DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

FACULTY OF MATHEMATICS AND INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE
N° :……………23……………………..

DOMAINE: Mathematics and Informatics
FILIERE: Informatics
OPTION: INFORMATION SYSTEMS AND SOFTWARE ENGINEERING
…………………………….

A Dissertation in Fulfillment
For the Requirements of the Degree of Master
In Computer Science
By:

BLIZAK Widad
BOUDJOUDI Ferial Amina

Entitled

Plant Disease Classification using Fine- grained Techniques

Presented publicly to the jury:

BOURAHLA Mustapha	University of M'sila	President
DEBBI Hichem	University of M'sila	Superviser
BOUZAROURA Ahlem	University of M'sila	Examiner

Academic Year: 2023/2024

DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

FACULTY OF MATHEMATICS AND INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE
N° :…………23………………………..

DOMAINE: Mathematics and Informatics

FILIERE: Informatics
OPTION: INFORMATION SYSTEMS AND SOFTWARE ENGINEERING
…………………………….

A Dissertation in Fulfillment
For the Requirements of the Degree of Master
In Computer Science
By:

BLIZAK Widad
BOUDJOUDI Ferial Amina

Entitled

Plant Disease Classification using Fine- grained Techniques

Presented publicly to the jury:

BOURAHLA Mustapha	University of M'sila	President
DEBBI Hichem	University of M'sila	Superviser
BOUZAROURA Ahlem	University of M'sila	Examiner

Academic Year: 2023/2024

DEDICATIONS

“We would like to dedicate this dissertation to our beloved families and dear friends, who have been an unwavering source of support and motivation throughout our studies. Their steadfast presence and continuous encouragement have had a profound impact on our success.

We are grateful for their enduring love and sacrifices, and we thank them for all the support and encouragement they have provided us. This dissertation would not have been possible without their presence and boundless support.

We dedicate this work to them with all our love and gratitude.”

Acknowledgements

“First of all, we say thanks to God for agreeing us to do this work and providing us with health and wellness so that we can endure the problems and difficulties for writing this memo and continue until the end.
We extend our gratitude to Supervisor Hicham Debbi for his guidance and assistance and provide us with his expertise and knowledge throughout this work.
Finally, we would like to express our deep gratitude to our family, friends, and all those who have contributed to us with the support and encouragement that has always been present with us. “

Contents

List of Figures

List of Tables:

General Introduction

Agriculture is one of the most vital sectors upon which global food security depends. As the world's population continues to grow, the need to improve agricultural productivity and ensure crop quality increases. However, agriculture faces significant challenges due to plant diseases, which can lead to substantial crop losses and negatively impact the global economy. These challenges require innovative and effective solutions for early diagnosis and efficient treatment of plant diseases.

In recent years, deep learning and computer vision technologies have seen tremendous advancements, creating new opportunities to enhance the processes of classifying and diagnosing plant diseases. These technologies rely on deep neural networks capable of analyzing images and extracting complex patterns with high accuracy, thereby enhancing the ability of farmers and agricultural experts to quickly and effectively identify and treat diseases.

This thesis focuses on developing a model for classifying plant leaf diseases using deep learning and computer vision techniques. We use a three different datasets containing images of leaves affected by common diseases (grape, watermelon, and tomato) —fruits chosen because they are among the most consumed and produced in Algeria. We train a model based on convolutional neural networks (CNNs) to analyze and accurately classify these images by capturing all subtle fine-grained features. Our goal through this work is to improve diagnostic accuracy and reduce the time and costs associated with disease detection, thereby contributing to enhanced agricultural productivity and sustainability.

The thesis includes the following chapters:

Chapter 01 : Food Security and Plant Leaf Diseases.

Chapter 02 :Deep Learning .

Chapter 03 : Plant leaf diseases for computer vision.

Chapter 04 : Implementation and Analysis of Results.

Chapter 1: food security and plant leaf diseases

1. Introduction

Agriculture stands as the cornerstone of ensuring food security, yet plant diseases pose a significant threat to this goal. These diseases impact both the quantity and quality of crops, leading to substantial economic losses. In this chapter, we discuss the importance of food security and shed light on plant leaf diseases and their negative impact. We specifically focus on leaf diseases of watermelon, tomato, and grape.

2. Food Security

FS is commonly understood as the state when all people at all times have physical, social and economic access to sufficient, nutritious and safe food to meet their dietary needs and food preferences for an active and healthy life [9].

2.1. The components of FS

Availability: Sufficient quantities of food are consistently available on a national or global scale through production, imports, or food aid.

Access: Individuals and households have physical, economic, and social access to nutritious and culturally appropriate food, including the ability to afford and obtain food without discrimination or barriers.

Utilization: Food is utilized effectively by individuals, with access to safe and nutritious food, clean water, and adequate sanitation, as well as knowledge of proper nutrition and food preparation practices.

Stability: Food security is maintained over time, with access to food not being significantly disrupted by factors such as price volatility, natural disasters, conflict, or economic instability.

These components collectively ensure that people have access to sufficient, safe, and nutritious food to meet their dietary needs and lead healthy, active lives.

Fig 1 Four components of Food Security .[50]

2.2. Factors Affecting Food Security

Despite the importance of food security, many factors can lead to its deterioration. While most of these factors are not new, they worsen the existing global food system imbalance. This section will discuss the major factors that impact food security:

Climate Change: Food security has a strong link with the environment. Climate change is one of our most pressing environmental issues, and farmers and food producers worldwide are already feeling its effects. Climate change leads to droughts, floods, and other extreme weather conditions that damage crops and livestock. This reduces the availability of food and the quality and nutrition of the available food.

Population Growth: As the world's population grows, the demand for food also increases. This puts a strain on already scarce resources, making it difficult to meet the needs of everyone. In addition, population growth also leads to urbanization, which can result in the loss of agricultural land and the displacement of farmers.

Economic Instability: The global economy has a big impact on food security. When the economy is doing well, there is usually an increase in food production and availability. However, during economic downturns, the opposite is true. Poor economic conditions can lead to job loss, reduced incomes, and increased food prices. This makes it difficult for people to access the food they need, leading to hunger and malnutrition.

Energy Security: Food production, transportation, and distribution require a lot of energy. This energy comes from oil, gas, and other fossil fuels. As these resources become scarce, the cost of food production increases. This not only makes food more expensive for consumers but also reduces its availability.

COVID-19: The COVID-19 pandemic has had a major impact on food security. The virus has led to production disruptions, supply chain problems, and economic instability. This has made it difficult for people to access the food they need, leading to hunger and malnutrition. In addition, the pandemic has also led to an increase in food prices.

Conflicts: From the Syrian Civil War to the conflict in Sudan, wars and conflicts are major causes of food insecurity. They often lead to the displacement of people, damage to infrastructure, and loss of agricultural land. This makes it difficult for people to produce or access the food they need.

Water Security: Just like energy, water is essential for the production of food. When water resources are scarce or contaminated, it can lead to crop loss and reduced food production. This not only makes food more expensive but also decreases its availability. If this continues, it could lead to a global water crisis.

2.3. The Importance of food security

Food security **is essential** because it guarantees that all individuals have continuous access to sufficient, safe, and nutritious food to maintain a healthy and productive life. Its importance lies in several interconnected aspects: [9]

- **Economic Stability**: Ensuring food security helps prevent poverty and fosters economic growth by enabling people to earn wages and contribute productively to society .
- **Health and Wellbeing**: Adequate nutrition promotes better overall health, particularly among children, whose cognitive and physical development relies heavily on proper nutrition .
- **Women's Empowerment:** Women's involvement in farming and earning power contributes directly to improved food security, leading to stronger communities and reduced vulnerabilities .
- **Environmental Resilience:** Addressing food security challenges involves managing climate change risks and protecting natural resources, ensuring sustainability for future generations .
- **Conflict Reduction:** Food insecurity exacerbates tensions between nations and communities, while achieving food security can promote peace and stability .
- **National Defense**: Food security enhances national defense capabilities by maintaining a healthy and capable workforce .
- **Global Cooperation:** Collaboration among governments, international agencies, and private entities is vital to address food security challenges and achieve global goals such as zero hunger .

.

3. Plant Leaf Diseases

Plant leaf diseases are a group of conditions that affect the structures of plant leaves, whether they are agricultural crops or plants in the wild. These diseases can cause damage to the leaves, impact plant growth, and reduce productivity. Plant leaf diseases encompass a wide range of pathogenic factors, including fungi, bacteria, viruses, worms, and other pests.

3.1. The importance of classifying plant leaf diseases:

Identification of threats: Classification helps identify various diseases affecting plant foliage, allowing for targeted management strategies.

Precision in treatment: By categorizing diseases, appropriate treatments can be applied, optimizing resource allocation and minimizing damage.

Monitoring and prevention: Classification facilitates disease monitoring, enabling early detection and prevention measures to safeguard plant health.

Research and development: Classifying diseases aids in research efforts, fostering a deeper understanding of disease etiology and supporting the development of resistant plant varieties.

International collaboration: Standardized disease classification enables consistent communication and collaboration among researchers and agricultural stakeholders globally, fostering shared knowledge and best practices.

In summary, disease classification serves as a cornerstone for effective plant health management, supporting timely interventions, informed decision-making, and sustainable agricultural practices.

3.2. Tomato Leaf Diseases

Tomatoes are of great importance in Algeria, serving as a major source of income and export, and providing significant employment opportunities, thereby contributing to food security. However, tomatoes are susceptible to numerous diseases that affect their quality and production:

- **Bacterial Spot:** Causes small black spots on leaves surrounded by a yellow halo, also affecting the fruit.
- **Early Blight:** A fungal disease that causes brown spots on lower leaves, appearing as concentric dark rings.
- **Late Blight:** A fungal disease causing dark lesions on leaves, leading to plant wilting and death.
- **Leaf Mold:** Caused by the fungus *Passalora fulva*, leading to yellowish-brown spots on leaves.

- **Septoria Leaf Spot:** Caused by the fungus *Septoria lycopersici*, producing small circular lesions on leaves.
- **Two-Spotted Spider Mite:** A pest that causes stippling and yellowing of leaves.
- **Target Spot:** Causes dark circular lesions on leaves and fruit, caused by the fungus *Corynespora cassiicola*.
- **Tomato Yellow Leaf Curl Virus:** Transmitted by whiteflies, causing leaf curling and yellowing.
- **Tomato Mosaic Virus:** Causes mosaic patterns of light and dark green on leaves.

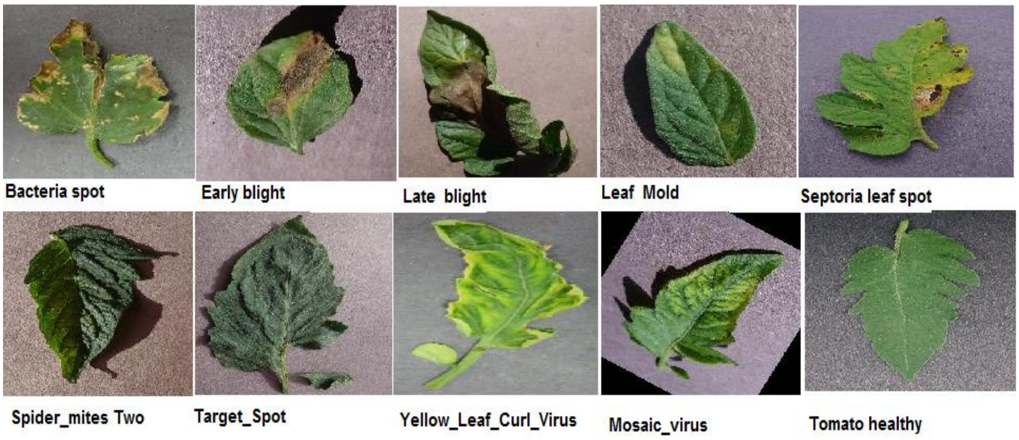

Fig 2 Some tomato leaf diseases [37]

3.3. Grape Leaf Diseases

Grapes hold significant economic importance in Algeria, used in producing fresh and dried fruits (raisins). However, grapevines face numerous diseases that threaten their productivity:

- **Esca (Black Measles):** Develops in warm, humid climates, affecting leaves, stems, flowers, and berries. It can cause leaves to dry out and turn yellow or brown, potentially leading to plant death.
- **Black rot:** a fungal disease that affects various plants, causing dark, sunken lesions on leaves, stems, and fruit. It's particularly common in warm and humid conditions.

- **Leaf Blight and Anthracnose:** Begins with small yellow spots on leaf edges that turn into concentric brown lesions, causing leaf drying and drop.

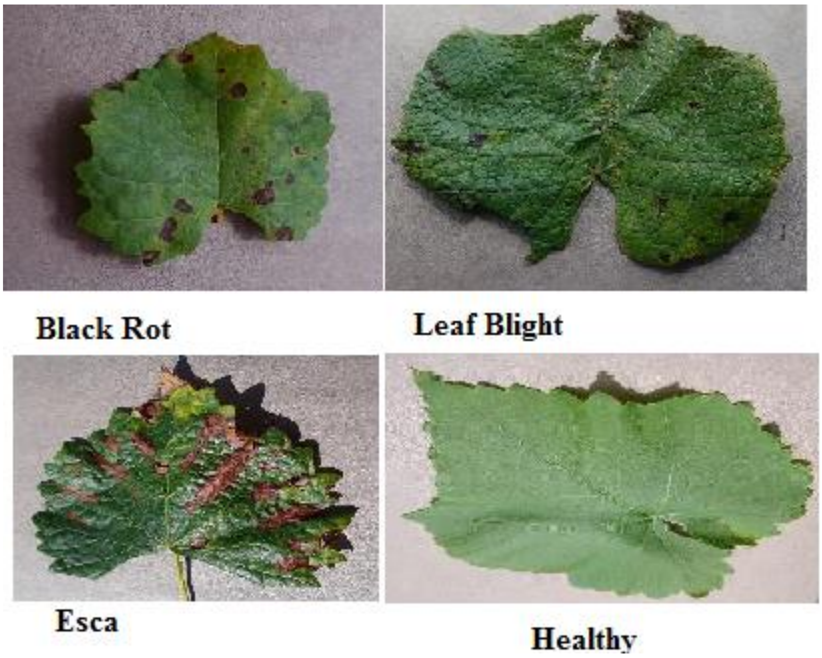

Fig 3 some grape leaf diseases[39]

3.4. Watermelon Leaf Diseases

Watermelon is a favorite fruit in Algeria, especially during summer, playing a vital economic role in supporting agriculture and providing income for farmers. However, watermelon is susceptible to several diseases affecting its quality and yield:

- **Anthracnose:** A fungal disease caused by *Colletotrichum orbiculare*, leading to dark, sunken lesions on leaves, stems, and fruits.
- **Downy Mildew:** A fungal disease presenting as bright yellow spots on the upper leaf surface, which later turn brown and necrotic, with gray or purple mildew on the lower surface.
- **Mosaic Virus:** A viral disease caused by various strains such as Watermelon Mosaic Virus (WMV) and Papaya Ringspot Virus (PRSV-W), transmitted by aphids and affecting infected seeds.

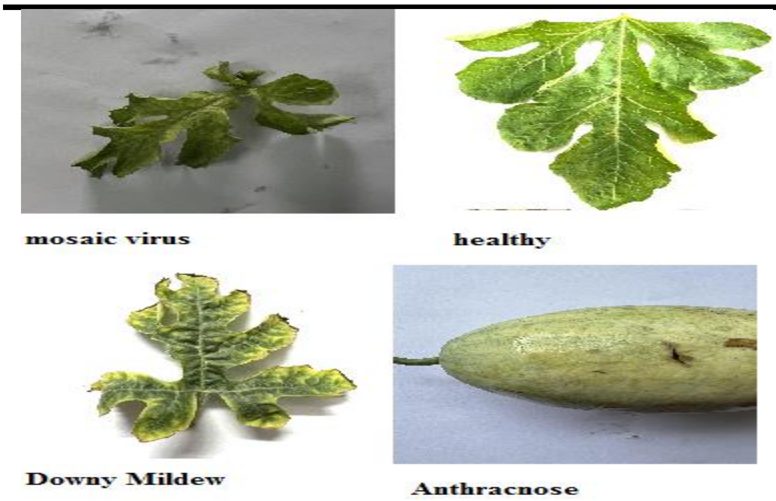

Fig 4 some watermelon diseases ***[38]***

4. Conclusion

In this chapter, we have reviewed the significance of food security and the challenges posed by plant diseases. Leaf diseases of watermelon, tomato, and grape illustrate the significant threat these diseases pose to agricultural productivity. There is an urgent need to develop accurate and effective diagnostic and treatment techniques, which we will address in the forthcoming chapters using machine learning and neural networks.

Chapter 02: Deep Learning .

1. Introduction

In recent years, deep learning has become an integral part of the advancement of artificial intelligence technologies, allowing deep neural networks to simulate the complex capabilities of the human brain. Deep learning is an extension of machine learning, distinguished by its ability to handle vast amounts of data and provide intelligent solutions for a wide range of applications. This chapter delves into the theoretical foundations of deep learning, its history, the functioning of deep neural networks, and practical applications in fields such as computer vision, natural language processing, and speech recognition.

2. Deep Learning

Deep learning, a branch of machine learning, employs intricate neural networks known as deep neural networks to replicate the intricate decision-making capabilities of the human brain. Most artificial intelligence (AI) applications today rely on some form of deep learning.

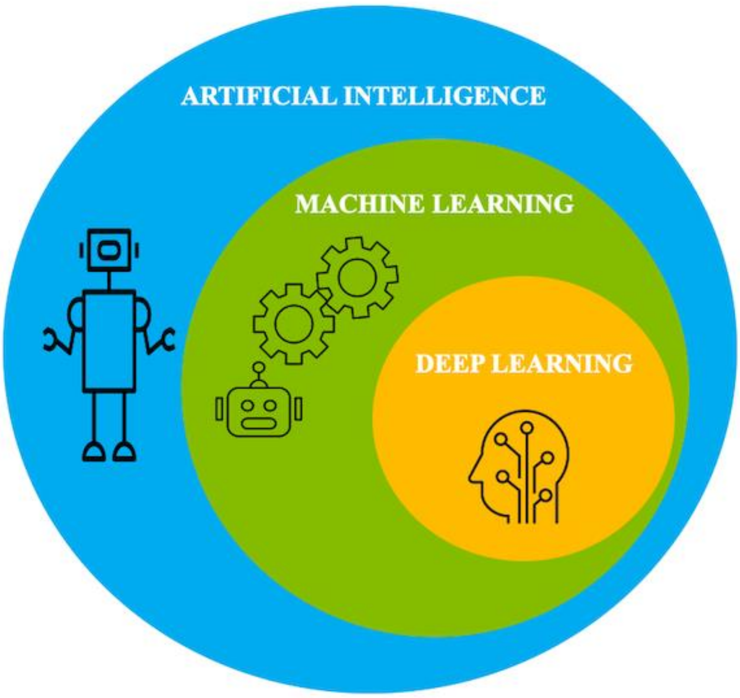

Fig 5 machine learning , deep learning and artificial intelligence.[40]

2.1. History of Deep Learning

Deep learning dates back to the mid-20th century, with early concepts emerging around neural networks. In the 1980s, Geoffrey Hinton made significant advancements in shape recognition, and coined the term "deep learning" in 2006. The introduction of long short-term memory (LSTM) for neural networks occurred in the 1990s. In the 2000s, the popularity of "deep learning" surged following the effectiveness of many-layered neural networks. By the 2010s, deep learning techniques achieved remarkable progress, reaching human-level performance in certain tasks. Today, deep learning is a key area of artificial intelligence, with profound impacts in fields like computer vision, speech recognition, and natural language processing.

2.2. Machine learning vs Deep learning

The differences between deep learning and machine learning

Machine Learning	**Deep Learning**
Utilizes statistical algorithms to uncover hidden patterns and relationships within data.	Leverages artificial neural network architectures to discover hidden patterns and relationships
Effective with smaller datasets.	Requires larger datasets for effective training.
Ideal for tasks with fewer labeled examples.	Well-suited for complex tasks such as image processing and natural language processing.
Generally requires less time for model training	Typically demands more time for model training
Relies on manual extraction of features from data	Automatically learns relevant features from data in an end-to-end manner.
Results are typically less complex and easier to interpret	Results can be more complex and harder to interpret, often viewed as a black box.
Can operate on standard CPUs or with lower computing power.	Requires high-performance computing resources with GPUs for efficient

	training.

Table 1 machine learning vs deep learning

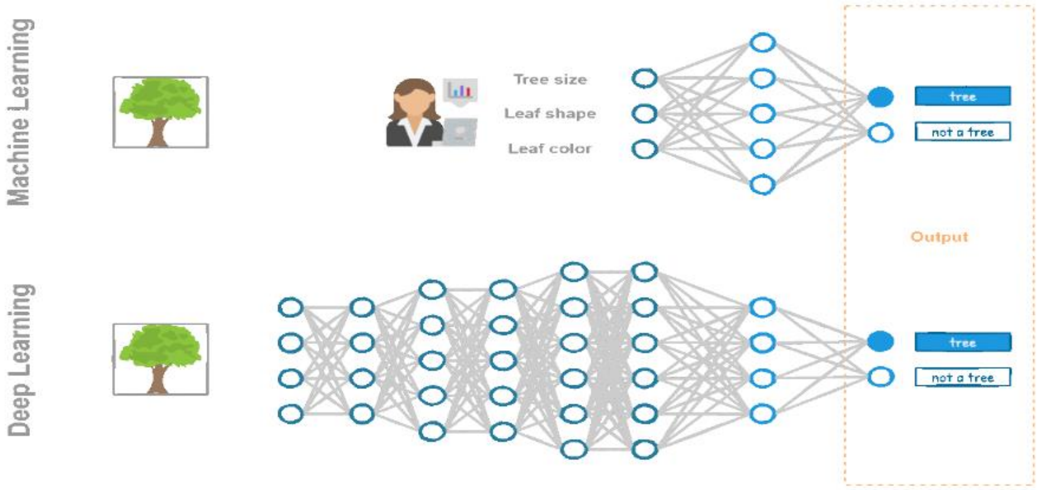

Fig 6 Machine Learning vs Deep Learning [41]

2.3. Neural networks

Firstly , we will talk about the neural networks :

2.3.1. Neural networks

Neural networks, also known as artificial neural networks (ANNs), represent a technique within artificial intelligence aimed at instructing computers on data processing, drawing inspiration from the workings of the human brain. This approach, classified under deep learning, employs interconnected nodes or neurons organized in layered structures reminiscent of the human brain. An adaptable system is established, enabling computers to learn from errors and enhance performance over time. Consequently, artificial neural networks strive to tackle intricate tasks like document summarization or facial recognition with greater accuracy.

2.3.2. How Neural Networks work

Neural networks are computational models inspired by the human brain. These networks consist of artificial units called "neurons," which work together to solve a specific problem. Let's take a look at the fundamental aspects of how neural networks operate :

- **Input Layer:**

  - Receives information from the external world.

  - Analyzes the data and passes it to the next layer.

- **Hidden Layer :**

  - Receive inputs from the previous layer or from other hidden layers.

  - Analyze and process the data further.

  - Neural networks can have multiple hidden layers.

- **Output Layer** :

  - Provides the final result of the data processing operations.

  - Can consist of a single or multiple output nodes, depending on the type of problem.

2.4. Deep learning models :

2.4.1. Deep Neural Networks (DNNs):

- Deep neural networks are the foundation of deep learning. They consist of multiple layers of interconnected neurons (nodes).

-Each layer in a DNN processes the input data and passes it to the next layer, progressively extracting higher-level features.

- DNNs can be used for tasks such as classification, regression, and feature learning.

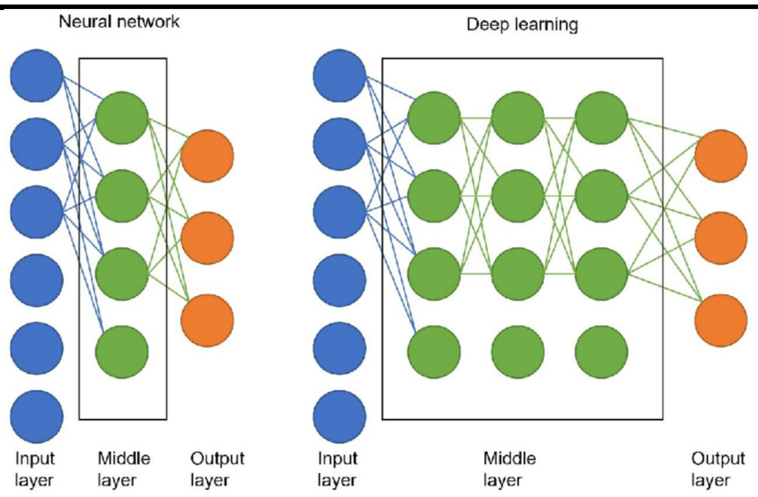

Fig 7 Neural network vs Deep learning [42]

2.4.2. Convolutional Neural Networks (CNNs):

- CNNs are specialized neural networks designed for processing structured grid-like data, such as images and videos.
- They consist of convolutional layers, pooling layers, and fully connected layers.
- CNNs excel at capturing spatial dependencies in data and are widely used in tasks such as image classification, object detection, and image segmentation.

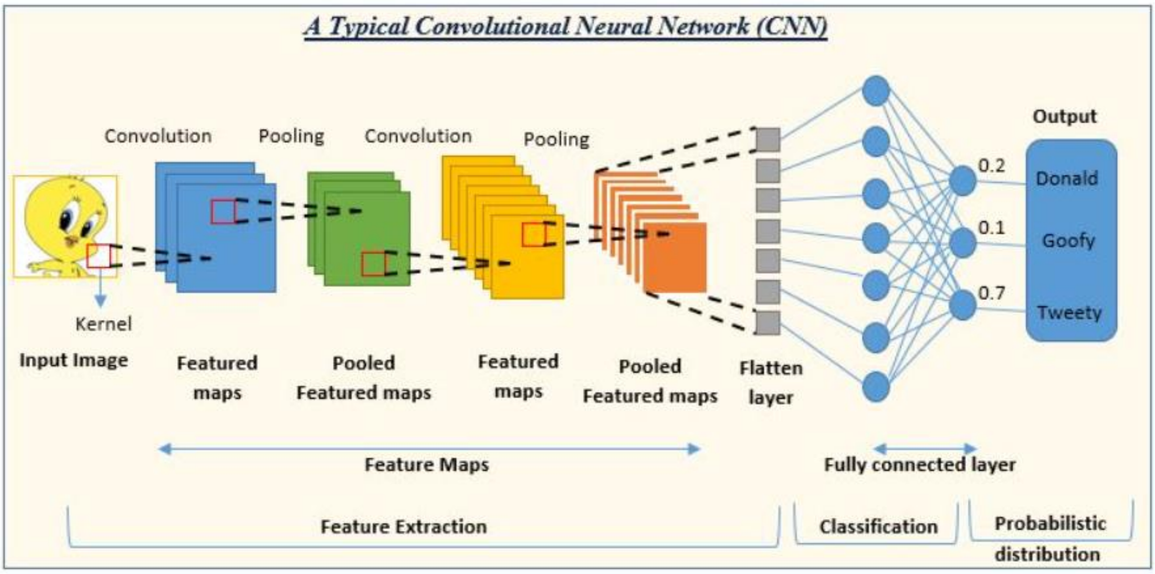

Fig 8 a typical covolution neural network[43]

2.4.3. Recurrent Neural Networks (RNNs):

- RNNs are designed to handle sequential data by maintaining internal state or memory.
- They have connections that loop backward, allowing information to persist.
- RNNs are suitable for tasks such as natural language processing, speech recognition, time series prediction, and sequence generation.

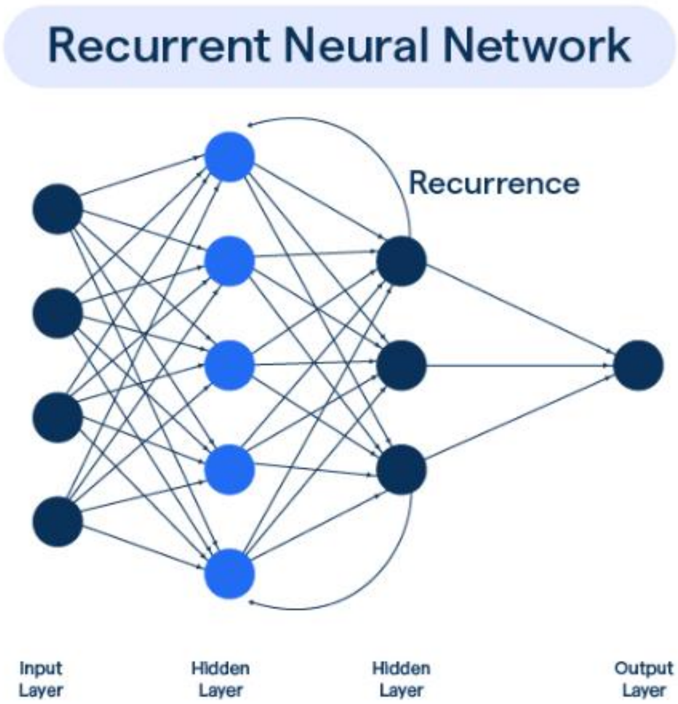

Fig 9 recurrent neural network [44]

Each of these components plays a crucial role in various deep learning applications, and they can be combined or modified to suit specific tasks and datasets.

2.5. Application of Deep Learning :

2.5.1. Computer Vision:

○ Object Detection and Recognition: Deep learning models excel at identifying and locating objects within images and videos. This capability is crucial for self-driving cars, surveillance systems, and robotics.

○ Image Classification: Deep learning algorithms classify images into categories (e.g., animals, plants, buildings). Applications include medical imaging, quality control, and image retrieval.

○ Image Segmentation: Deep learning techniques segment images into different regions, allowing specific features to be identified.

2.5.2. Natural Language Processing (NLP):

- Automatic Text Generation: Trained deep learning models can generate summaries, essays, and other text based on existing corpora.
- Language Translation: Deep learning enables accurate translation between languages, facilitating global communication.
- Sentiment Analysis: By analyzing text, deep learning models determine whether it conveys a positive, negative, or neutral sentiment. This is valuable for customer service, social media monitoring, and political analysis.
- Speech Recognition: Deep learning algorithms transcribe spoken words, enabling speech-to-text conversion, voice search, and voice-controlled devices.

2.5.3. Reinforcement Learning:

Deep learning plays a significant role in reinforcement learning, where agents learn to make decisions by interacting with an environment. Applications include game playing, robotics, and financial trading.

2.5.4. Other Notable Applications:

- **Self-Driving Cars**: Deep learning powers autonomous vehicles, handling complex scenarios through exposure to millions of data points.
- **Healthcare:** Deep learning aids in personalized medicine, detecting diseases from medical images, and predicting patient outcomes.
- **Entertainment**: From colorizing black-and-white images to adding sound to silent

movies, deep learning enhances creative content.

- **Fraud Detection**: Deep learning models analyze patterns to identify fraudulent transactions.
- **Virtual Assistants**: NLP-based virtual assistants like Alexa and Siri rely on deep learning for natural language understanding.
- **Virtual Assistants**: NLP-based virtual assistants like Alexa and Siri rely on deep learning for natural language understanding.

3. Computer vision :

Computer vision is a field within artificial intelligence and computer science that focuses on enabling machines to interpret, understand, and analyze visual data from the world around us. The goal of computer vision is to create intelligent systems capable of performing tasks that typically require human-level visual perception, such as object detection, recognition, tracking, and segmentation. Computer vision encompasses a wide range of techniques and approaches, allowing models to learn from large amounts of visual data, such as images and videos. There have been many recent advancements in the field of computer vision, thanks to progress in deep learning and neural networks.

3.1. Computer Vision Applications:

Here are a few notable examples:

3.1.1. Image Generation using GPT-3 :

- ✓ The GPT-3 language model developed by OpenAI can generate realistic images from textual descriptions.
- ✓ By guiding the neural network with a linguistic description, the model can produce detailed images representing the described scenario.

3.1.2. Agriculture :

In agriculture, computer vision aids in crop monitoring, pest detection, yield estimation, and automated harvesting through techniques such as satellite imagery analysis and drone-based imaging. Additionally, it assists in the early detection and classification of **plant leaf diseases** to mitigate losses for farmers.

3.1.3. Autonomous Vehicles :

- ✓ Computer vision is a key technology enabling autonomous vehicles to navigate and understand their environment.
- ✓ Recent advancements in this field include platforms like NVIDIA DRIVE AGX, which utilize deep learning algorithms for real-time image processing and decision-making in autonomous vehicles.

3.1.4. Medical Imaging :

Computer vision is also used to enhance medical imaging, with recent developments including AI-based systems for lung cancer diagnosis and diabetic retinopathy detection.

3.1.5. Robotics :

- ✓ Computer vision is crucial for enabling robots to perceive and

  interact with the world around them.
- ✓ Recent advancements include deep learning-based systems for object recognition, visual grasping, and manipulation.

3.2. computer vision tasks :

Four common computer vision tasks:

- ✓ **Image Classification :** Assigning a label or category to an entire image based on its content.
- ✓ **Object Detection :** Identifying and localizing specific objects within an image.
- ✓ **Semantic Segmentation** : Pixel-level classification of objects in an image to create a detailed mask of each object.
- ✓ **Face Recognition**: Identifying and verifying individuals based on their facial features.

These tasks are crucial in enabling computers to understand and interpret visual information, leading to various applications across industries such as healthcare, automotive, security, and entertainment.

4. Transfert Learning :

Transfer learning enables the reuse of pre-trained models and datasets for new tasks and domains, reducing the need for extensive data and annotations. By starting with a pre-trained model, known as a backbone model, the amount of new data required for training a new model is minimized, resulting in improved performance on the target task. While transfer learning in machine learning involves fine-tuning pre-trained models on related tasks, transfer learning in the human brain allows individuals to apply previously learned knowledge and skills to new contexts. Despite their related concepts, the underlying mechanisms differ between machine learning and human learning.

4.1. Pre-trained neural networks:

There are several pre-trained neural networks that have gained significant popularity and have been widely used in various computer vision applications. Here are some of the most famous ones:

VGG(Visual Geometry Group) :

Is a family of deep neural networks that achieved top performance in the 2014 ImageNet Challenge. The VGG models are characterized by their deep architecture, with up to 19 layers, and have been widely used for various computer vision tasks, such as object recognition and localization.

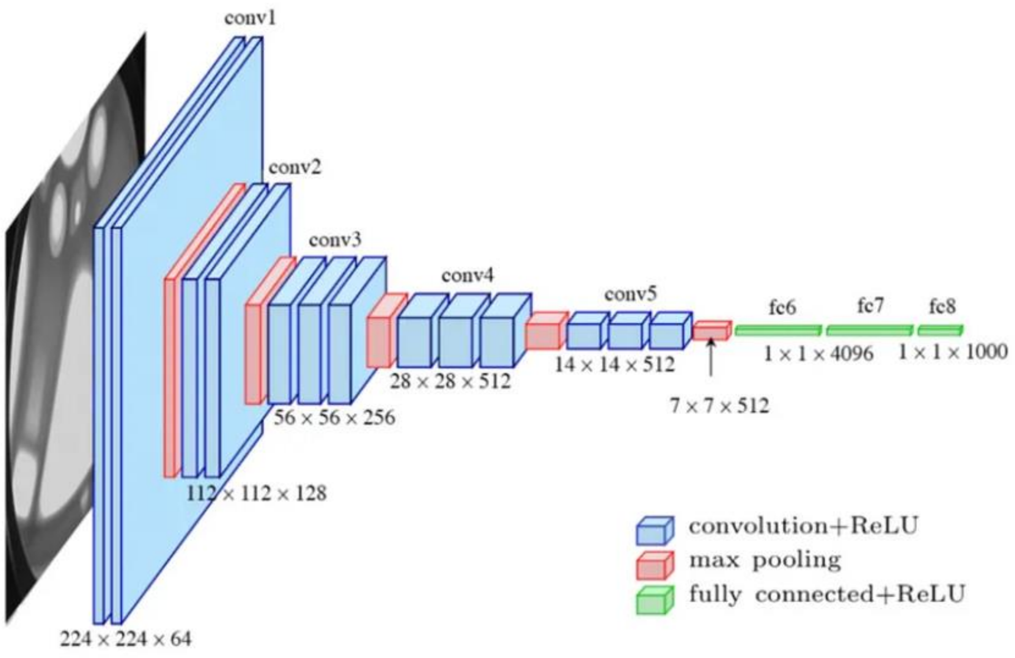

Fig 10 The architecture of VGG16 [45]

ResNet (Resdual Networks):

is another family of deep neural networks that won the ImageNet Challenge in 2015. ResNet models are characterized by their residual blocks, which allow for easier training of very deep neural networks with over 100 layers. ResNet models have been widely used for various computer vision tasks, such as object recognition and detection.

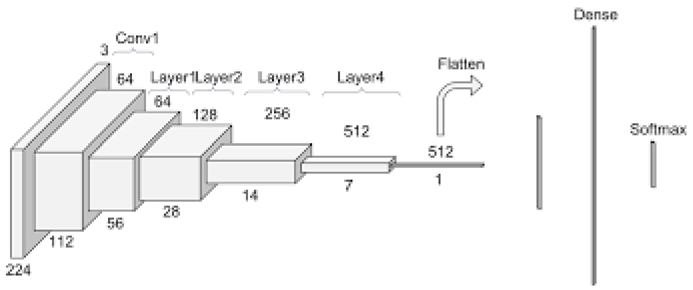

Fig 11 **the architecture of ResNet [46]**

Inception:

Inception is a family of deep neural networks that was introduced by Google in 2014. The Inception models are characterized by their use of multiple parallel convolutional layers at different scales to extract features from images. Inception models have been widely used for various computer vision tasks, such as image classification and object detection.

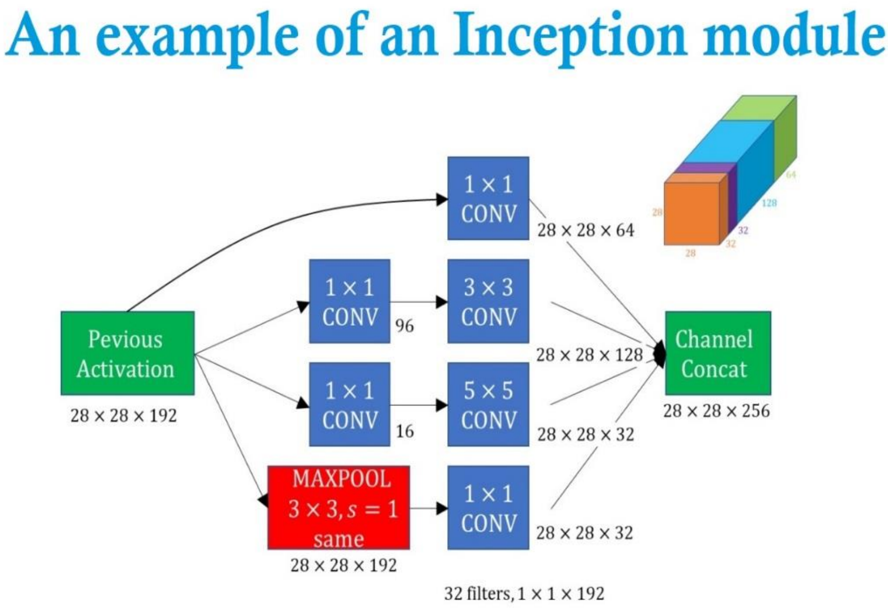

Fig 12 An example of inception module [47]

<u>MobileNet</u>:

MobileNet is a family of deep neural networks that was designed for mobile and embedded devices with limited computing resources. MobileNet models are characterized by their lightweight architecture, which enables fast inference on mobile devices. MobileNet models have been widely used for various computer vision tasks, such as object recognition and detection on mobile devices.

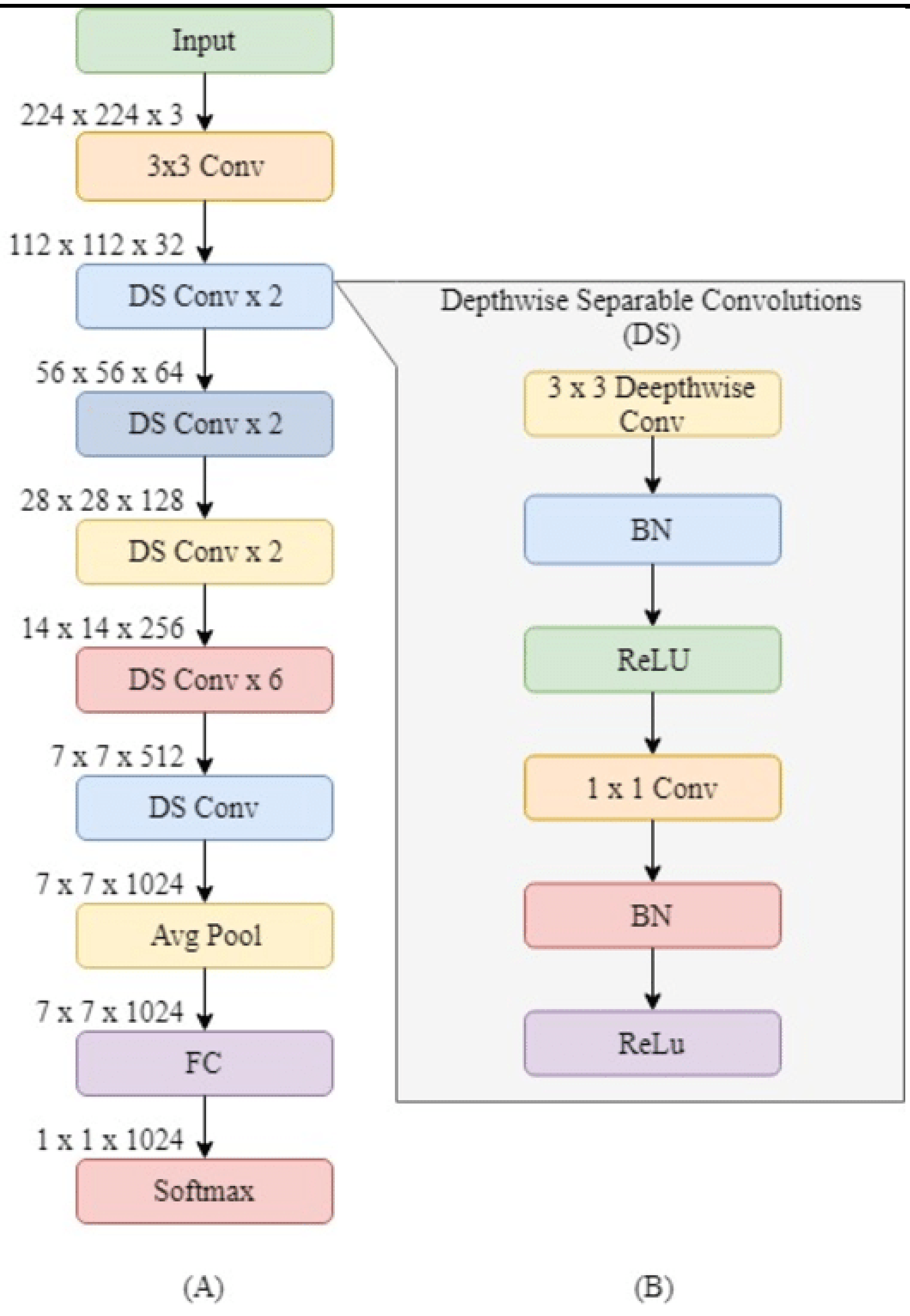

Fig 13 the architecture of mobilenet.[48]

5. Conclusion :

In conclusion, deep learning represents one of the most significant technological leaps of our era. With its ability to learn from data and improve performance over time, deep learning has become an essential tool in developing advanced applications that directly impact our daily lives. From enhancing healthcare services to developing autonomous vehicles, deep learning continues to offer new possibilities and innovative solutions. As research and development in this field progress, we can expect further improvements and discoveries that will enhance the role of artificial intelligence in the future.

.

Chapter3 :plant leaf diseases using computer vision.

1. Introduction :

In this chapter, we will examine previous computer vision techniques for diagnosing plant leaf diseases. Through analysis of outcomes, we aim to get insights for refining our own model, and thus towards achieving early diagnosis and elevating plant health management.

2. The importance of early diagnosis of plant diseases:

Early diagnosis of plant diseases is the cornerstone of modern agriculture and significantly contributes to:

- **Controlling the spread of diseases:** Early detection can limit the spread of diseases across crops, reducing the likelihood of widespread infections.
- **Reducing economic losses:** By minimizing damage to crops, farmers can maintain production levels and quality, thus mitigating economic losses.
- **Improving resource utilization:** Early intervention reduces the need for large quantities of pesticides, benefiting the environment and public health.
- **Increasing agricultural productivity:** Preserving plant health leads to increased productivity and improved crop quality

3. Computer vision techniques used in previous works:

3.1. Fine-tuned architectures :

3.1.1. MobileNet V2:

MobileNetV2 is a convolutional neural network (CNN) architecture designed to be lightweight and efficient for mobile and embedded devices. It was developed by researchers at Google as an enhancement to the original MobileNet model. This architecture is characterized by its depthwise separable convolutions and inverted residual blocks, which help reduce the computational cost while maintaining performance.

- **Using MobileNet V2 for Plant Disease Classification:**

In the study conducted by the researchers, including Siti Zulaikha Mohd Zakaria et al [1], a pre-trained MobileNet V2 model was used to classify tomato leaf diseases. The model was fine-tuned and trained using a dataset of tomato leaf images from the PlantVillage database, which included 4671 images categorized into healthy leaves and leaves infected with three types of diseases: leaf mold, late blight, and mosaic virus. The images were divided into two sets: a training set containing 3471 images and a test set containing 1200 images. The image dimensions were adjusted to fit the requirements of the MobileNet V2 model (224 × 224 pixels), and transfer learning techniques were employed to achieve accurate disease classification.

3.1.2. **EfficientNet**:

EfficientNet is a deep learning model that strikes a balance between model size and performance. It relies on the concept of compound scaling of depth, width, and resolution. It excels in classification and diagnostic tasks, leveraging its ability to efficiently capture complex shapes and patterns in data. This model automatically adjusts its size based on these three factors, making it easier to select the appropriate model for a specific task. EfficientNet's outstanding performance makes it suitable for tasks requiring quick and accurate decision-making based on complex data.

- **Using EfficientNet for Plant Disease Classification:**

In the study conducted by the researchers, including Ümit Atila et al[36] , the pre-trained EfficientNet B5 and B4 deep learning models were used to classify plant leaf diseases. The models were fine-tuned and trained using the PlantVillage dataset, which included a total of 55,448 images categorized into 38 classes, encompassing various diseased and healthy plant leaves. Additionally, an augmented version of the dataset, increasing the total number of images to 61,486, was used to enhance the models' performance. The image dimensions were adjusted to fit the requirements of the EfficientNet B5 and B4 models, and transfer learning techniques were employed to achieve accurate disease classification.

3.1.3. **AlexNet** :

AlexNet is a deep neural network model introduced in 2012. It is known for its depth and utilization of techniques like ReLU activation and max-pooling. It has been used in various applications including image classification, where it can recognize patterns in images and classify them with high accuracy.

- **Using AlexNet for Plant Disease Classification:**

In the study conducted by Mohammed Brahimi et al[2], the researchers employed the AlexNet model, which had been pre-trained on the ImageNet dataset, for the classification of tomato leaf diseases. The methodology began with data preparation, where a dataset consisting of 14,828 images from the PlantVillage database was utilized. These images depicted tomato leaves categorized into nine distinct disease classes. Following this, the images were resized to meet the input dimensions required by AlexNet, typically 227 × 227 pixels, the transfer learning was utilized, leveraging the pre-trained features of the AlexNet model. The final fully connected layer of AlexNet was then replaced to align with the nine disease classes under consideration. Subsequently, the model underwent fine-tuning using the tomato leaf image dataset to optimize its weights for the specific classification task.

.

3.2. Ensemble learning :

- **Definition :**

Ensemble learning is a methodology in machine learning that combines multiple individual models to improve the overall performance of the system. The main goal of using this technique is to reduce variance and bias while increasing the accuracy and stability of predictions. There are several types of ensemble learning techniques, including:

- **Boosting:**

Enhances model performance by building a series of models where each model corrects the errors of the previous one.

- **Bagging:**

Creates multiple training models by taking random samples with replacement from the original dataset and training an independent model on each sample, then aggregating the results of these models.

- **Random Forest:**

A type of bagging that uses multiple decision trees and combines their results to produce the final prediction.

- **Voting:**

Aggregates the predictions of several models and uses voting to determine the final result, which can be based on majority voting or averaging predictions.

- **Using Ensemble learning for Plant Disease Classification:**

In the study conducted by Jerry Gao et al [3]., ensemble learning techniques were employed to achieve better results in the detection and classification of grape leaf diseases. The researchers developed four pre-trained deep learning models, including VGG16, MobileNet, and AlexNet, which were modified and trained using a dataset of diseased grape leaf images. Subsequently, the researchers created an ensemble model that combines these four models, likely using techniques such as majority voting or averaging to aggregate the results of the individual models. By combining different models, they were able to significantly improve the accuracy of detection and classification by reducing variance and achieving more reliable predictions. This approach allowed the researchers to leverage the strengths of each model and mitigate their weaknesses, leading to superior performance in the early detection of grape leaf diseases.

3.2.1. INAR-SSD : (Integrated Neural Architecture for Real-time Single Shot multibox Detector):

- **INAR-SSD :**

In the INAR-SSD model, the GoogLeNet Inception architecture and Rainbow Concatenation technique are combined to form an enhanced model capable of real-time detection and classification of objects with high accuracy and optimal speed. This integration enhances the model's ability to recognize apple leaf diseases accurately, thereby improving its performance and speeding up the disease detection process.

- **Using INAR-SSD for Plant Disease Classification:**

In a study conducted by Bin Liu et al[8], a model for real-time detection of apple leaf diseases was developed using deep learning techniques and improved convolutional neural networks (CNNs). The researchers first constructed an Apple Leaf Disease Dataset (ALDD) using images taken in both laboratory and field conditions, along with data augmentation and image labeling techniques. They then developed a new model for detecting apple leaf diseases using a deep convolutional neural network, incorporating the Inception architecture from GoogLeNet and the Rainbow concatenation technique to enhance the detection of multi-scale diseases and small lesions ,their proposed model, named INAR-SSD (Single Shot Multibox Detector with Inception and Rainbow concatenation), was trained using a dataset containing 26,377 images of diseased apple leaves. Furthermore, the models used in this study involved pre-trained architectures. The basic VGGNet was enhanced by integrating the Inception module from GoogLeNet to form VGG-INCEP, which improves the feature extraction performance for multi-scale disease spots. This pre-trained architecture was further modified and combined with other techniques like Rainbow concatenation to boost the detection performance, especially for small disease spots.

3.3. Fine-Grained architectures :

3.3.1. vision transfomers :

Vision Transformers (ViTs) are models that apply transformer architecture, originally designed for natural language processing, to image processing. They divide an image into small patches and use self-attention mechanisms to understand the relationships between these patches.

SEViT :

A type of Vision Transformer that employs self-supervised learning techniques, such as contrastive learning, to train the model on unlabeled data, extracting useful features.

- **Using SEViT for Plant Disease Classification :**

In the study conducted by Qingtian Zeng et al[4], a model named SEViT (Squeeze-and-Excitation Vision Transformer) was proposed for large-scale and fine-grained plant disease classification. The researchers integrated two main networks to achieve this goal , First, they utilized an enhanced SE-ResNet101 network, which includes a Squeeze-and-Excitation (SE) module. This SE module works by weighting the channel information in the input image using two fully connected layers, allowing the network to focus on channels that contain significant information. This enhances the expression of relevant features, thereby improving the ability to distinguish between similar diseases .Second, the features extracted from the SE-ResNet101 network were passed to a Vision Transformer (ViT) network. The ViT relies on a Self-Attention module that processes the entire image as a context domain, enabling it to capture global contextual information and establish long-range dependencies on the target. This approach improves classification accuracy, The researchers employed a transfer learning strategy by loading pre-trained weights from the ImageNet21K dataset for both the SE-ResNet101 and ViT networks, which significantly enhanced the model's performance.

IBTN :

A type of Vision Transformer that uses iterative backward attention to progressively refine the focus on important parts of an image

- **Using IBTN for Plant Disease Classification :**

In the research conducted by Ying Li et al [6], the Interactive Bilinear Transformation Network (IBTN) technique was utilized to improve plant disease recognition and protection in garden design. This technique relies on integrating features from neural network transformations and deep transformation techniques to enhance feature extraction and interaction. The IBTN model is based on a progressive architecture that gradually reduces token length while increasing their dimensions, maintaining the model's capacity. This technique is used to extract key features from segmented image parts and process them to capture both local and global features of the image. The study demonstrates the effectiveness of the IBTN model in recognizing plant diseases across various plant types with high accuracy and efficiency, representing a significant advancement in the field of plant disease detection using advanced deep learning techniques. However, the study also notes some limitations, such as not covering all types of plant diseases, indicating the need for further research to expand the application scope to include a broader range of diseases and plants.

3.3.2. Contrastive Learning :

Triplet Loss is a type of contrastive learning used in deep learning to train neural networks to distinguish between different categories of data. The method relies on selecting three data samples (a triplet) consisting of:

1. An anchor sample (A).
2. A positive sample (P) from the same category.
3. A negative sample (N) from a different category.

The objective of Triplet Loss is to minimize the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative sample. The basic equation used is:

$$L = \max(d(a,p) - d(a,n) + \alpha , 0)$$

Where:

- ***d*** represents the distance (usually Euclidean distance).
- ***a*** is the anchor sample.
- ***p*** is the positive sample.
- ***n*** is the negative sample.
- ***α*** is the margin that sets the minimum difference between the distances.

- **Using Contrastive Learning for Plant Disease Classification :**

Yu Xia et al [5] conducted a study on the classification of potato leaf diseases, proposing a model called **CLCNN** (Contrastive Learning Convolutional Neural Network). This model incorporates a Projection Head and a Contrastive Loss function into the Vgg16 network. The model comprises three main units: Encoder, Classifier, and Projection Head. Images of diseased potato leaves were collected from the internet and from potato farms at Yunnan Normal University. The data was split into training and testing sets in a 6:2 ratio,The study included two stages of potato disease (early blight and late blight), each divided into 4 grades, resulting in 8 categories.

3.3.3. Bi-CNN's :

Deep Bi-CNN stands for "Deep Binary Convolutional Neural Network." This architecture utilizes a deep neural network with binary convolutional layers, where weights and inputs are converted into binary values instead of real values. This conversion helps reduce the size of data and computational cost, leading to faster training and prediction processes.

In a study by Ramesh Babu Ch and colleagues [7], Bi-Convolutional Neural Networks (Bi-CNNs) were employed to classify plant leaf diseases. The team adapted VGG and ResNet models to extract visual features from images of diseased leaves, then combined the results of these models in a bi-convolutional stage. The combined outputs were then fed into a fully connected dense network for disease classification. The model was evaluated using standard classification metrics on an extensive test dataset.

4. Results of the related work:

Architectures used	Models	Reference	Species	Datasets source	Accuracy
Fine-tuned architectures	Mobilenet V2	Siti Zulaikha Muhammad Zaki et al [1]	tomato	The PlantVillage	95.94%
	EfficientNet	Ümit Atila et al [36]	_	The PlantVillage	98.42%

	Alexnet	Mohammed Brahimi et al[2]	Tomato	published in (Goodfellow, Bengio, and Courville 2016)	98.66%
Ensemble learning	VGG16, MobileNet, and AlexNet	Jerry Gao et al [3]	Grape	_	99%
	INAR-SSD	Bin Liu et al [8]	_	captured in the laboratory	78.80%
Fine-grained architectures	SEViT	Qingtian Zeng et al [4]	_	From Baidu Baike, Wikipedia, and various agricultural websites.	88.3%
	IBNT	Ying Li et al [6]	Tomato	from the 2018 AI challenge competition,	98.67%
	Bi-CNN's	Ramesh Babu Ch et al [7]	Tomato	The dataset was publicly available for research	91.20%

Table 2 the summary results of related works

5. Conclusion :

From this chapter, we can deduce that research in plant pathology is of great importance for understanding the diseases that affect plants and identifying the most effective ways to combat them. Such research contributes to improving agricultural production and preserving food security. Technological progress, especially the use of convolutional neural networks and deep learning, has revolutionized the diagnosis and classification of plant diseases, providing precise tools for early disease detection and reducing losses. However, there are still challenges facing researchers, such as developing models capable of dealing with the large diversity of disease types and their environmental conditions, which offer opportunities for innovative solutions. The results obtained from this research have the potential to improve agricultural practices and enhance plant resistance to diseases, contributing to greater sustainability in agricultural systems. The conclusions drawn show that innovation in the field of plant pathology can significantly change how agricultural diseases are managed and enhance the ability of plants to grow healthily, benefiting agricultural production and global food security**.**

Chapter 4

Implementation and Results Analysis

1. introduction :

In this chapter, we provide a detailed explanation of the implementation process and analyze the results obtained from our study on plant leaf disease classification. We aim to offer a thorough understanding of the data preprocessing steps, the adaptation of a pre-trained model for our specific task, and the evaluation of the model's performance on different datasets. We will begin by presenting an overview of the datasets used, followed by the customization process of the pre-trained VGG19 model. Finally, we will discuss the experimental results and the development of a practical application for disease diagnosis.

2. Datasets Overview :

We worked on three different types of datasets related to Plant leaf Disease .

2.1 Grape Dataset:

This dataset is designed for the task of grape leaf disease recognition and classification and is available on Kaggle, contributed by PUSHPA LAMA [39]. It contains 9027 images of grape leaves, classified into 3 different disease classes (black rot, Esca, and Leaf Blight) and 1 class for healthy leaves.

The dataset is divided into two parts: 80% for training and 20% for testing as follows :

Diseases	Train	Test
Blackrot	1888	472
Esca	1920	480
Leaf Blight	1722	430
Healthy	1692	423
	7222	1805

Table 3 informations of grapes dataset

2.2 Tomato Dataset:

This dataset provided by Mohammed HMIMOU on Kaggle comprises 22930 images of tomato plants affected by various diseases, along with images of healthy tomato plants[37]. The dataset categorizes tomato plant images into 8 disease classes(Bacterial Spot ,Early Blight,Late Blight, Leaf Mold , Septoria Leaf Spot , Spider Mites (Two-Spotted Spider Mite),Target Spot, Mosaic Virus) and 1 "Healthy" class.

The dataset is divided into two parts: 80% for training and 20% for testing as follows :

Diseases	Train	test
Bacterial Spot	1702	425
Early Blight	1920	480
Late Blight	1851	463
Leaf Mold	1882	470
Septoria Leaf Spot	1745	436
Spider Mites	1741	435
Target Spot	1827	457
Mosaic Virus	1790	448
YellowLeaf	1961	490
healthy	1926	481
	18345	4585

Table 4 informations of tomato dataste

2.3 Watermelon Dataset:

This dataset was provided by Sujay Kapadnis on Kaggle[38]. It is a recent contains images of watermelon plants affected by various diseases, as well as images of healthy watermelon plants. There are a total of 6930 images in the dataset. The dataset is divided into two parts original and augmented ,each part contains the same four diseases.

We have divided into two parts: 80% for training and 20% for testing as follows :

	Diseases	Train	Test	
Original	Anthracnose	124	31	1155
	Downy_Mildew	304	76	
	Mosaic_Virus	332	83	
	Healthy	164	41	
Augmented	Anthracnose	620	155	5775
	Downy_Mildew	1520	380	
	Mosaic_Virus	1660	415	
	Healthy	820	205	

Table 5 informations of watermelon dataset

3. Model :

We have adopted a pre-trained model on fine-grained dataset, which is flowers dataset . We believe that an architecture dedictaed for capturing every subtule fine-grained detail, would be able to classify plants leafs deseases with high accuracy. The adopted model is VGG19 pretrained on ImageNet, and it has been fine-tuned for our purpose based on similar backbone architecture for the flowers dataset.

3.1 Pre-trained VGG19 Model Loading :

Pre-trained VGG19 model is loaded, trained on the ImageNet dataset, with excluding the top layers to customize the model for plant leaf classification task**.**

```
vgg_model = VGG19(weights = "imagenet", include_top=False, input_shape = (224, 224, 3))
```

Fig 14 pre-trained vgg19 model loading

3.2 Freezing and Selective Training of Layers:

Most of the model's layers are frozen to prevent weight modifications during training, except for 5 selected layers to make them trainable for fine-tuning the model on the new data.

```
print("layers freezed")
for layer in vgg_model.layers[:-5]:
    print(layer.name)
    layer.trainable=False

print("layers to be trained")
for layer in vgg_model.layers[1:4]:
    print(layer.name)
    layer.trainable=True
```

Fig 15 freezing and selective training of layers

3.3 Creating Input Layer and Passing it through VGG19 Model:

An input layer is created to receive images of dimensions (224x224x3), and then passed through the VGG19 model to obtain extracted features.

```
input = Input(shape=(224, 224, 3),name = 'image_input')
output_vgg16_conv = vgg_model(input)
```

Fig 16 creating input layer

3.4 Adding Custom Layers:

Several layers are added to customize the model:

Batch Normalization: To improve training speed and stability.

MaxPooling2D: To reduce data dimensions and increase abstraction.

Dropout: To reduce overfitting.

Flatten: To reshape data suitable for dense layers.

Dense layers: The number of units in dense layers is reduced to simplify the model and prevent overfitting, with a final layer having 10 units for classifying the required categories.

```
x = BatchNormalization()(output_vgg16_conv)
x = MaxPooling2D(pool_size=(2, 2), padding='same')(x)
x = Dropout(0.2)(x)

x = Flatten()(x)

x = Dense(512, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.3)(x)

x = Dense(256, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.3)(x)

x = Dense(10, activation='softmax')(x)
```

Fig 17 adding custom layers

3.5 Model Definition and Displaying Summary:

The model is defined using input and custom layers, then a summary of the model is displayed showing details of layers and the number of trainable and non-trainable parameters.

```
model = Model(inputs=input, outputs=x)

model.summary()
```

Fig 18 model definition and displaying summary

- After configuring the model by adding custom layers and freezing most of the pre-trained VGG19 layers, here is the final model summary, which shows the details of the various layers and the number of trainable and non-trainable parameters.

Layer (type)	Output Shape	Param #
image_input (InputLayer)	(None, 224, 224, 3)	0
vgg19 (Functional)	(None, 7, 7, 512)	20,024,384
batch_normalization (BatchNormalization)	(None, 7, 7, 512)	2,048
max_pooling2d (MaxPooling2D)	(None, 4, 4, 512)	0
dropout (Dropout)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 512)	4,194,816
batch_normalization_1 (BatchNormalization)	(None, 512)	2,048
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
batch_normalization_2 (BatchNormalization)	(None, 256)	1,024
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2,570

Total params: 24,358,218 (92.92 MB)
Trainable params: 13,809,226 (52.68 MB)
Non-trainable params: 10,548,992 (40.24 MB)

Fig 19 summary of model

3.6. Train model :

The model was trained using the specified dataset, with validation data utilized to assess model performance. The number of training epochs was set to 15, and a set of early responses were executed during the training process, including early stopping, tensorboard logging, and checkpoint verification.

```
model.fit(train_set, #specifing the training data
      validation_data= test_set, #specifing the validation data
      epochs=15, #the amount of epochs is the amount of repetions in this case 10
      callbacks=[earlystopping,tensorboard,chkpt]) #this is saying that the model must do the callbacks
```

Fig 20 training model

3.7. Save weights model:

This code saves the weights of a deep learning model trained using the Keras library in HDF5 format and converts the saved model to TensorFlow Lite format :

```
model.save_weights('vgg19TransferTomato.weights.h5')
```

Fig 21 save weights

4. Results and Analysis:

4.1 (Tomato Dataset):

The loss and accuracy plots indicate that the VGG19 model is learning effectively from the data. The training loss decreases apidly , reaching near zero, The validation loss follows a similar trend, suggesting good generalization. the accuracy plot demonstrates a rapid increase in training accuracy, reaching near 97%, and the validation accuracy follows a similar trend, demonstrating the model's ability to adapt well. Although there are minor fluctuations in the validation curve, the overall performance is consistent, indicating no significant overfitting.

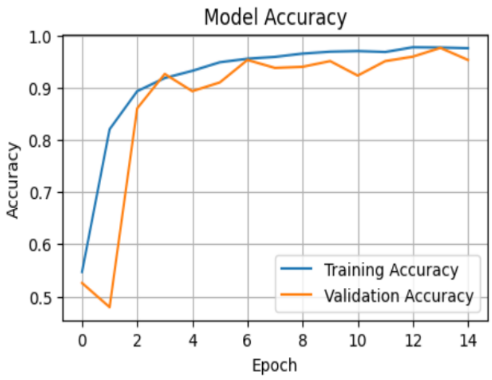

Fig 22 accuracy plot tomato

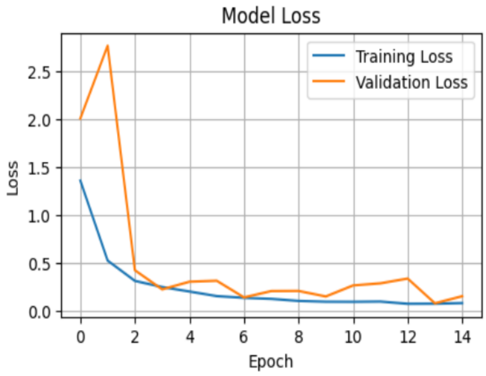

Fig 23 loss plot tomato

4.2. Grape Dataset:

The loss plot shows that the training loss decreases rapidly and steadily, reaching near zero, indicating effective learning by the model. The validation loss follows a similar trend, stabilizing at a low value with minor fluctuations, suggesting good generalization to unseen data with no significant overfitting. Similarly, the accuracy plot demonstrates a rapid increase in training accuracy, reaching near 100%, and the validation accuracy follows a similar trend, stabilizing at a high level close to the training accuracy with minor fluctuations. This indicates the model generalizes well and demonstrates strong performance.

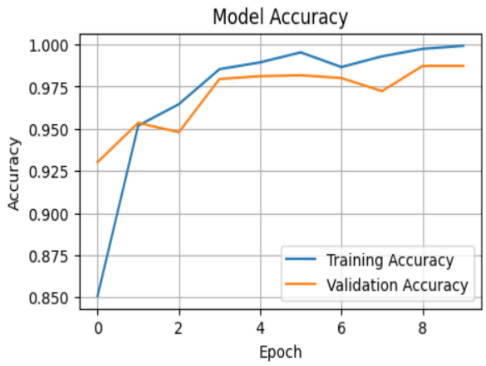

Fig 24 accuracy plot grape

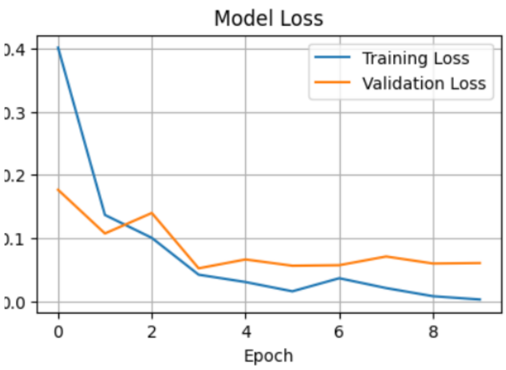

Fig 25 loss plot grape

4.3. Watermelon Dataset:

The model showed excellent performance on the training set for the watermelon dataset, achieving a high accuracy of 99.45%. However, its performance on the validation set was lower, with an accuracy of 86.8% and relatively high loss, indicating the presence of overfitting. However, when testing the model on other datasets, it showed excellent results, confirming its strength and effectiveness. It appears that the Watermelon dataset contains significant diversity in its characteristics, may be small in size, making it challenging for the model to generalize the learned patterns. Therefore, it can be concluded that the challenges associated with the Watermelon dataset stem from the nature of the data itself, and improving or increasing the quality of the data may be necessary to enhance performance.

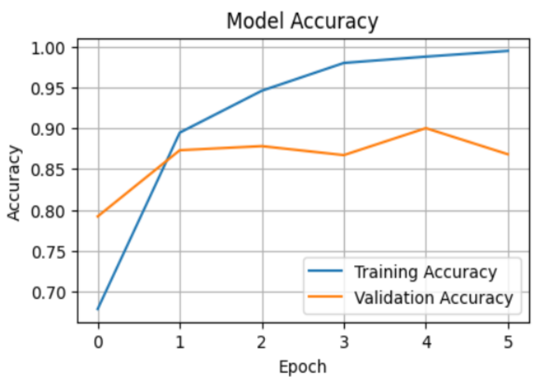

Fig26accuracy plot watermelon

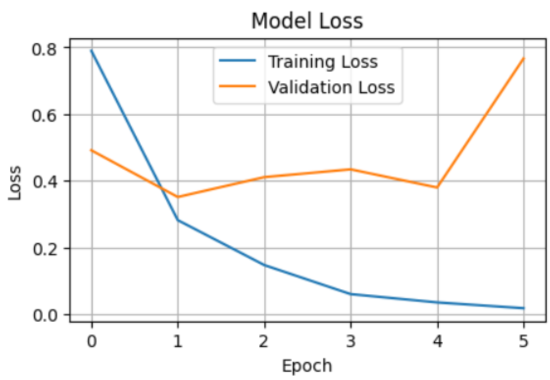

Fig 27 loss plot watermelon

4.3 Comparing our results with previous work:

	Authors	Models	Accuracy
TOMATO	Siti Zulaikha Muhammad Zaki et al [1]	Mobilenet V2	95.64%
	Mohammed Brahimi et al [2]	Alexnet	98.66%
	Ying Li et al [6]	IBNT	98.67%
	Ramesh Babu Ch et al [7]	Bi-CNN's	91.20%
	Our contribution	Vgg19	96.75%
GRAPE	Jerry Gao et al [3]	Ensemble learning	99%
	Our contribution	Vgg19	99%
WATERMELON	**Our contribution**	Vgg19	89%

Table 6: Comparing our results with previous work:

5. Application :

We developed an application to classify plant leaf diseases using Flutter and TensorFlow Lite. The application helps farmers and hobbyists identify diseases that affect their plants quickly and easily by taking pictures of the leaves with their mobile phones or choosing a picture from the gallery.

5.1 Converting the Model to TensorFlow Lite:

After training the model on plant leaf disease data using TensorFlow, we converted the trained model to TensorFlow Lite (tflite) format to make it suitable for mobile devices. The tflite format is lightweight and fast, making it ideal for mobile apps.

```
[ ]
    model = tf.keras.models.load_model("/content/grapes.h5")

    converter = tf.lite.TFLiteConverter.from_keras_model(model)
    converter.optimizations = [tf.lite.Optimize.DEFAULT]
    tflite_model = converter.convert()

    with open("/content/grapes.tflite", "wb") as f:
        f.write(tflite_model)
```

Fig 28 convertin the model to tf lite

- **Prepare Label Files:** Save the names of the classifications (diseases) in a text file to be used in the app.

5.2. Designing the User Interface (UI):

5.2.1 Welcome Screen:

The app starts with a simple welcome screen that includes a welcome message and a button to go to the main screen.

Fig 29 welcom screen

5.2.2 Main Screen:

- Fruit Selection: Users can select the type of fruit they want to analyze for diseases by tapping on one of the available options: "Watermelon," "Tomato," or "Grape."After selection a simple dialog appears, offering clear choices between camera and gallery options.

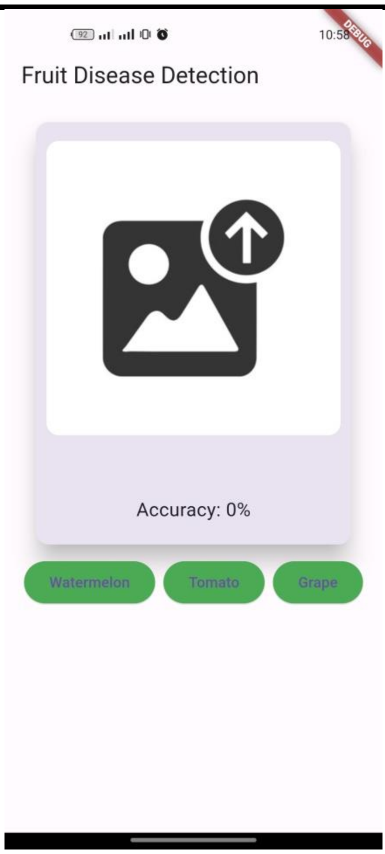

Fig 30 main screen

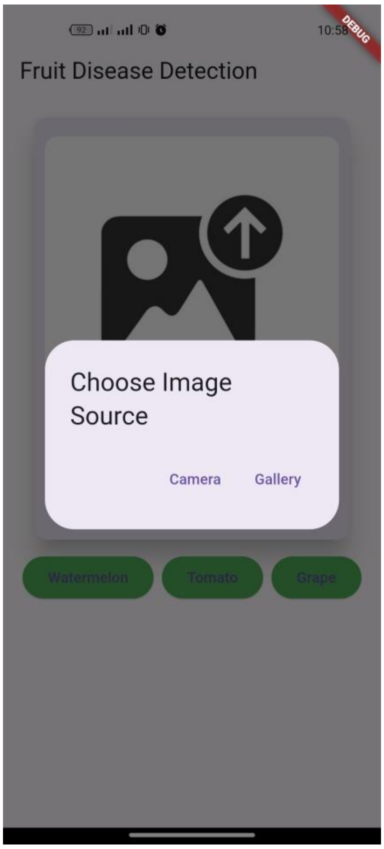

Fig 31 choosing of image

Once an image is uploaded, it is displayed prominently in the central area of the screen. Below the uploaded image, users receive instant feedback on the disease classification. The identified disease and its confidence level are clearly displayed.

Fig 32 the classification screen

6. Used libraries :

6.1 TensorFlow : is an open source machine learning framework that provides a comprehensive, flexible ecosystem of tools, libraries, and community resources for developing and deploying AI-powered applications.

6.2 Keras: is an open-source library that provides a Python interface for artificial neural networks. Initially developed as independent software, Keras was later integrated into the TensorFlow library and expanded to support multiple frameworks like JAX and PyTorch. It is designed to enable fast experimentation with deep neural networks, focusing on user-friendliness, modularity, and extensibility.

6.3 OpenCV (Open Source Computer Vision Library): is an open-source computer vision and machine learning software library. It contains more than 2500 optimized algorithms for various tasks.

6.4 Matplotlib: is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

6.5 Pandas: is an open-source data analysis and manipulation tool that is fast, powerful, flexible, and user-friendly. It is developed on top of the Python programming language.

6.6 NumPy : serves as the cornerstone for scientific computing in Python. This library furnishes a multidimensional array object, alongside an array of derived objects like masked arrays and matrices. Moreover, it boasts a myriad of functions for rapid operations on arrays, spanning mathematical computations, logical operations, shape manipulation, sorting, selection, I/O tasks, discrete Fourier transforms, elementary linear algebra, fundamental statistical operations, random simulation, and beyond

7 Development Tools:

7.1 Flutter: developed by Google, is an open-source framework designed to create visually stunning, natively compiled applications that run seamlessly across multiple platforms, all while being developed from a unified codebase.

Fig 33 Flutter

7.2. TensorFlow Lite: empowers developers to implement on-device machine learning by facilitating the execution of their models on mobile, embedded, and edge devices.

Fig 34 TensorFlow Lite

7.2 Visual Studio Code: is a versatile source-code editor developed by Microsoft. It's built using the Electron Framework, making it compatible across multiple platforms including Windows, Linux, and macOS. VS Code boasts an extensive array of functionalities such as debugging tools, syntax highlighting, intelligent code completion, code snippets, code refactoring utilities, and seamless integration with Git

Fig 35 visual studio code

7.3Android Studio : developed by Google, is an IDE specifically designed for building Android apps. It supports languages like Java, Dart, Kotlin, and C++, and provides essential features such as code completion, debugging, and project management.

Fig 36 android studio

7.4 Kaggel: Is a subsidiary of Google, Kaggle is an online community for data scientists and machine learning engineers. It provides a platform where users can discover datasets for building AI models, share their own datasets, collaborate with peers, and participate in competitions aimed at solving data science challenges and Served as the development environment for executing the code and conducting experiments.

Fig 37 kaggle

8 Conclusion :

In this chapter, we have outlined the comprehensive process of implementing a deep learning model for the classification of plant leaf diseases. From reviewing and preprocessing the datasets to customizing and fine-tuning the pre-trained VGG19 model, we have demonstrated the effectiveness of our approach. The results showed that our model is capable of accurately identifying various plant diseases, underscoring the potential benefits of employing deep learning techniques in agriculture. By converting the trained model to TensorFlow Lite, we were able to create an application that aids farmers and hobbyists in diagnosing plant diseases efficiently using their mobile devices. This advancement represents a crucial step toward enhancing smart farming practices and mitigating crop losses due to diseases.

General Conclusion :

In this thesis, we addressed the topic of plant disease classification using deep learning and computer vision techniques, which are modern technologies that represent a promising future for smart agriculture. In the initial chapters, we reviewed the importance of food security and the challenges it faces due to plant diseases, focusing on their impact on grape, watermelon, and tomato crops.

We then provided a detailed explanation of deep learning and computer vision techniques and the significance of integrating these technologies to improve the accuracy and efficiency of plant disease diagnosis. By reviewing previous work, it became evident that there has been significant progress in this field, although challenges still remain that require innovative solutions.

In the practical part of the thesis, we explained the process of data collection and preparation, the design of the deep model used in leaf disease classification, and the analysis of results which demonstrated high diagnostic effectiveness. We also developed a practical mobile application that facilitates the use of these technologies by farmers and agricultural experts, enhancing their ability to deal with diseases more quickly and efficiently.

Bibliographie

[1] Siti Zulaikha Muhammad Zaki, Mohd Asyraf Zulkifley, Marzuraikah Mohd Stofa, Nor Azwan Mohammed Kamari, Nur Ayuni Mohamed .Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600, UKM Bangi, Selangor, Malaysia.

[2] Mohammed Brahimi, Kamel Boukhalfa & Abdelouahab Moussaoui (2017) Deep Learning for Tomato Diseases: Classification and Symptoms Visualization, Applied Artificial Intelligence, 31:4, 299-315, DOI: 10.1080/08839514.2017.1315516

[3] Gao, J., & Mastan, S. (2020). Grape leaf disease detection and classification using machine learning. International Journal of Engineering and Technical Research, 9(12), 11-16.

[4] Qingtian Zeng1 · Liangwei Niu1 · Shansong Wang1 · Weijian Ni1.Received: 25 June 2022 / Accepted: 28 November 2022 / Published online: 19 December 2022. The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022.

[5] Yu Xia, Mingjing Tang & Wei Tang (2023) Fine-grained Potato Disease Identification Based on Contrastive Convolutional Neural Networks, Applied Artificial Intelligence, 37:1, 2166233, DOI: 10.1080/08839514.2023.2166233

[6] Xianyang Normal University, Xianyang, 712000, China School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an, 710021, China

[7] Intelligent Automation & Soft Computing DOI:10.32604/iasc.2022.017706 BCNN

[8] P. Jiang et al.: Real-Time Detection of Apple Leaf Diseases Using Deep-Learning Approach Based on Improved CNNs n

[9] NATIONAL INSTITUTE OF FOOD AND AGRICULTURE. (N.D.). FOOD & NUTRITION SECURITY. RETRIEVED FROM HTTPS://WWW.NIFA.USDA.GOV/TOPICS/FOOD-NUTRITION-SECURITY

[9] DATAMYTE. (N.D.). THE IMPORTANCE OF FOOD SECURITY. RETRIEVED FROM HTTPS://DATAMYTE.COM/BLOG/FOOD-SECURITY-IMPORTANCE/

[10] FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. (N.D.). RETRIEVED FROM HTTPS://WWW.FAO.ORG/DOCUMENTS/CARD/ES?DETAILS=4ABD5809-F519-4D0B-A3A3-0A4CFCDDD03B%2F

[11] WORLD VISION CANADA. (N.D.). THE BASICS OF FOOD SECURITY. RETRIEVED FROM HTTPS://WWW.WORLDVISION.CA/STORIES/FOOD/THE-BASICS-OF-FOOD-SECURITY

[12] USAID. (N.D.). AGRICULTURE AND FOOD SECURITY. RETRIEVED FROM HTTPS://WWW.USAID.GOV/AGRICULTURE-AND-FOOD-SECURITY

[13] FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. (2022). RETRIEVED FROM HTTPS://WWW.FAO.ORG/FAO-STORIES/ARTICLE/AR/C/1479575/

[14] NATIONAL FOOD INSTITUTE. (N.D.). DO WE HAVE FOOD SECURITY OR DON'T WE?. RETRIEVED FROM HTTPS://NATIONALFOODINSTITUTE.COM.AU/NFI/DO-WE-HAVE-FOOD-SECURITY-OR-DONT-WE/

[15] FRONTIERS IN SUSTAINABLE FOOD SYSTEMS. (2021). RETRIEVED FROM HTTPS://WWW.FRONTIERSIN.ORG/ARTICLES/10.3389/FSUFS.2021.753996/FULL

[16] NATIONAL HORTICULTURE BOARD. (N.D.). RETRIEVED FROM HTTPS://WWW.NHB.GOV.IN/PDF/FRUITS/GRAPE/GRA002.PDF

[17] OKLAHOMA STATE UNIVERSITY EXTENSION. (N.D.). WATERMELON DISEASES. RETRIEVED FROM HTTPS://EXTENSION.OKSTATE.EDU/FACT-SHEETS/WATERMELON-DISEASES.HTM

[18] PYTHON PACKAGE INDEX. (N.D.). TENSORFLOW. RETRIEVED FROM HTTPS://PYPI.ORG/PROJECT/TENSORFLOW/

[19] WIKIPEDIA. (N.D.). KERAS. RETRIEVED FROM HTTPS://EN.WIKIPEDIA.ORG/WIKI/KERAS

[20] OPENCV. (N.D.). OPEN SOURCE COMPUTER VISION LIBRARY. RETRIEVED FROM HTTPS://OPENCV.ORG/

[21] MATPLOTLIB. (N.D.). RETRIEVED FROM HTTPS://MATPLOTLIB.ORG/

[22] PANDAS. (N.D.). RETRIEVED FROM HTTPS://PANDAS.PYDATA.ORG/

[23] NUMPY. (N.D.). RETRIEVED FROM HTTPS://NUMPY.ORG/DOC/STABLE/

[24] FLUTTER. (N.D.). RETRIEVED FROM HTTPS://FLUTTER.DEV/

[25] TENSORFLOW. (N.D.). TENSORFLOW LITE GUIDE. RETRIEVED FROM HTTPS://WWW.TENSORFLOW.ORG/LITE/GUIDE

[26] VISUAL STUDIO CODE. (N.D.). RETRIEVED FROM HTTPS://CODE.VISUALSTUDIO.COM/

[27] ANDROID DEVELOPERS. (N.D.). ANDROID STUDIO. RETRIEVED FROM HTTPS://DEVELOPER.ANDROID.COM/STUDIO?HL=AR

[28] KAGGLE. (N.D.). RETRIEVED FROM HTTPS://WWW.KAGGLE.COM/DISCUSSIONS/GENERAL/328265

[29] GEEKSFORGEEKS. (N.D.). INTRODUCTION TO DEEP LEARNING. RETRIEVED FROM HTTPS://WWW.GEEKSFORGEEKS.ORG/INTRODUCTION-DEEP-LEARNING/

[30]RUN. (N.D.). DEEP LEARNING FOR COMPUTER VISION. RETRIEVED FROM HTTPS://WWW.RUN.AI/GUIDES/DEEP-LEARNING-FOR-COMPUTER-VISION

[31] GEEKSFORGEEKS. (N.D.). DEEP LEARNING TUTORIAL. RETRIEVED FROM HTTPS://WWW.GEEKSFORGEEKS.ORG/DEEP-LEARNING-TUTORIAL/?REF=PREVIOUS_ARTICLE

[32] IBM. (N.D.). AI VS. MACHINE LEARNING VS. DEEP LEARNING VS. NEURAL NETWORKS. RETRIEVED FROM HTTPS://WWW.IBM.COM/THINK/TOPICS/AI-VS-MACHINE-LEARNING-VS-DEEP-LEARNING-VS-NEURAL-NETWORKS

[33] TECHTARGET. (N.D.). DEEP LEARNING (DEEP NEURAL NETWORK). RETRIEVED FROM HTTPS://WWW.TECHTARGET.COM/SEARCHENTERPRISEAI/DEFINITION/DEEP-LEARNING-DEEP-NEURAL-NETWORK

[34] INSIDEBIGDATA. (2023). TRANSFER LEARNING IN COMPUTER VISION. RETRIEVED FROM HTTPS://INSIDEBIGDATA.COM/2023/07/24/TRANSFER-LEARNING-IN-COMPUTER-VISION/

[35]AMAZON WEB SERVICES. (N.D.). WHAT IS COMPUTER VISION?. RETRIEVED FROM HTTPS://AWS.AMAZON.COM/AR/WHAT-IS/COMPUTER-VISION/

[36] ATILA, Ü., UÇAR, M., AKYOL, K., & UÇAR, E. (2020). PLANT LEAF DISEASE CLASSIFICATION USING EFFICIENTNET DEEP LEARNING MODEL. *ECOLOGICAL INFORMATICS*, 61, 101182. https://doi.org/10.1016/j.ecoinf.2020.101182

[37] Noulas, A. (2021). *Tomato Leaf Disease Dataset*. Kaggle. Retrieved June 5, 2024, from

https://www.kaggle.com/datasets/noulam/tomato

[38] Kapadnis, S. (2021). Watermelon Disease Recognition Dataset. Kaggle. Retrieved June 5, 2024, from c

[39]Pushpala, M. (2021). *Grape Disease*. Kaggle. Retrieved June 5, 2024, from https://www.kaggle.com/datasets/pushpalama/grape-disease

[40] Yasun, Ş. (2024, June 24). Lung X-ray image analysis with CNN-based approaches. LinkedIn post. Retrieved from https://www.linkedin.com/posts/%C5%9Fafak-yasun-570022213_lung-x-ray-image-analysis-with-cnn-based-activity-7113882768614305792-cCsp

[41] "WordPress SEO Expert", (2024, June 24). Machine Learning vs Deep Learning. LinkedIn post. Retrieved from https://www.linkedin.com/pulse/machine-learning-vs-deep-certified-ai-wordpress-seo-expert-fsatf

[42] Researcher, M. (2024**).** Deep Learning and Neural Networks: A Comparison. Image and information retrieved from ResearchGate. Available at: https://www.researchgate.net/figure/Deep-learning-and-a-neural-network-Deep-learning-neural-networks-are-similar-to_fig2_355704301

[43]Analytics Vidhya. (2022). A Typical Convolutional Neural Network Diagram. Image retrieved from https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/

[44] BotPenguin. (2024). Recurrent Neural Network. Image and information retrieved from https://botpenguin.com/glossary/recurrent-neural-network.

[45] Lekhuyen. (2024). The Architecture of VGG16. Image and information retrieved from https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484.

[46] Towards Data Science. (2024). Understanding and Visualizing ResNets. Image and information retrieved from https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8.

[47] DataHacker. (2024). An Example of Inception Module. Image and information retrieved from https://datahacker.rs/building-inception-network/.

[48] ResearchGate. (2024). Illustration of the MobileNet Architecture. Image and information retrieved from https://www.researchgate.net/figure/Illustration-of-the-MobileNet-architecture-A-The-overall-MobileNet-architecture-and_fig3_340470168.

[50] **K. Sikhonja, J. Maraghi, & I. Ndije. (2021).** Analysis and Assessment of Resource Use Efficiency in Food Production Systems**. *Frontiers in Sustainable Food Systems*. Available at:** **https://www.frontiersin.org/articles/10.3389/fsufs.2021.753996/full**

Abstract

Plant diseases significantly impact agricultural productivity and global food security. This study aims to develop a precise model for classifying plant leaf diseases using fine-grained techniques involving deep learning and computer vision. Focusing on three different datasets—tomato, grape, and watermelon leaves, which are among the most consumed and produced fruits in Algeria—we employed the VGG19 convolutional neural network (CNN) to analyze and accurately classify images of diseased and healthy leaves. The model demonstrated high accuracy and robustness, which were further validated through various performance metrics. A practical application was also developed to facilitate real-time disease diagnosis, aiding farmers in effective crop management and enhancing food security.

Keywords: Plant disease classification, deep learning, VGG19, computer vision, food security, mobile application, precision agriculture

ملخص:

الأمراض النباتية تؤثر بشكل كبير على إنتاجية الزراعة والأمن الغذائي العالمي. تهدف هذه الدراسة إلى تطوير نموذج دقيق لتصنيف أمراض أوراق النباتات باستخدام تقنيات متقدمة تشمل التعلم العميق والرؤية الحاسوبية. وبالتركيز على ثلاث مجموعات بيانات مختلفة - أوراق الطماطم، العنب، والبطيخ، التي تعد من أكثر الفواكه استهلاكاً وإنتاجاً في الجزائر - استخدمنا شبكة التلافيف العصبية VGG19 (CNN) لتحليل وتصنيف صور الأوراق المريضة والصحية بدقة. أظهر النموذج دقة عالية وقوة، مما تم التحقق منه من خلال مقاييس أداء متنوعة. كما تم تطوير تطبيق عملي لتسهيل تشخيص الأمراض في الوقت الحقيقي، مما يساعد المزارعين في إدارة المحاصيل بشكل فعال وتعزيز الأمن الغذائي.

الكلمات المفتاحية: تصنيف أمراض النبات، التعلم العميق، VGG19، الرؤية الحاسوبية، الأمن الغذائي، تطبيق المحمول، الزراعة الدقيقة.

Résumé

Les maladies des plantes ont un impact significatif sur la productivité agricole et la sécurité alimentaire mondiale. Cette étude vise à développer un modèle précis pour classifier les maladies des feuilles des plantes en utilisant des techniques avancées impliquant l'apprentissage profond et la vision par ordinateur. En se concentrant sur trois ensembles de données différents - feuilles de tomate, de raisin et de pastèque, qui sont parmi les fruits les plus consommés et produits en

Algérie - nous avons employé le réseau de neurones convolutifs (CNN) VGG19 pour analyser et classifier avec précision les images des feuilles malades et saines. Le modèle a démontré une haute précision et robustesse, ce qui a été validé par divers métriques de performance. Une application pratique a également été développée pour faciliter le diagnostic des maladies en temps réel, aidant les agriculteurs à gérer efficacement leurs cultures et à renforcer la sécurité alimentaire.

Mots-clés : Classification des maladies des plantes, apprentissage profond, VGG19, vision par ordinateur, sécurité alimentaire, application mobile, agriculture de précision.