Plant Care and Disease Detection Using Pattern Recognition

Piere John Michael
dept. Computer Science
Misr University for Science and Technology
Giza,Egypt
94065@must.edu.eg

Yousab Mena Gad
dept. Computer Science
Misr University for Science and Technology
Giza,Egypt
94095@must.edu.eg

Mohab Mohamed Abdelsadek
dept. Computer Science
Misr University for Science and Technology
Giza,Egypt
94267@must.edu.eg

Kerolos Ashraf Ateya
dept. Information Systems
Misr University for Science and Technology
Sohag,Egypt
94189@must.edu.eg

Abdelrahman Hamdy Ahmed
dept. Information Systems
Misr University for Science and Technology
Giza,Egypt
89676@must.edu.eg

Khaled Abd El-Salam Ali
Head dept. Inforamtion Systems
Misr University for Science and Technology
Giza,Egypt
khaled.abdelsalam@must.edu.eg

AL. Shaimaa Bahaa Bahaa
dept. Computer Science
Misr University for Science and Technology
Giza,Egypt
shimaa.nasr@must.edu.eg

TA. Ahmed Abdallah Mahmoud
dept. Computer Science
Misr University for Science and Technology
Giza,Egypt
ahmed.asoliman@must.edu.eg

***Abstract*—The well-being of plants is paramount for maintaining a healthy environment and sustaining life on Earth. In this study, we developed a mobile application called "Plant Care and Disease Detection Using Pattern Recognition" to address these challenges. The application uses a deep learning model based on Convolutional Neural Networks (CNN) to analyze plant leaf images and determine the presence of diseases with 97% accuracy and an F1-score of 98%. The app connects to a pre-trained model with 37 classes, including 13 different plant diseases and healthy plants, totaling 125,319 images. Users can scan a plant leaf with their mobile device's camera, and the model provides a diagnosis or confirms that the plant is healthy. The app schedules watering and fertilizing tasks based on specific plant needs, with notifications to remind users of care times. Additionally, the app uses Firebase for authentication and Fire store for database management, allowing users to sign up, create profiles, and recover passwords if forgotten. This application improves plant care by offering a reliable solution for disease detection and care scheduling. The broader implications suggest that this technology can support sustainable agricultural practices and contribute to global environmental efforts by reducing the need for chemical treatments through early detection. Ultimately, the "Plant Care and Disease Detection Using Pattern Recognition" app showcases how deep learning can transform plant care, fostering a healthier plant ecosystem and benefiting individual gardeners, the agricultural sector, and the broader environment.**

Keywords—Plant disease detection, Convolutional neural networks (CNN), Flutter application development, Firebase, Classification, Deep learning

I. Introduction

Artificial Intelligence (AI) is a transformative branch of computer science concerned with creating systems that can perform tasks which typically require human intelligence. These tasks include learning, reasoning, problem-solving, perception, and language understanding. At its core, AI is about the development of algorithms that enable machines to perform complex tasks, such as recognizing speech, making decisions, and identifying patterns. The term “AI” encompasses a broad range of technologies, including machine learning, deep learning, and natural language processing (NLP). AI systems are powered by data and algorithms, and they learn from patterns or features in the data. Machine learning, a subset of AI, uses algorithms trained on data to create models that can make predictions or perform tasks. For example, machine learning models can recommend TV shows, identify the fastest route to a destination, or translate text from one language to another. AI’s impact is widespread, affecting many aspects of daily life and work. It has the potential to drive innovation across various sectors, including healthcare, finance, education, and more. As AI continues to evolve, it promises to offer even more sophisticated capabilities that could reshape the way we live and work [1] - [4]. Plant care, a practice as ancient as agriculture itself, has evolved from rudimentary techniques to sophisticated methods that leverage data and automation. The essence of nurturing plants lies in understanding their unique watering and fertilizing rhythms. Traditional methods often lead to suboptimal care—either through excess or deficiency. However, AI-driven solutions now enable precise control over these critical factors, ensuring plants receive the exact care they need at the optimal time [5] – [8]. At the forefront of AI’s agricultural applications is the use of Convolutional Neural Networks (CNNs) for leaf disease detection. CNNs, a class of deep learning models, excel in image recognition tasks, making them ideal for identifying patterns indicative of plant diseases. By training on vast datasets of plant imagery, CNNs can detect and diagnose health issues with unprecedented speed and accuracy, far surpassing human capabilities [9]. The integration of AI in plant care and disease detection culminates in user-friendly mobile applications developed with platforms like Flutter. These applications democratize access to advanced agricultural technology, allowing users to effortlessly scan plant leaves, receive instant diagnostics, and manage care schedules [10] . The seamless fusion of AI-driven insights with intuitive app interfaces empowers even novice gardeners to provide expert-level care to their plants.

The research consists of five parts. The first part consists of Introduction. The second part consists of related works. The third part consists of the proposed system. The fourth part consists of results and discussion. The fifth part consists of the conclusion.

II. Related Work

1. *Grape Leaf Disease Detection Using Deep Learning, 2023*

The research paper discusses the benefits of using deep learning for identifying diseases in grape leaves. It highlights the capability of deep learning to facilitate accurate and timely illness diagnosis, process massive volumes of picture data, and improve over time with additional data. The paper emphasizes that applying deep learning to identify diseases on grape leaves could lead to improvements in vineyard management, decreased crop losses, and support for sustainable agriculture. The study reviews common grape diseases, such as black rot and black measles, and the challenges associated with manual disease diagnosis. It also discusses the methodology and implementation of deep learning for grape leaf disease detection, utilizing a Grape Leaf Dystrophy Dataset. The paper presents the testing and results of the deep learning model, including measures such as accuracy, recall, precision, and F-score. The conclusion highlights the potential of using deep learning for disease detection in grape leaves and outlines future research directions, including the need for large-scale datasets, improved model interpretability, and robustness to environmental factors. The potential for improved disease detection in grape leaves is seen as promising for the future of the agricultural industry. Research objectives and methodology. Objectives: Research grape leaf diseases in depth. In order to use deep learning for disease classification in grape leaves. Create an automated system that can determine whether or not grape leaves are healthy. Use metrics like accuracy, recall, precision, and f1 score to assess how well the deep learning implementation is doing. Methodology: The primary goal of this study is to develop a classification system for grape leaf recognition based on a deep learning algorithm. Filtering and scaling the picture to pixels are two examples of preprocessing procedures. After the data has been cleaned and prepared, it will be split into training and testing sets. The plan is to use a deep learning technique to train the model, and then to put it to the test on a separate dataset. A further step involves evaluating the algorithm's effectiveness in practice. Research process flowchart for symptom-based grape leaf disease diagnosis. Black rot, black measles, health, and many more are among the numerous common illnesses. Discoloration, stains, lesions, deformations, and powdery growth on the leaves are all symptoms unique to each illness. Findings and conclusions**.** Deep learning can revolutionize viticulture by accurately identifying grape leaf diseases with 90.44% accuracy. It enables early diagnosis and efficient disease management. Challenges: Grape leaf diseases are complex, but convolutional neural networks (CNNs) excel in distinguishing healthy and sick leaves. Benefits: Deep learning captures subtle patterns from massive image data, improving accuracy over time. Obstacles: Robust models require diverse training datasets. Future: Ongoing research promises advanced detection models, benefiting vineyard management and sustainable agriculture [11]-[15].

2. *MobileNet Based Apple Leaf Diseases Identification, 2020*

The paper proposes a low-cost, stable, and high precision method for identifying common apple leaf diseases, such as Alternaria leaf blotch and rust, using the MobileNet model. This method is cost-effective and suitable for deployment on mobile devices. The paper highlights the instability and time-consuming nature of current apple leaf disease inspection methods by experienced experts, leading to the proposal of a method that can offer stable identification results, be deployed on mobile devices, and achieve high precision. Experiments were conducted using apple disease datasets collected in Shaanxi Province, China, and the proposed MobileNet model's effectiveness was demonstrated. The paper also compares the efficiency and precision of the MobileNet model with other deep learning models such as ResNet152 and InceptionV3. The results show that the MobileNet model is the most efficient, with an average handling time of 0.22 seconds per image, while maintaining high accuracy. Research objectives and methodology. Objectives: Develop a low-cost apple leaf disease identification method. Improve stability of apple leaf disease inspection. Achieve high precision for disease identification. Compare the proposed method with existing models. Methodology: Develop a deep learning-based system for apple leaf disease identification using the MobileNet model. This method aims to be: Low-cost and mobile-friendly (MobileNet) [16]-[18].

Stable and less prone to human error High precision in disease identification. Dataset Construction: A crucial aspect for achieving stable identification results. Agriculture experts from the Chinese Academy of Agricultural Sciences collected data from Shaanxi Province, China. The data set focuses on two main apple leaf diseases: Alternaria leaf blotch and rust. A total of 334 images were collected, categorized as 164 Alternaria leaf blotch and 170 rust images. The selection criteria for image collection considered factors like: Shape and color of the leaves, Number of diseases per leaf Model Selection and Comparison [19]: The paper explores three different deep learning models: MobileNet (proposed solution) - focuses on efficiency and mobile deployment .ResNet152 - known for high precision in image recognition tasks .InceptionV3 - another high-performance image recognition model .The research will compare the performance of these models for apple leaf disease identification in terms of: Accuracy/Precision .Efficiency (processing speed) .Findings and conclusions. The experiment investigated the accuracy of three deep learning models for apple leaf disease identification: MobileNet, InceptionV3, and ResNet152. The accuracies are MobileNet is73.50%. InceptionV3 is 75.59%, ResNet152 is 77.65%. The paper demonstrates the trade-off between efficiency and accuracy. While MobileNet offers the fastest processing speed, it achieves slightly lower accuracy compared to more complex models. This finding suggests that MobileNet presents a compelling option for real-time applications on mobile devices where speed is a priority, while more complex models might be preferred in scenarios demanding the highest possible accuracy [20].

TABLE 1. COMPAING BETWEEN PAPERS.

Title	Methodology	Advantages	disadvantages
Grape Leaf Disease Detection Using Deep Learning	• Utilizes a deep learning algorithm, likely a Convolutional Neural Network (CNN). • Employs a Grape Leaf Dystrophy Dataset for training and testing. • Evaluates performance using metrics like accuracy, recall, precision, and F1-score.	• High Accuracy (90.44%): Effective in distinguishing healthy and diseased leaves. • Real-Time Detection: Enables swift identification for timely intervention. • Feature Extraction: Automatically extracts relevant features from images for accurate diagnosis.	• Limited to Specific Plant Type: Model is designed for grape leaves only. • Diverse Training Data Needed: Requires a more diverse dataset for robustness. • Loss Value (27.48%) suggests room for improvement in reducing prediction errors.
MobileNet Based Apple Leaf Diseases Identification	• Develops a system using the MobileNet model, known for efficiency on mobile devices. • Employs a dataset of apple leaf images collected in China, focusing on Alternaria leaf blotch and rust diseases. • Compares MobileNet with ResNet152 and InceptionV3 models for efficiency (processing speed) and accuracy.	• Reduced workload for experts: Automates disease detection, freeing up expert time. • Low-cost and mobile-friendly: MobileNet model is suitable for deployment on mobile devices in the field.	• Limited dataset: Might not encompass all apple leaf disease variations, impacting accuracy for uncommon diseases. • Limited to Specific Plant Type: Model is designed for apple leaves only. • Accuracy trade-off: MobileNet prioritizes speed with a 73.50% accuracy, which might be lower than desired for some

III. PROPOSED SYSTEM

In the ever-growing realm of mobile applications, this project introduces "Plant Care and Disease Detection Using Pattern Recognition" a user-friendly and intelligent mobile application designed to empower plant enthusiasts and cultivators of all experience levels. This application leverages the power of deep learning, specifically a Convolutional Neural Network (CNN) model, to provide real-time plant health assessments directly on a user's mobile device. By integrating disease detection with personalized plant care schedules, this application aims to bridge the gap between knowledge and action, fostering a more informed and successful approach to plant care. The core functionality revolves around image recognition utilizing the trained CNN model, which has been meticulously trained on a comprehensive dataset encompassing 37 distinct plant diseases and healthy plant variations. With an impressive 97% accuracy, the model analyzes the captured leaf images and promptly delivers a diagnosis, identifying diseases by name or confirming a plant's health. Beyond disease detection, the app supports a holistic plant care routine. Users can create personalized plant profiles and link them to specific watering and fertilizing schedules. This feature, along with automated task reminders and a user account system for managing profiles, ensures each plant receives the tailored care it needs to thrive. In essence, "Plant Care and Disease Detection" represents a comprehensive and user-centric mobile application that merges cutting-edge deep learning technology with practical plant care guidance. By offering a seamless blend of disease identification, personalized plant profiles, and automated task reminders, this application empowers users to cultivate a flourishing and healthy home garden.

1. *System architecture*

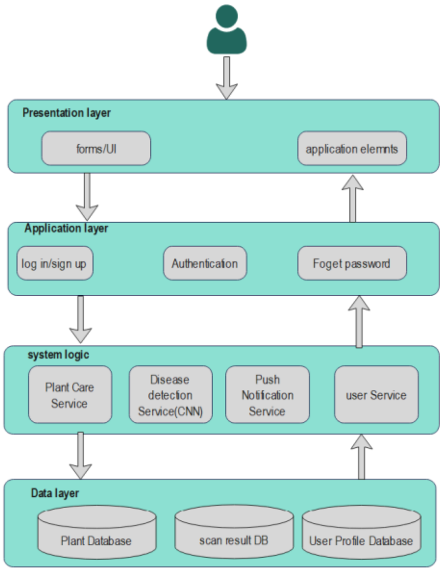


Fig.1. System architecture.

1.1. Presentation Layer

- This layer represents the user interface (UI) elements that users interact with directly with their mobile devices.
- It is represented by elements like buttons for capturing a plant image, displaying the captured image, and showing the disease diagnosis or confirmation of plant health.

1.2. Application Layer

- Manages user interface.
- Components:
  - Log In/Sign Up: Handles user authentication and registration.
  - Authentication: Verifies user credentials during login.
  - Forget Password: Deals with password recovery.

1.3. System Logic Layer

- This layer contains the business logic or rules that govern the application's behavior. It interacts with the application layer and the data layer to process requests and manage data.
- it involves:
  - Implementing the logic for creating plant profiles.
  - Associating specific watering/fertilizing schedules with different plant types based on the database.
  - Sending notifications for watering/fertilizing tasks.
  - Managing plant, including adding new plants, storing watering/fertilizing schedules, and retrieving this information.
  - Receiving the diagnosis results from the API.
  -

1.4. Data Layer

- This layer is responsible for storing and managing the application's data. It interacts with the business logic layer to provide and update data as needed.
- Represented by the Firebase Fire store database. The database might store:
    - User information (profiles, credentials)
    - Plant data (plant types, associated watering/fertilizing schedules)

2. Algorithms used

2.1 CNN

Convolutional neural networks, also known as CNNs [21]-[26], are a specific type of neural networks that are generally composed of the following layers:

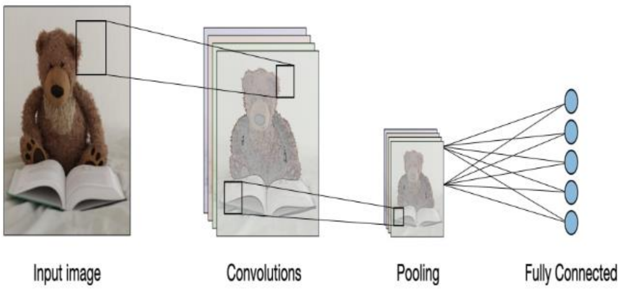
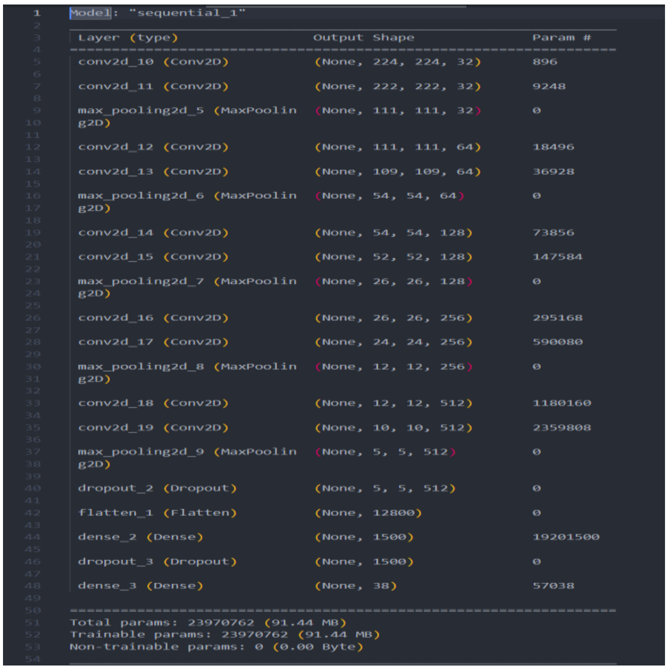


Fig.2. CNN architecture.

Types of layers:

1. Convolution layer (CONV): The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size F and stride S. The resulting output O is called feature map or activation map.
2. Pooling (POOL): The pooling layer (POOL) is a down sampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.
3. Fully Connected (FC): The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores [35].

The created deep CNN model that powers our "Plant Care and Disease Detection" application. This meticulously crafted architecture combines layers designed to process and extract features at varying levels of complexity, enabling accurate disease identification and plant health assessment.



Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 224, 224, 32)	896
conv2d_11 (Conv2D)	(None, 222, 222, 32)	9248
max_pooling2d_5 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_12 (Conv2D)	(None, 111, 111, 64)	18496
conv2d_13 (Conv2D)	(None, 109, 109, 64)	36928
max_pooling2d_6 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_14 (Conv2D)	(None, 54, 54, 128)	73856
conv2d_15 (Conv2D)	(None, 52, 52, 128)	147584
max_pooling2d_7 (MaxPooling2D)	(None, 26, 26, 128)	0
conv2d_16 (Conv2D)	(None, 26, 26, 256)	295168
conv2d_17 (Conv2D)	(None, 24, 24, 256)	590080
max_pooling2d_8 (MaxPooling2D)	(None, 12, 12, 256)	0
conv2d_18 (Conv2D)	(None, 12, 12, 512)	1180160
conv2d_19 (Conv2D)	(None, 10, 10, 512)	2359808
max_pooling2d_9 (MaxPooling2D)	(None, 5, 5, 512)	0
dropout_2 (Dropout)	(None, 5, 5, 512)	0
flatten_1 (Flatten)	(None, 12800)	0
dense_2 (Dense)	(None, 1500)	19201500
dropout_3 (Dropout)	(None, 1500)	0
dense_3 (Dense)	(None, 38)	57038

Total params: 23970762 (91.44 MB)
Trainable params: 23970762 (91.44 MB)
Non-trainable params: 0 (0.00 Byte)

Fig.3. The structure of the created CNN model.

٢.2. Firebase authentication

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices [27]-[29]. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more [30].

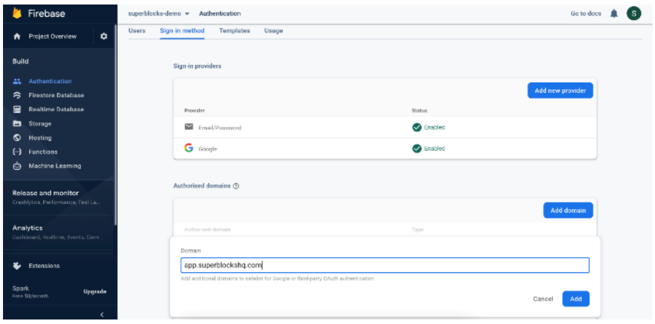


Fig.4. Authenticate using Firebase Authentication.

3. Dataset

We collected more than labelled 96k images of healthy and infected plant leaves for training the CNN model from aa source such as Kaggle [8]. Many images in our dataset are in their natural environments because object detection is highly dependent on contextual information. Our dataset is divided into two parts: training and validation. A dataset of 125,319 images of the leaves of 13 different plant species was used to train and validate the proposed Deep CNN model. There are a total of 37 disease classes, and each one represents either a healthy plant or one that has been infected.

Fig.5. Random sample of the dataset.

4. *Implemented system functions*

4.1. CNN

The CNN model is implemented using Keras development environment [9]. Keras is an open-source neural network library written in Python, which uses TensorFlow [10] as a back-end engine. Keras libraries running on top of TensorFlow make it relatively easy for developers to build and test deep learning models written in Python. For instance, we used the keras. preprocessing. image. ImageDataGenerator library to augment some images in our dataset via several geometric transformations; therefore, our model would never see twice the same image. This helps to avoid overfitting and helps the model generalize better. The training images must have the same size before feeding them as input to the model. Our model was trained with colored (RGB) images with resized dimensions of 224 × 224 pixels. The model is connected to the application through an API. Our plant disease detector model is considered a multi-class classification problem, where it classifies the input image as belonging to one or more of the 37 disease classes. And as shown in figure 4-5 the accuracy of the model reached 99% for the training and 97% for validation, which means that our dataset and the fine-tuned parameters were a good fit for the model.

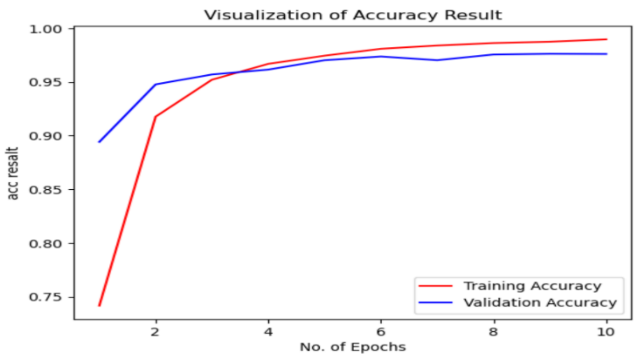


Fig.6. Accuracy visualization.

4.2. Profile Edits

Users can sign up and create their profiles within the app. Profile information includes name, bio, profile picture and background, phone number, and password. Users can edit their profile details as needed. Personalized profiles enhance the user experience and allow for tailored interactions.

4.3. Plant Scanning

Users can capture plant leaf images using their device's camera. Alternatively, they can select existing images from their gallery. The deep learning model (CNN) processes these images to determine whether the plant is healthy or affected by a disease. This feature empowers users to quickly assess their plant's condition and take necessary actions.

4.4. Adding plant to list

Users can create a personalized list of plants within the app. The app then automatically generates watering and fertilizing schedules based on the specific plant's needs. These schedules are stored in the database and associated with each plant. Users can view their plant list on the home page, where tasks (watering, fertilizing) are displayed as actionable items.

4.5. Task Notification

The application provides timely notifications to remind users of upcoming tasks. For example, if it's time to water a specific plant, the app sends a notification. Users can customize notification preferences (e.g., frequency, time of day). These reminders ensure that plant care tasks are not overlooked.it is implemented using this flutter extension "flutter_local_notifications" is a Flutter plugin that allows developers to create and manage local notifications in Flutter apps. Local notifications are messages displayed to users at specified times or events, set by the app rather than a server.

5. Tools

5.1. Flutter

Flutter is a powerful and versatile open-source framework developed by Google for building beautiful, natively compiled applications across multiple platforms from a single codebase [11].

5.1.2 Firebase

Firebase is a comprehensive app development platform provided by Google. It enables developers to build, manage, and grow their apps across various platforms, including mobile (iOS and Android), web, and more [12] .

5.3. Python

Python is a popular high-level, interpreted, interactive, and object-oriented programming language. It was created by Guido van Rossum and released in 1991. Python is designed to be highly readable and uses English keywords frequently, making it accessible to beginners. It has fewer syntactical constructions compared to other languages [13].

5.3.1 Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. Keras provides a convenient way to define and train almost any kind of deep learning model [9].

5.3.2 TensorFlow

TensorFlow is a popular open-source framework for machine learning and deep learning. Developed by the Google Brain Team, it provides tools and libraries for creating, training, and deploying machine learning models. TensorFlow is used for building and training deep neural networks. It supports tasks like image recognition, natural language processing, and more[10].

5.3.3 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools [14].

IV. Results and Discusion

1. Results of the Developed Application

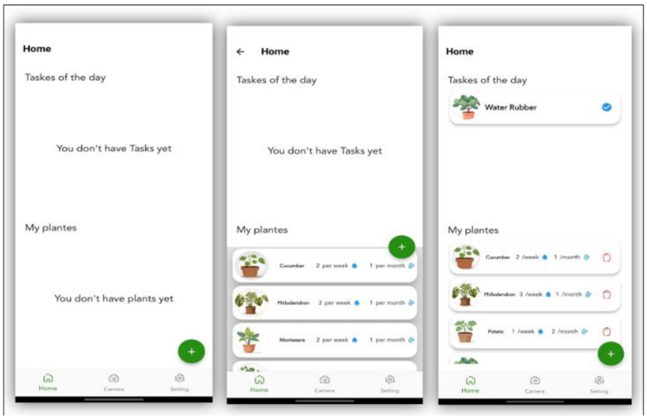


Fig.7. Home page , adding plant and tasks of the day.

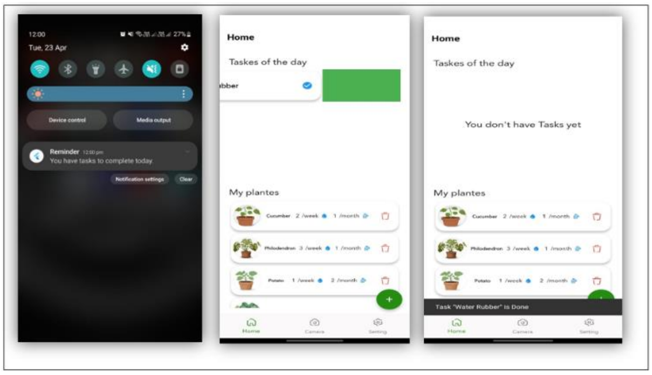


Fig.8. Notification and finishing tasks.

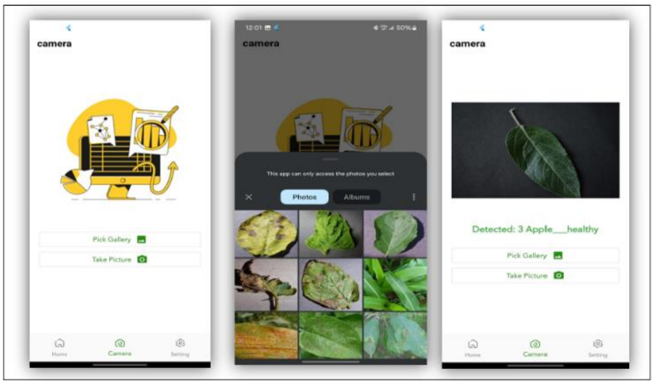


Fig.9. Disease detection.

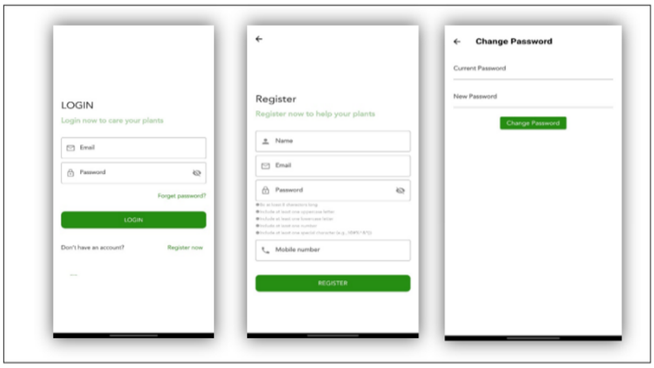


Fig.10. Login & registration.

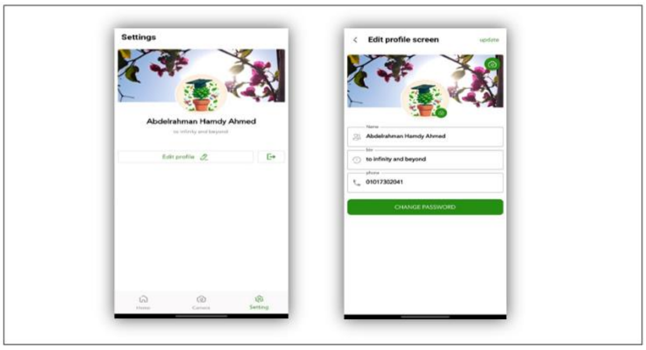


Fig.11. Profile & Edite profile.

2. CNN Model Evaluation

For classification accuracy, we observed that our system delivers good results in natural conditions even when the plant images are captured from different distances from the camera, orientations, and illumination conditions. Figure 5-4 shows some samples of the successful recognition of varying plant leaf diseases.

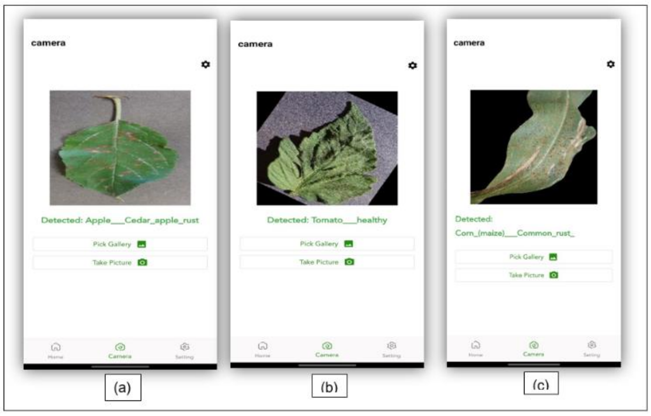


Fig.12. Disease detection sample.

For testing purposes, we can use various measures for testing, and they are: The precision ratio describes the performance of our model at predicting the positive class. It is calculated by dividing the number of true positives by the sum of the true positives and false positives, as follows:

$$\text{Precision} = \frac{TruePositive}{Truepositive + FalsePositive} \quad (1)$$

The recall ratio is calculated as the ratio of the number of true positives divided by the sum of the true positives and the false negatives, measures the proportion of true positives identified out of all actual positive cases, indicating the model's ability to correctly identify relevant instances, as follows:

$$\text{Recall} = \frac{TruePositive}{Truepositive + FalseNegatives} \qquad (2)$$

F1-score ratio is calculated by a weighted average of both precision and recall, is the harmonic mean of precision and recall, providing a single metric to evaluate the balance between false positives and false negatives, False positives occur when a model incorrectly predicts a positive outcome when it should have been negative, like diagnosing a healthy plant as diseased. False negatives happen when a model incorrectly predicts a negative outcome when it should have been positive, such as missing a disease in a plant that is actually affected, as follows:

$$\text{F-measure} = 2 * \frac{Precision}{Precision + Recall} \qquad (3)$$

	precision	recall	f1-score	support
Apple___Apple_scab	0.97	0.93	0.95	721
Apple___Black_rot	0.98	0.99	0.99	684
Apple___Cedar_apple_rust	0.90	0.97	0.93	549
Apple___healthy	0.98	0.99	0.98	1047
Blueberry___healthy	0.97	0.97	0.97	937
Cherry_(including_sour)___Powdery_mildew	0.99	0.99	0.99	737
Cherry_(including_sour)___healthy	0.97	0.98	0.97	764
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0.95	0.92	0.93	584
Corn_(maize)___Common_rust_	0.99	0.99	0.99	838
Corn_(maize)___Northern_Leaf_Blight	0.96	0.98	0.97	777
Corn_(maize)___healthy	0.99	1.00	1.00	814
Grape___Black_rot	0.98	0.98	0.98	844
Grape___Esca_(Black_Measles)	1.00	0.99	1.00	895
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	0.99	0.99	753
Grape___healthy	0.98	0.99	0.99	602
Orange___Haunglongbing_(Citrus_greening)	1.00	1.00	1.00	2156
Peach___Bacterial_spot	0.98	0.98	0.98	1149
Peach___healthy	0.99	0.99	0.99	635
Pepper,_bell___Bacterial_spot	0.94	0.97	0.96	793
Pepper,_bell___healthy	0.98	0.98	0.98	942
Potato___Early_blight	1.00	0.99	0.99	785
Potato___Late_blight	0.97	0.98	0.98	785
Potato___healthy	0.99	0.96	0.97	502
Raspberry___healthy	0.99	0.98	0.98	569
Soybean___healthy	0.98	0.98	0.98	505
Squash___Powdery_mildew	0.96	0.99	0.98	434
Strawberry___Leaf_scorch	0.99	0.98	0.99	777
Strawberry___healthy	0.99	0.98	0.99	612
Tomato___Bacterial_spot	0.96	0.98	0.97	425
Tomato___Early_blight	0.88	0.97	0.92	782
Tomato___Late_blight	0.99	0.86	0.92	1058
Tomato___Leaf_Mold	0.99	0.96	0.97	759
Tomato___Septoria_leaf_spot	0.95	0.87	0.91	436
Tomato___Spider_mites Two-spotted_spider_mite	0.98	0.96	0.97	435
Tomato___Target_Spot	0.96	0.99	0.97	879
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1.00	1.00	1.00	2098
Tomato___healthy	0.99	0.98	0.99	481
not_plant_leaf(please try agin)	0.88	0.95	0.92	321
accuracy			0.98	29864
macro avg	0.97	0.97	0.97	29864
weighted avg	0.98	0.98	0.98	29864

Fig.13. The Precision vs. Recall vs f1-score Values of the CNN Model for All Disease Classes.

Figure 14 shows the classification accuracy and prediction time across the 37 disease classes. The CNN model achieved an overall average classification accuracy of 97.59%.This is evident that users can diagnose any plant disease in their agricultural fields or home plants using a handy mobile app in a fast way .

Evaluating Model

```
[40]
train_loss, train_acc = cnn.evaluate(training_set)
print('Training accuracy:', train_acc)

2983/2983 [==============================] - 865s 290ms/step - loss: 0.0138 - accuracy: 0.9957
Training accuracy: 0.9956838488578796

[41]
val_loss, val_acc = cnn.evaluate(validation_set)
print('Validation accuracy:', val_acc)

934/934 [==============================] - 269s 288ms/step - loss: 0.1050 - accuracy: 0.9760
Validation accuracy: 0.9759576916694641
```

Fig.14. Evaluating the model.

3. Discission

3.1. Introduction

The "Plant Care and Disease Detection Using Pattern Recognition" mobile application was designed to offer a comprehensive solution for plant health management, leveraging deep learning and a database-driven plant care system. This app combines plant disease detection with plant care scheduling, allowing users to scan plant leaves for disease identification and manage watering and fertilizing tasks through a single interface**.**

3.2. Deep Learning Model and Performance

The core feature of the app is the disease detection capability, powered by a Convolutional Neural Network (CNN) model. The model was trained and validated on a dataset comprising 37 classes, representing 13 different plant diseases and healthy plant leaves. With an accuracy rate of 97% and an F1-score of 98%, the app provides a high level of reliability in disease detection. This allows users to quickly determine whether their plant is healthy or identify the specific disease affecting it.

3.3. Features and User Experience

Beyond disease detection, the app includes plant care features to help users maintain their plants. Users can add plants to their personal list, and the app provides automated scheduling for watering and fertilizing based on the plant type. These tasks are displayed on the homepage, and the app sends notifications to remind users when it's time to water or fertilize their plants.

The app also includes user profile management, allowing users to sign up, edit their profile information (such as name, bio, phone number, password, profile picture, and background), and retrieve their profile in case they forget their password.

3.4. Comparison with Related Work

To assess the relative performance and innovation of the "Plant Care and Disease Detection" app, it is useful to compare it with two related works: "MobileNet-Based Apple

Leaf Diseases Identification" and "Grape Leaf Disease Detection Using Deep Learning."

3.4.1. MobileNet-Based Apple Leaf Diseases Identification

This work focuses on apple leaf disease detection, using the MobileNet architecture with an accuracy rate of 73.5%. Compared to the "Plant Care and Disease Detection" app's CNN model with 97% accuracy, the MobileNet-based approach demonstrates lower performance. This suggests that the deep learning architecture used in the new app may offer a more robust solution for plant disease identification. Additionally, the scope of the "Plant Care and Disease Detection" app is broader, covering multiple plant species and diseases, whereas the MobileNet-based work is specific to apple leaves.

3.4.2. Grape Leaf Disease Detection Using Deep Learning

This work uses a CNN algorithm for grape leaf disease detection, achieving a 90.44% accuracy rate. While this is higher than the MobileNet-based approach, it still falls short of the "Plant Care and Disease Detection" app's 97% accuracy. The broader scope of the new app, covering a variety of plants and diseases, along with additional plant care features, makes it more versatile than the grape leaf-focused study.

TABLE 2. ACCURACY COMPARISON WITH RELATED WORKS.

System	Accuracy
MobileNet-Based Apple Leaf Diseases Identification	73.5 %
Grape Leaf Disease Detection Using Deep Learning	90.44 %
Proposed System	**97 %**

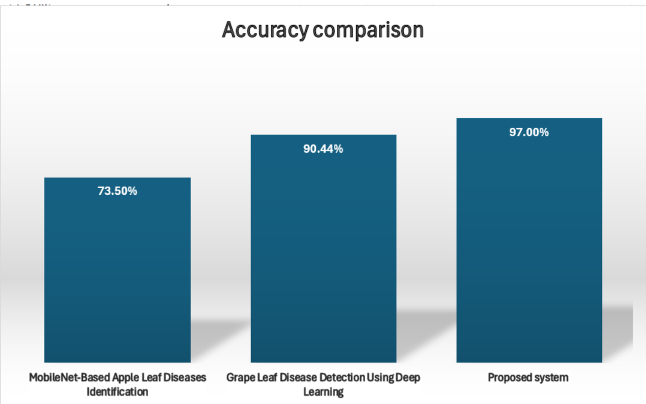


Fig.15. Accuracy comparison chart with related works.

V. CONCLUSION

The increasing neglect of plant health and the difficulty in identifying plant diseases have created a need for innovative solutions to support plant owners in providing proper care. The "Plant Care and Disease Detection Using Pattern Recognition" mobile application addresses these challenges by utilizing deep learning technology to detect plant diseases and by providing a comprehensive plant care system. The primary problems addressed in this project were plant health neglect due to busy schedules, limited botanical knowledge, difficulty in identifying plant diseases, and a lack of guidance for proper plant care. The proposed solution uses a Convolutional Neural Network (CNN) to detect plant diseases with 97% accuracy, allowing users to scan plant leaves and get immediate feedback on their health. The app's comprehensive care system provides personalized reminders for watering and fertilizing, reducing plant health neglect by promoting consistent care routines. Key contributions of this project include the integration of deep learning with plant care, using a dataset with 37 classes, totaling 125,319 images. The application, developed with Flutter, uses Firebase for user management and authentication, enabling users to sign up, create profiles, and manage their plant care schedules. The app's high accuracy in disease detection supports sustainable agricultural practices by reducing the need for chemical treatments through early detection. Overall, the "Plant Care and Disease Detection Using Pattern Recognition" application demonstrates the potential for deep learning and mobile technology to transform plant care practices, offering a comprehensive and practical solution for plant owners. The positive results suggest that this approach can contribute to healthier plant ecosystems and support sustainable practices in the long term.

VI. ACKNOWLEGMENT

We extend our heartfelt thanks to the Dean of our College of Computer and Artificial Intelligence Technology, Dr. Rania El-Gohary, and our supervising professor, Dr. Khalid Abd El-Salam whom we sought his help and professional advice from the beginning of this book to the end. Our gratitude also goes to AL. Shaimaa Bahaa and TA. Ahmed Abdallah for their guidance and support.

REFERENCES

[1] What Is Artificial Intelligence? Definition, Uses, and Types.[Online]. Available: What Is Artificial Intelligence? Definition, Uses, and Types | Coursera

[2] Guide to Houseplant Care Scheduling: Watering, Pruning, Repotting and More. [Online]. Available: Guide to Houseplant Care Scheduling: Watering, Pruning, Repotting and More (plantcareforbeginners.com)

[3] Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. [Online]. Available: Electronics | Free Full-Text | Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach (mdpi.com)

[4] Grape Leaf Disease Detection Using Deep Learning. [Online]. Available:https://www.jsrtjournal.com/index.php/JSRT/article/view/32

[5] MobileNet Based Apple Leaf Diseases Identification. [Online]. Available: https://link.springer.com/article/10.1007/s11036-020-01640-1

[6] Architecture of Convolutional Neural Networks. [Online]. Available:https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

[7] Firebase authentication. [Online]. Available: https://firebase.google.com/docs/auth

[8] Kaggle. [Online]. Available: https://www.kaggle.com/

[9] Keras: A Python Deep Learning . [Online]. Available: https://keras.io/

[10] Tensorflow: A Machine Learning Platform. 2021. [Online]. Available: https://www.tensorflow.org/

[11] Flutter. [Online]. Available : https://developers.google.com/learn/topics/flutter

[12] Firebase . [Online]. Available: https://firebase.google.com/

[13] Python . [Online]. Available: https://www.python.org/doc/essays/blurb/

[14] Flask web framework . [Online]. Available: https://en.wikipedia.org/wiki/Flask_(web_framework)#cite_note-2

[15] Vishnoi, Vibhor Kumar, Krishan Kumar, and Brajesh Kumar. "Plant disease detection using computational intelligence and image processing." Journal of Plant Diseases and Protection 128 (2021): 19-53.

[16] Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. Information processing in agriculture, 8(1), 27-51.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (pp. 249-253).

[17] Abade, A., Ferreira, P. A., & de Barros Vidal, F. (2021). Plant diseases recognition on images using convolutional neural networks: A systematic review. Computers and Electronics in Agriculture, 185, 106125.

[18] Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683-56698.

[19] Sharma, P., Hans, P., & Gupta, S. C. (2020, January). Classification of plant leaf diseases using machine learning and image preprocessing techniques. In 2020 10th international conference on cloud computing, data science & engineering (Confluence) (pp. 480-484). IEEE

[20] Gayathri, S., Wise, D. J. W., Shamini, P. B., & Muthukumaran, N. (2020, July). Image analysis and detection of tea leaf disease using deep learning. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 398-403). IEEE.

[21] Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2022). A novel deep learning method for detection and classification of plant diseases. Complex & Intelligent Systems, 1-18.

[22] Xie, G., Attar, H., Alrosan, A., Abdelaliem, S. M. F., Alabdullah, A. A. S., & Deif, M. (2024). Enhanced diagnosing patients suspected of sarcoidosis using a hybrid support vector regression model with bald eagle and chimp optimizers. PeerJ Computer Science, 10, e2455.

[23] Deif, M. A., Attar, H., Hafez, M. A., Alomoush, W., & Al-Faiz, H. (2025). Automatic Sarcoidosis Stage Classification Based on Gray Level Co-occurrence Matrix Features. Appl. Math, 19(1), 197-208

[24] Ahmed, F. R., Alsenany, S. A., Abdelaliem, S. M. F., & Deif, M. A. (2023). Development of a hybrid LSTM with chimp optimization algorithm for the pressure ventilator prediction. Scientific Reports, 13(1), 20927.

[25] Attar, H., Ahmed, T., Rabie, R., Amer, A., Khosravi, M. R., Solyman, A., & Deif, M. A. (2024). Modeling and computational fluid dynamics simulation of blood flow behavior based on MRI and CT for Atherosclerosis in Carotid Artery. Multimedia Tools and Applications, 83(19), 56369-56390.

[26] Singh, V., Sharma, N., & Singh, S. (2020). A review of imaging techniques for plant disease detection. Artificial Intelligence in Agriculture, 4, 229-242.

[27] Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. Artificial Intelligence in Agriculture, 6, 23-33.

[28] Ashok, S., Kishore, G., Rajesh, V., Suchitra, S., Sophia, S. G., & Pavithra, B. (2020, June). Tomato leaf disease detection using deep learning techniques. In 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 979-983). IEEE.

[29] Nagaraju, M., & Chawla, P. (2020). Systematic review of deep learning techniques in plant disease detection. International journal of system assurance engineering and management, 11(3), 547-560.

[30] Shrivastava, V. K., & Pradhan, M. K. (2021). Rice plant disease classification using color features: a machine learning paradigm. Journal of Plant Pathology, 103(1), 17-26.

[31] Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105393.

[32] Chen, Junde, Jinxiu Chen, Defu Zhang, Yuandong Sun, and Yaser Ahangari Nanehkaran. "Using deep transfer learning for image-based plant disease identification." Computers and Electronics in Agriculture 173 (2020): 105393.

[33] Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8), 707.

[34] Vasavi, P., Punitha, A., & Rao, T. V. N. (2022). Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: A review. International Journal of Electrical and Computer Engineering, 12(2), 2079.

[35] Agarwal, M., Gupta, S. K., & Biswas, K. K. (2020). Development of Efficient CNN model for Tomato crop disease identification. Sustainable Computing: Informatics and Systems, 28, 100407.