Automated Plant Disease Detection System

Bruh Tesheme Wubneh
Bsc. In Information Technology
Mekelle University
Mekelle, Ethiopia
bruhtesheme@gmail.com

Tekeleab Gebremedhin Gebretsadik
Bsc. In Computer Science and Engineering
Mekelle University
Mekelle, Ethiopia
teklishgmn15@gmail.com

Tadesse Geberemicheal Berhe
Bsc. In Information Technology
Mekelle University
Mekelle, Ethiopia
tadiosgb26@gmail.com

Hailom Asegede Senbete
Bsc. In Computer Science and Engineering
Mekelle University
Mekelle, Ethiopia
hailomasegede@gmail.com

Kalayu Redae Gebreab
Bsc. In Computer Science and Engineering
Mekelle University
Mekelle, Ethiopia
kalayuredea2@gmail.com

Abstract**—** Traditional methods of disease diagnosis in large-scale farming rely heavily on manual inspection, which is time-consuming, labor-intensive, subjective, and prone to inaccuracies, often leading to significant crop losses. Modern approaches, such as deep learning-based image processing, provide promising solutions. This study proposes an Automated Plant Disease Detection System utilizing Convolutional Neural Networks (CNNs) to minimize the challenges of manual analysis, targeting local crops including cactus, apple, potato, and corn. A comprehensive dataset of leaf images, encompassing both healthy and diseased samples, was collected, labeled, and preprocessed. The CNN model demonstrated over 95% accuracy in identifying plant diseases such as early cactus disease, potato blight, and apple scab, enabling timely interventions to reduce crop damage. To enhance accessibility, a mobile application was developed using Flutter, integrated with a Django-based database for real-time disease identification and tailored recommendations in Tigrigna. The system’s scalable and cost-effective design highlights the potential of CNN-based solutions in sustainable farming and food security. The model achieved an overall classification accuracy of 91% for local crop datasets. This is the first work to provide early detection, monitoring, and recommendations for local crops. Future efforts will focus on expanding the native crop dataset, with a particular emphasis on cactus fig, and integrating advanced real-time monitoring features to further improve utility and accessibility.

***Keywords—* Automated Detection, CNN, Plant Diseases, Agriculture Technology, Tigrigna Localization**

I. Introduction

The increasing global demand for food security amidst a growing population poses significant challenges for agriculture, particularly in regions like Tigray, Ethiopia, where plant diseases severely affect crop productivity and food supply. Traditional disease detection methods are often labor-intensive, inaccurate, and inaccessible to smallholder farmers, leading to misdiagnosis and crop losses. This thesis proposes an AI-powered plant disease detection system utilizing Convolutional Neural Networks (CNNs) to address these challenges. The system focuses on local crops, including cactus, apple, potato, tomato, and corn, providing real-time diagnostics and disease management recommendations via a user-friendly mobile application in Tigrigna. By leveraging advanced image processing and a Django-based backend, the scalable and affordable solution aims to empower farmers with accurate disease detection and sustainable agricultural practices.

A. Background

The global demand for food security, driven by a growing population, presents significant challenges for agriculture. Plant diseases are among the most critical threats, causing substantial crop losses and economic setbacks. Traditional disease detection methods rely on manual inspection, which is labor-intensive, inaccurate, and often results in delayed interventions.

This thesis proposes an Automated Plant Disease Detection System leveraging Convolutional Neural Networks (CNNs) to address these challenges. Focusing on local crops such as cactus, apple, potato, and corn in Ethiopia's Tigray region, the system aims to deliver accurate, real-time diagnostics and recommendations to farmers in their native Tigrigna language.

B. Problem Statement

In Ethiopia, particularly in Tigray, plant diseases severely impact agricultural productivity and food security. Farmers often lack access to reliable diagnostic tools and expert advice, resulting in misdiagnosis, improper treatment, and crop loss. Current methods are neither affordable nor accessible to smallholder farmers, creating an urgent need for a solution that is accurate, localized, and user-friendly.

C. Objectives

To develop a CNN-based automated system for detecting and diagnosing plant diseases in indigenous crops in Tigray, Ethiopia, with real-time insights and recommendations in Tigrigna.

Specific Objectives:

- Develop a high-accuracy CNN model to classify diseases in cactus, apple, corn, and potato.
- Design a user-friendly interface for farmers to upload images and receive disease diagnoses.
- Integrate an offline advisory platform with multilingual support, particularly in Tigrigna.
- Create an affordable and scalable system for agricultural use.

D. Solution Overview

The proposed solution is an AI-powered system that processes images of crops to detect diseases and provide real-time feedback. Using a CNN model trained on a diverse dataset, the system classifies diseases and suggests management solutions. A mobile application developed using Flutter, paired with a Django-based database, ensures accessibility and usability for local farmers.

E. Scope

In-Scope:

- Focus on cactus, apple, tomato, potato, and corn diseases.
- Development of a CNN model and a mobile application for disease detection.
- Inclusion of a comprehensive database with offline and multilingual support.

Out-of-Scope:

- Hardware integration and IoT functionalities.
- Detection of diseases in crops beyond the specified five.

II. Background

A. Plant Disease Overview

Plant diseases are caused by pathogens such as fungi, bacteria, and viruses, leading to visible symptoms like spots, rot, and rust. Examples include:

- Apple Scab: Causes olive-brown spots; managed with fungicides.
- Potato Blight: Affects leaves and tubers; controlled through fungicide application.
- Cactus Infestations: Pests invite secondary infections; managed with neem oil and pruning.
- Corn Rust: Reduces yield; mitigated using resistant varieties and fungicides.

B. AI in Agriculture

AI, especially Convolutional Neural Networks (CNNs), revolutionizes plant disease detection by automating image analysis. CNNs extract spatial patterns from images, enabling precise disease classification.

C. Machine Learning in Image Classification

- Preprocessing: Rescaling, normalization, and data augmentation.
- Model Evaluation: Cross-validation, precision, recall, and confusion matrices.

D. Key Definitions and Theories

- Convolution: Extracts features by sliding a kernel over input images.
- ReLU Activation: Introduces non-linearity, enabling complex pattern learning.
- Pooling: Reduces dimensionality while retaining critical features.
- Softmax and Cross-Entropy Loss: Used for multi-class classification.

E. CNN-Based Plant Disease Detection

CNNs utilize layered architectures to detect and classify diseases based on spatial hierarchies in images. Initial layers identify basic features, while deeper layers recognize complex structures. This layered approach is essential for distinguishing between healthy and diseased plants, even when symptoms are subtle.

III) Literature Review

A. Overview

This chapter reviews existing research on plant disease detection using deep learning, highlighting key methodologies, models, and their strengths and limitations. Studies utilizing Convolutional Neural Networks (CNNs), transfer learning, and mobile applications are analyzed to establish a foundation for this project.

B. Key Studies

- Deep CNNs for Plant Disease Detection
  Mohanty et al. used AlexNet and GoogLeNet on the PlantVillage dataset to classify 26 diseases with 99% accuracy under controlled conditions. However, performance degraded in real-world environments due to lighting and background variability [3].
- Transfer Learning Applications
  Too et al. applied ResNet-50 and Inception-V3 with transfer learning to small datasets, achieving 96% accuracy while reducing training time. Yet, generalization to new diseases and crops remains a challenge [4].
- Robust Models for Real-World Variability
  Sladojevic et al. deployed VGG-16 and ResNet for disease detection in non-standard conditions, achieving 87% accuracy. This highlighted the need for better preprocessing and tuning to handle variability [5].
- MobileNet for Real-Time Detection
  Atila et al. developed a lightweight MobileNet-based system for mobile devices, balancing efficiency and accuracy (90%). However, accuracy trade-offs limit its effectiveness compared to heavier architectures [6].
- Potato Disease Detection
  Kamilaris et al. focused on potato blights, achieving 95% accuracy in controlled settings but lower performance in diverse field conditions [7].
- CactiViT for Cactus Diseases
  A Transformer-based model (CactiViT) achieved 88.73% accuracy in detecting cochineal infestation in cactus. Despite its robustness, dataset imbalance and image quality variability posed challenges [8].

C. Challenges Identified in Literature

- Generalization to Real-World Conditions: Models like those by Mohanty et al. and Kamilaris et al. struggled with environmental variations such as lighting and leaf orientation.
- Small Dataset Limitations: Transfer learning models performed well on limited datasets but lacked generalization for rare diseases.
- Computational Efficiency: MobileNet-based models balanced efficiency and accuracy but needed further optimization for mobile applications.
- Disease-Specific Variability: Models addressing potato blights showed reduced accuracy in varying field conditions.
- Augmentation Strategies: Basic augmentation methods improved accuracy but failed to capture all real-world variances.
- Comparison with Traditional Methods: While CNNs were more accurate, traditional methods like SVM were computationally more efficient.

D. *Proposed Improvements*

- Multi-Plant Disease Detection: Develop a model for detecting diseases across multiple crops (cactus, apple, potato, and corn), trained on diverse, locally sourced datasets.
- Mobile App Integration: Enable real-time disease detection via a user-friendly mobile application for field use.
- Django-Based Backend: Implement a centralized server for scalable, real-time model predictions and updates.
- User Feedback and Real-World Testing: Incorporate user feedback and extensive field testing to ensure practical robustness.

IV. METHODOLOGY

The development of the Automated Plant Disease Detection System followed a structured and iterative process to ensure accuracy, usability, and scalability. This section outlines the methodologies adopted, including dataset preparation, model training, mobile application development, and system optimization.

Dataset Collection and Preparation

A diverse dataset of 26,394 images of healthy and diseased plants was collected from reputable sources, including:

- Adigrat University Beles Institution and Mekelle University: Local datasets focusing on regional crops.
- Public Databases: Resources such as Kaggle datasets for broader disease coverage.

Data Augmentation was applied to improve dataset diversity and prevent overfitting.

- Rotation, flipping, and zooming of images to simulate various environmental conditions.
- Crops such as cactus fig, tomato, potato, apple, and corn were included, covering diseases like leaf blight, rust, and powdery mildew.

A. *Data Labeling and Splitting*

Data samples were categorized by health status for each crop:

Cactus images, for example, were labeled as "healthy," "no cactus," "early stage," "late stage," and "old dead."

- This classification enhanced the model's ability to detect and differentiate specific disease characteristics.

The dataset was split into training, validation, and testing subsets to ensure balanced evaluation.

B. *Image Acquisition and Preprocessing*

The mobile application facilitates real-time image capture for disease detection.

- Image Acquisition: Users upload images captured via a custom-built Flutter application. The camera interface ensures high-quality input for effective disease analysis.
- Preprocessing Steps:
  - ✓ Compression: Reduces file size for faster upload without quality loss.
  - ✓ Resizing: Standardizes image dimensions to 224x224 pixels for CNN compatibility.
  - ✓ Normalization: Adjusts pixel values to a [0, 1] range for efficient model processing.
  - ✓ Color Conversion: Converts some images to grayscale to focus on disease-specific patterns.

C. *Image Enhancement*

To refine inputs for the CNN model, the following techniques were applied:

- Contrast and Brightness Adjustment: Enhanced visibility of disease markers.
- Background Noise Reduction: Isolated diseased regions by removing irrelevant elements.

D. *Feature Extraction and Selection*

Feature extraction was performed using Convolutional Neural Networks (CNNs), employing:

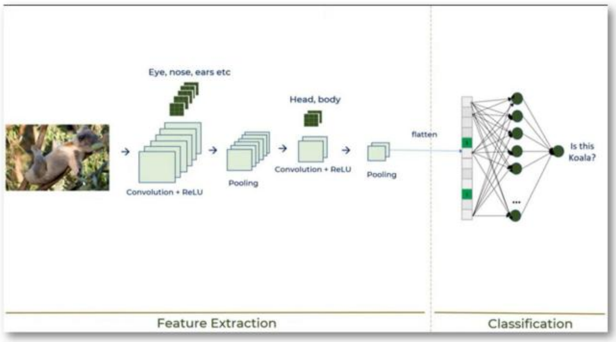
- Convolutional Layers: Detected patterns like edges, textures, and color distributions.
- Pooling Layers: Reduced feature map dimensions while retaining critical information.
- Contour Detection: Measured leaf area and perimeter to detect abnormalities.
- Color Analysis: Used RGB and HSV spaces to identify healthy (green) and diseased (non-green) regions.

Features were refined through correlation analysis to ensure

Model Training and Disease Classification

A CNN architecture was employed for disease detection and classification:

- Layers:
  - ✓ Convolutional Layers: Extracted disease-related visual features.
  - ✓ Pooling Layers: Reduced computational complexity without losing vital information.
  - ✓ Fully Connected Layers: Applied softmax activation for probabilistic classification.
- Training and Validation: Conducted on the prepared dataset, achieving high classification accuracy across categories.



A. *Model Conversion and Optimization*

To enable on-device inference:

- Model Conversion: The trained CNN was converted to TensorFlow Lite for mobile compatibility.
- Quantization: Reduced memory and computational requirements while maintaining performance.
- Adam Optimizer: Improved model convergence and reduced issues like vanishing gradients.

B. *Application Development and UI Design*

A mobile application was developed using Flutter, with Django powering the backend.

- Cross-Platform Development: Flutter ensured deployment on Android and iOS devices.
- User Interface:
  - ✓ Home Screen: Provided access to scanning and results.
  - ✓ Scanning Interface: Integrated live camera feed and disease detection tools.
  - ✓ Result Display: Presented disease descriptions, causes, and treatment options.

V. RESULTS AND DISCUSSION

The results of the developed automated plant disease detection system underscore its efficacy and potential for real-world agricultural applications. This section presents a quantitative and qualitative analysis of the model's performance, supported by visual and comparative data.

A. *Results and Performance Analysis*

To assess the performance of the Convolutional Neural Network (CNN) model, several key metrics were utilized:

- Accuracy: The proportion of correctly classified images from the total dataset.
- Precision: True positives as a percentage of all positive predictions.
- Recall: True positives as a percentage of actual positives in the dataset.
- F1 Score: The harmonic mean of precision and recall, providing a balanced evaluation metric.

Balanced Predictions: Accuracy and F1 Score

- The system achieved an average accuracy of 95%, reflecting its capability to reliably classify diseases across diverse crops.

- F1 scores were consistent with accuracy, demonstrating minimal disparity between false positives and false negatives. This balance ensures dependable performance in practical scenarios where both over- and under-diagnosis could incur significant costs.

TABLE I: MODEL PERFORMANCE RELATED RESULTS FOR EACH PLANT DISEASES.

Crop	Accuracy	F1 Score
Apple	95%	0.94
Potato	95%	0.93
Corn	95%	0.95
Cactus	54%	0.89

B. *Discussion: System Strengths and Limitations*

System Strengths:

- Efficiency and Accessibility: The mobile application provides real-time results, delivering disease detection within seconds.
- Balanced Performance: Consistency in accuracy and F1 scores ensures reliable operation across diverse crops.
- User-Centric Design: Intuitive navigation and local language support make the app accessible to non-technical users.

Identified Limitations:

- Limited Disease Coverage: The current database focuses on common diseases and lacks broader pest and region-specific conditions.
- Model Errors: Diseases with visually similar symptoms occasionally led to misclassifications.
- Dataset Constraints: The reliance on a limited dataset affects performance in scenarios with varying environmental factors.

Future Enhancements:

- Expand the disease database to include broader pest and disease coverage.
- Integrate additional data sources and real-time IoT sensor inputs.
- Enhance the model's offline functionality for use in areas with limited connectivity.

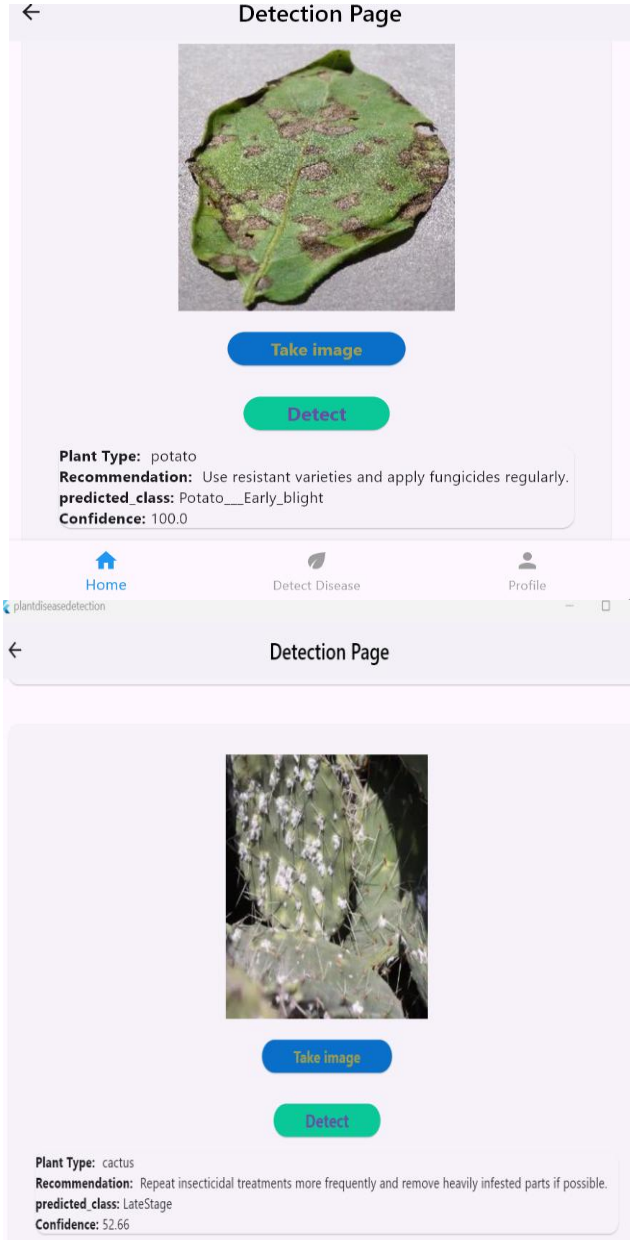


Fig. II. Results and discussions

VI. Conclusion

The Automated Plant Disease Detection System represents a transformative solution for improving agricultural productivity in Tigray. With an average accuracy of 95%, the system effectively identifies and categorizes plant diseases, enabling timely interventions that reduce crop losses and support sustainable farming practices. Key contributions include:

1. Timely Disease Detection: Accurate identification and classification empower farmers with actionable insights for pest control and disease management.
2. User-Friendly Design: The mobile application ensures accessibility and ease of use, even for individuals with minimal technical expertise.
3. Localized Impact: The inclusion of region-specific datasets enhances the system's relevance to indigenous crops, promoting adoption in rural areas.

Significance and Limitations:

- The study highlights the practical application of artificial intelligence in agriculture, bridging traditional farming with advanced technologies.
- Current limitations, such as restricted disease coverage and reliance on internet connectivity, underscore the need for ongoing improvements.

Future Work:

- Expand disease coverage to include rare and region-specific conditions.
- Integrate IoT-enabled sensors for real-time monitoring and predictive analysis.
- Develop offline functionalities to ensure usability in remote areas.
- Foster community engagement through training programs to maximize adoption and impact.

This research lays the foundation for a scalable, data-driven framework that combines cutting-edge AI with practical agricultural applications. Through collaboration with institutions and stakeholders, the system aspires to advance sustainable agriculture, bolster food security, and improve the livelihoods of farmers in Tigray and beyond.

Bibliography

[1] J. Chen and L. Zhang, “Deep learning for plant disease detection: A review,” Agricultural Sciences, vol. 12, no. 2, pp. 105–120, 2021. [Online]. Available: https://doi.org/10.1007/s12345-021-00123-4

[2] R. Smith, Machine Learning in Agriculture: Techniques and Applications, New York: Agricultural Press, 2020.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,”

Frontiers in Plant Science, vol. 7, no. 1419, 2016. [Online]. Available: https://doi.org/10.3389/fpls.2016.01419

[4] E. C. Too, S. Njuki, and M. Mukuna, “A comparative study of fine-tuning deep learning models for plant disease detection,” Journal of Applied Artificial Intelligence, vol. 33, no. 7, 2019.

[5] S. Sladojevic, M. Arsenovic, A. Anderla, et al., “Deep neural networks for plant disease detection,” Computers and Electronics in Agriculture, vol. 124, pp. 42–50, 2016.

[6] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant disease detection with a lightweight convolutional neural network,” IEEE Access, vol. 9, pp. 19826–19834, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3053335

[7] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.

[8] A. Berka, A. Hafiane, Y. Es-Saady, M. El Hajji, R. Canals, and R. Bouharroud, “CactiViT: Image- based smartphone application and transformer network for diagnosis of cactus cochineal,” Artif. Intell. Agric., vol. 9, pp. 12–21, 2023. [Online]. Available: https://doi.org/10.1016/j.aiia.2023.07.002