SmartLeaf: Real-Time Diseased Plant Detection Using Convolutional Neural Networks for Urban Gardening Project in Barangay Santo Tomas, Pasig City

Prepared By: Baljon, John Lance B., Ocariza, James Andrew O., Rafer, Britney P.

Introduction

Urban gardening in areas like Barangay Santo Tomas, Pasig, allows people to grow plants in small spaces such as parks, backyards, rooftops, and community centers. This supports Pasig City's goal of becoming more eco-friendly through the "Pasig Green City" program (Ohmyhome, 2023). However, one major challenge is the spread of plant diseases. Many gardeners, especially beginners, find it hard to detect early signs of disease. They often rely on manual checking, which is time-consuming, inaccurate, and usually done too late when the plant is already damaged (Singh et al., 2022).

To address this, the study will develop a mobile application called SmartLeaf that uses Convolutional Neural Networks (CNN) to detect diseased plants in real time. CNN is a deep learning method that can analyze images and detect patterns agriculture (Pacal et al., 2024). The app allows users to simply take a photo of a plant to check if it is healthy or not. This solution is affordable and user-friendly, making it accessible to local gardeners who may not have technical knowledge or expensive equipment (Siddiqua et al., 2022). The goal is to provide a faster, simpler, and more accurate way to detect diseased plants and support the urban gardening community in Barangay Santo Tomas.

Purpose

The purpose of this study is to develop a mobile-based application called *SmartLeaf* that utilizes Convolutional Neural Networks (CNN) to detect whether a plant is healthy or diseased based on images of its leaves. This aims to address the limitations of manual checking methods, which are often time-consuming, inaccurate, and prone to human error. By providing real-time detection through a user-friendly mobile platform, the study seeks to make diseased plants identification more accessible, especially for non-expert urban gardeners in communities like Barangay Sto. Tomas in Pasig City. Through this, the study hopes to contribute to the promotion of sustainable urban gardening by helping gardeners protect their plants and improve overall plant health.

Comparison of Algorithms

This case study evaluates three CNN architectures to determine the best model for SmartLeaf:

- **CNN-Sequential** – A simple CNN model with convolutional, pooling, and dense layers designed as a baseline lightweight architecture trained from scratch. While fast, it has lower accuracy.

- **ResNet50** – A deep 50-layer pretrained residual network using frozen layers for transfer learning, improving accuracy and generalization with minimal training. However, it is slower and requires more storage.
- **MobileNetV2** – A lightweight, efficient pre-trained model optimized for mobile use with depth wise separable convolutions, balancing accuracy and speed for real-time deployment.

Each model was developed using Keras with image processing tasks supported by libraries like OpenCV and NumPy. Performance was evaluated using metrics including accuracy, F1-score, inference time, and model size to determine the most effective machine learning approach.

Performance Results

The table below presents the performance of each model based on four key metrics: Accuracy, F1-Score, Inference Time, and Model Size. The dataset contains two classes: healthy and not healthy. Each model was evaluated to see how well it performs in classifying these conditions.

Model	Accuracy	F1-Score		Interference Time	Model Size
		healthy	nothealthy		
CNN-Sequential	94.44%	94%	95%	~ 0.6139 ms	~ 84.64 mb
ResNet50	95.45%	95%	96%	~ 14.2980 ms	~ 93.61 mb
MobileNetV2	98.48%	98%	99%	~ 7.3616 ms	~ 11.05 mb

Table: Performance Results of each model.

Confusion Matrix: [[88 2] [9 99]]	Confusion Matrix: [[85 5] [4 104]]	Confusion Matrix: [[88 2] [1 107]]
CNN-Sequential	MobileNetV2	ResNet50

Table: Confusion Matrix of each model.

Based on the performance results, **MobileNetV2** is the best choice for SmartLeaf due to its highest accuracy (98.48%) and strong F1-scores (98% healthy, 99% diseased). While **CNN-Sequential** is the fastest, it has low accuracy, making it less ideal for diseased plants detection.

ResNet50 has good accuracy but is significantly slower (~14.2980 ms) and has the largest file size (~93.61 MB), making it impractical for mobile use.

MobileNetV2 is lightweight (~11.05 MB) and fast, ensuring efficient real-time detection on mobile devices.

Discussion

Among the models tested, MobileNetV2 emerged as the most suitable for the SmartLeaf application because it balances accuracy, inference speed, and storage efficiency. Since the goal is real-time detection on mobile devices, MobileNetV2's compact size ensures that it runs smoothly without straining resources.

Practical Considerations

- **Storage Efficiency:** MobileNetV2 is significantly smaller (~11 MB) compared to CNN-Sequential (~84 MB) and ResNet50 (~93 MB), making it ideal for phones with limited storage.
- **Real-Time Performance:** The inference speed of MobileNetV2 (~7.3616 ms) ensures rapid diseased plant detection, which is crucial for urban gardeners who need immediate results.
- **Reducing False Diagnoses:** The high F1-score minimizes false positives and false negatives, ensuring that gardeners receive reliable diagnoses without frequent errors.

The deployment of SmartLeaf powered by MobileNetV2 can greatly support local urban gardening efforts in Barangay Santo Tomas, allowing gardeners to:

- **Reduce plant loss** by detecting diseased plants early.
- **Improve productivity** with faster and more accurate plant health assessments.
- **Encourage tech-based sustainable gardening** in urban spaces.

While MobileNetV2 performs well, future improvements could focus on handling varying image conditions, such as low lighting or complex backgrounds. Additionally, exploring offline functionality would make SmartLeaf even more accessible to communities with limited internet connectivity.

This study successfully demonstrates that CNN-based diseased plant detection can improve urban gardening. After evaluating three CNN architectures, MobileNetV2 proved to be the most effective, combining high accuracy, fast inference time, and minimal storage requirements. As urban gardening continues to expand, SmartLeaf offers a practical and scalable solution for diseased plant detection, helping gardeners grow healthier plants and contribute to a more eco-friendly Pasig City.

References:

Ishak Pacal, Ismail Kunduracioglu, Alma, M. H., Muhammet Deveci, Seifedine Kadry, Nedoma, J., Vlastimil Slany, & Martinek, R. (2024). A systematic review of deep learning techniques for plant diseases. *Artificial Intelligence Review*, *57*(11). https://doi.org/10.1007/s10462-024-10944-7

Siddiqua, A., Kabir, M. A., Ferdous, T., Ali, I. B., & Weston, L. A. (2022). Evaluating Plant Disease Detection Mobile Applications: Quality and Limitations. *Agronomy*, *12*(8), 1869. https://doi.org/10.3390/agronomy12081869

Singh, S., Bajpai, R., Rashid, M. M., Basavaraj Teli, & Kumar, G. (2022, July 15). *Plant Disease Detection with the Help of Advanced Imaging Sensors*. https://doi.org/10.1201/9781003311782-8

Why is Pasig City the Green City and What Does It Mean for Homeowners? | Ohmyhome. (2023, May 18). OhMyHome. https://ohmyhome.com/en-ph/blog/why-is-pasig-city-the-green-city-and-what-does-it-mean-for-homeowners/

Dataset

The dataset used in this study is solely based on the publicly available PlantVillage dataset by roboflow, containing labeled images of healthy and diseased plant leaves. No data was collected directly from Barangay Sto. Tomas; however, testing of the developed model was focused on serving the urban gardening community in that area to evaluate its effectiveness and responsiveness. The dataset was divided into 2 classes: healthy and diseased leaves (generalized). In preprocessing, the images were resized to fit the model input requirements. Normalization was also applied to scale up the images. Along with these set-ups, data augmentation such as rotations and brightness of the image are also considered.

Comparison of Algorithms

This case study used three different CNN architectures, which include CNN-Sequential, a simple CNN with convolutional, pooling, and dense layers with dropout; it serves as a baseline lightweight model trained from scratch. ResNet50, a deep 50-layer pretrained residual network using frozen layers for transfer learning; this aims for higher accuracy and better generalization with less training. And MobileNetV2, a lightweight, efficient pre-trained model optimized for mobile use with depthwise separable convolutions; it balances accuracy and speed for real-time deployment. All the models use Keras, a high-level deep learning API that simplifies the process of building deep neural networks, and are capable of utilizing CNN architectures, to determine what is the best and most effective machine learning model to use for the study.

Training Environment

All models in this study were developed and trained using Python with Keras built on TensorFlow as the deep learning framework. Supporting libraries such as OpenCV and NumPy were used for image processing, while Matplotlib and scikit-learn were used for performance visualization and metric evaluation. The training process included resizing images to 128×128 pixels and normalizing pixel values. Models were saved in .keras format, and outputs such as confusion matrices and sample prediction batches were stored for further analysis.

Evaluation Metrics

The evaluation process employed consistent metrics across all models to ensure a fair comparison. These metrics included accuracy, precision, recall, F1-score, confusion matrix visualization, inference time measured on the test set, and model size in megabytes. These metrics provide insight into each model’s prediction correctness, balance between false positives and false negatives, speed, and feasibility for deployment on resource-limited devices. Additionally, sample prediction visualizations were generated to assess model performance qualitatively.

All models were developed and trained using Python, employing Keras built on TensorFlow as the deep learning framework. Image processing tasks were supported by libraries like OpenCV and NumPy, while Matplotlib and scikit-learn were utilized for visualizing performance metrics and evaluating results. The trained models were saved in the .keras format. Additionally, outputs such as confusion matrices and sample prediction batches were stored to facilitate further analysis

The evaluation process used consistent metrics across all models to ensure fair comparison. These metrics included accuracy, precision, recall, F1-score, confusion matrix visualization, inference time measured on the test set, and model size in megabytes. Together, these provided insights into the models' prediction correctness, balance between false positives and false negatives, inference speed, and suitability for deployment on resource-limited devices. Qualitative assessments through sample prediction visualizations were also generated to better understand the model's real-world performance.