss'], color='blue', label='Validation Loss')
plt.xlabel("Number of Epochs")
plt.ylabel("Loss")
plt.title("Model Loss")
plt.legend()

plt.tight_layout()
plt.savefig("VGG19.jpeg")
plt.show()

# ---------- Load Test Dataset ----------
test_set = tf.keras.utils.image_dataset_from_directory(
    '/home/aisssata/Documents/ICRISAT/Groundnut-Leaf-Diseases/Image_Plant/test',
    labels="inferred",
    label_mode="categorical",
    color_mode="rgb",
    batch_size=32,
    image_size=(224, 224),
    shuffle=False
```

```
# -------------------- Evaluate Model --------------------
test_loss, test_acc = model.evaluate(test_set)
print(f'Test Loss: {test_loss:.4f}')
print(f'Test Accuracy: {test_acc:.4f}')

# -------------------- Confusion Matrix --------------------
class_names = ["Alternaria Leaf Spot", "Healthy", "Leaf Spot (Early and Late)", "Rosette", "Rust"]
cm = np.array([
    [42, 1, 2, 0, 0],
    [ 0, 59, 1, 0, 0],
    [ 3, 2, 40, 0, 0],
    [ 0, 2, 0, 8, 0],
    [ 0, 0, 0, 0, 12]
])

plt.figure(figsize=(8, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names)
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
```

Appendix B: Code source for the application

https://github.com/doussou1999/CropDoctorapp.git

TABLE OF CONTENTS

DEDICATION v
ACKNOWLEDGEMENTS vi
DECLARATION viii
ABSTRAST xi
RESUME xii
Introduction 1
Chapter 1: Literature review 6
1.1. Overview of groundnut 6
1.1.1. Crop biology 6
1.1.2. Geographical distribution of the crop 7
1.1.3. Crop inflorescence 7
1.1.4. Importance of crop 8
1.1.5. Usages of crop 8
1.1.6. Disease constraints of crop 9
1.2. Description of groundnut leaf diseases and symptoms 10
1.2.1. Alternaria leaf spot 10
1.2.2. Healthy leaf ..... **Error! Bookmark not defined.**
1.2.3. Leaf spot (early and late) 11
1.2.4. Rosette disease 13
1.2.5. Rust disease 14
1.3. Impact of climate change on groundnut diseases spread and severity 16
1.4. Machine learning (ML) techniques 18
1.4.1. K-nearest neighbors (KNN) 18
1.4.2. Support vector machine (SVM) 19
1.5. Deep learning technique 20
1.5.1. Convolutional neural networks (CNN) 21
1.5.2. Transfer learning models 23
1.5.3. VGG16 model 23
1.5.4. VGG19 model 24
1.6. Plant disease detection software 25
1.7. Software requirements 27
1.7.1. Functional requirements 27
1.7.2. Non-functional requirements 28
1.7.3. User story 28

1.7.4. Design 29
1.7.5. Database design 30
Chapter 2 Materials and methods 32
2.1. Materials 32
2.1.1. Study area 32
2.1.2. Python programming language 32
2.2. Methods 33
2.2.1. Data collection 33
2.2.2. Image processing and analysis 35
2.2.3. Different models used for training 36
2.2.4. Software design architecture 39
2.2.5. Application development 40
Chapter 3 Results and discussion 45
3.1. Results 45
3.1.1. CNN model training 45
3.1.2. Confusion matrix of CNN 46
3.1.3. VGG16 model training 47
3.1.4. Confusion matrix of VGG16 48
3.1.5. VGG19 model training 50
3.1.6. Confusion matrix of VGG19 50
3.2. Discussion 52
Conclusion and perspectives 56
Bibliography references 58
Appendix A: Code VGG19 I
Appendix B: Code source for the application V