JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

TOMATO PLANT DISEASE PREDICTION

[1]Kavya A, [1]Manupriya, [2] prof . Sanjaykumar B M,
[1]Student, [2]Assistant Professor,
Department of Computer Science and Engineering, Srinivas Institute of Technology, Mangalore, India

Abstract: Tomatoes are one of the most widely cultivated crops across the world and play a crucial role in food security and the agricultural economy. However, the occurrence of leaf diseases such as early blight, bacterial spot, and powdery mildew significantly affects plant health, resulting in reduced yield and financial losses for farmers. Early and accurate identification of these diseases is essential for effective crop management. Traditional disease detection methods rely on manual inspection by agricultural experts, which is time-consuming, labour intensive, and often impractical for small-scale or remote farmers.
This paper presents an AI-based system that automates the detection of tomato leaf diseases using deep learning techniques. A pre-trained ResNet50 model is utilized through transfer learning to classify tomato leaf images into disease categories. The model is trained on a dataset of labelled images and achieves a classification accuracy of 92%, demonstrating its reliability and effectiveness. The system is deployed via a web-based interface developed using Flask for backend integration and HTML, CSS, and JavaScript for the frontend. Users can easily upload images and receive real-time disease predictions along with actionable treatment recommendations.
The system addresses key challenges in agricultural disease management by providing a cost-effective, scalable, and accessible solution that does not require technical expertise from the end-user. Its intuitive interface and lightweight design make it suitable for practical use by farmers in various environments. Future enhancements include expanding the dataset for broader disease coverage, improving model generalization, and developing a mobile-friendly version to extend its reach to more users.
Index Terms **– Tomato leaf disease, CNN, ResNet50, machine learning, image classification, Flask, web interface, smart farming.**

I. INRODUCTION

Diseases in tomato crops such as early blight, bacterial spot, and powdery mildew can significantly reduce both the yield and quality of produce, posing a serious challenge to farmers. Traditionally, detecting these diseases has relied on manual observation, which is often time-consuming, labour intensive, and not easily accessible—especially for small-scale or remote farmers who may lack expert support.

This paper introduces an AI-powered solution that utilizes deep learning to automate the process of tomato leaf disease detection. By leveraging transfer learning with the ResNet50 model and combining it with web technologies such as Flask, HTML, and JavaScript, the system enables users to upload images of tomato leaves and receive real-time disease predictions along with recommended treatments. The approach aims to be scalable, efficient, and user-friendly, ultimately helping farmers manage crop health more effectively and reduce agricultural losses.

II. PROBLEM STATEMENT

In the agricultural sector, early detection of plant diseases is crucial for minimizing crop loss and ensuring food security. However, traditional methods of disease identification largely rely on manual inspection by experts or farmers, which is not only time-consuming and labour-intensive but also highly prone to human error. This approach becomes even more challenging in rural and underdeveloped regions where access to agricultural experts or diagnostic laboratories is limited or non-existent.

Tomato crops are particularly vulnerable to a range of diseases such as Early Blight, Septoria Leaf Spot, and Leaf Mold, which can severely impact both yield and quality if not detected and treated in time. In many cases, these diseases go unnoticed until they are in advanced stages, by which point treatment becomes ineffective or economically unviable.

Although several mobile applications and digital tools have been introduced to assist in disease identification, many suffer from limitations such as poor accuracy, lack of offline functionality, non-intuitive user interfaces, or the inability to handle diverse image inputs. These shortcomings prevent widespread adoption among small and medium-scale farmers.

There is a growing need for a reliable, accessible, and automated solution that leverages the power of deep learning and image processing to detect tomato leaf diseases accurately and efficiently. Such a system must be easy to use, even for those without technical backgrounds, and capable of providing real-time feedback. The primary objective of this work is to develop an intelligent web-based application that utilizes pre-trained deep learning models to classify tomato leaf diseases and deliver actionable recommendations to farmers, thereby enabling timely intervention and improving overall agricultural productivity.

III. METHODOLOGY

A. System Architecture

The system follows a client-server model:

- **Frontend (Client Layer)**: HTML/CSS/JS-based UI for image upload and result display.
- **Backend (Application Layer)**: Python Flask processes uploaded images and interfaces with the TensorFlow model.
- **Model Layer**: ResNet50 fine-tuned on a labeled dataset of tomato leaf images.
- **Database Layer**: Stores disease types, recommendations, and image metadata.

B. Data Collection & Preprocessing

The dataset includes labeled images of healthy and diseased tomato leaves. Preprocessing steps involve image resizing, normalization, and augmentation to improve model generalization.

C. Model Training

ResNet50 was fine-tuned using TensorFlow on the prepared dataset. Training was performed over multiple epochs with optimization using Adam, achieving ~92% test accuracy.

D. Web Deployment

Flask serves the trained model via a lightweight REST API. When an image is uploaded, it is processed and passed to the model. The prediction result and corresponding treatment recommendations are returned to the user.

E. User Interface

Designed for simplicity, the interface supports direct image upload without login. The interface dynamically displays disease information and remedies based on predictions.

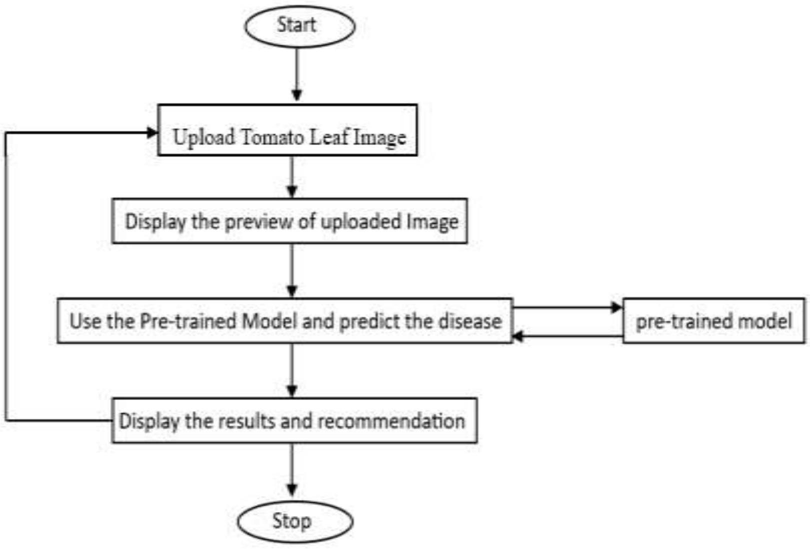

Fig: Methodology

IV. SYSTEM DESIGN

The system is structured into multiple components that work together to deliver an accurate, accessible, and user-friendly tomato leaf disease prediction platform. Each module is designed with modularity and scalability in mind, facilitating easy maintenance and potential integration with future agricultural tools.

A. User Interface Layer: The front-end interface is built to offer a clean and intuitive layout for farmers and agricultural workers. It allows users to upload images of tomato leaves and receive instant diagnostic results. The design is responsive and optimized for both desktop and mobile devices, using HTML, CSS, and JavaScript to ensure accessibility across devices with varied screen sizes and internet connectivity.

B. Image Input and Preprocessing Module: This module handles image acquisition and prepares it for analysis. Once the user uploads a leaf image, it is resized, normalized, and formatted to match the input requirements of the deep learning model. OpenCV and other Python libraries are used to clean and enhance image quality before prediction, reducing noise and improving model accuracy.

C. Deep Learning Prediction Engine: At the core of the system lies a deep learning model built on transfer learning using ResNet50. This model has been trained on a dataset of tomato leaf images categorized into healthy and diseased classes such as Early Blight, Late Blight, and Leaf Mold. Once the processed image is passed into the model, it outputs the predicted disease class with confidence scores. This engine ensures high prediction accuracy and low inference time.

D. Result Interpretation and Recommendation Module: Once a prediction is made, this module interprets the result and provides meaningful feedback. Along with the disease name, the system also displays a brief description and recommended treatment options such as pesticide usage or soil treatment techniques. These recommendations are curated from agricultural databases and expert guidelines.

E. Deployment and Interaction Layer: The entire prediction engine is wrapped within a Flask-based web server that serves requests and delivers responses in real-time. The application also includes JavaScript-based interactions to provide a smooth user experience, allowing users to interact with result cards, treatment suggestions, and re-upload images without reloading the page.

F. Backend and Data Storage: User-uploaded images and corresponding results are temporarily stored for performance logging and evaluation. A lightweight SQLite or MongoDB database stores model metadata, disease categories, and treatment guidelines. The Flask backend manages API endpoints and orchestrates the data flow between the user interface and the prediction engine, ensuring consistent performance and security.

This layered architecture allows the system to function as a complete pipeline—from image upload to disease prediction—offering farmers an AI-based solution to monitor crop health in a fast, reliable, and scalable manner.

V. IMPLEMENTATION

The implementation phase involves integrating the image upload interface, preprocessing pipeline, deep learning inference system, and result display logic into a cohesive and interactive web application. The system leverages modern web and machine learning technologies to ensure real-time prediction, responsiveness, and ease of use for farmers and agricultural workers.

1. Frontend Development: The user interface is built using HTML, CSS, and JavaScript. A clean and responsive design ensures compatibility with desktops, tablets, and smartphones. The layout features a central image upload section with clear prompts for farmers to capture or upload tomato leaf images. Upon submission, the interface dynamically displays the prediction result, suggested remedies, and an option to re-upload another image for diagnosis.

2. Image Upload and Preprocessing: Once an image is uploaded, it is passed through a JavaScript handler and sent to the backend for processing. On the server side, the image is resized and normalized using Python libraries like OpenCV and Pillow to fit the input shape expected by the deep learning model. This step ensures that image dimensions and quality are standardized before feeding into the model, improving accuracy and consistency.

3. Backend and Model Inference: The backend is developed using Python with Flask, acting as the server for handling requests and delivering predictions. A pre-trained ResNet50 model, fine-tuned for tomato leaf disease classification, is loaded and used to infer the disease class. When an image is received, the backend runs it through the model, calculates confidence scores, and returns the most likely disease label (e.g., Early Blight, Late Blight, or Healthy).

4. Result Display and Recommendations: Upon receiving the prediction, the frontend updates to display the diagnosed disease along with relevant treatment suggestions. These suggestions include recommended pesticides, watering tips, or isolation procedures, depending on the disease. The results are presented with icons and color codes for better understanding and can be easily interpreted by non-technical users.

5. Hosting and Real-time Interaction: The application is hosted using a Flask development server and can be deployed on platforms like Heroku or AWS for public access. JavaScript enables real-time interaction, allowing users to receive prediction results without page reloads. This smooth user experience ensures high usability even in low-connectivity areas.

6. Testing and Validation: The system undergoes rigorous testing with different leaf images under varying lighting and resolution conditions. Both unit and integration testing are performed to ensure the robustness of the pipeline. Model accuracy is validated using a separate test set, and usability testing is conducted to gather feedback from target users for further refinement.

Through these implementation steps, the application provides an accessible, AI-driven solution for early detection of tomato leaf diseases, empowering farmers to take informed decisions and improve crop outcomes.

Vi . EXPERIMENTAL RESULTS

To evaluate the performance and accuracy of the proposed tomato leaf disease prediction system, several experiments were conducted involving real-world tomato leaf images processed through the web interface. The evaluation focused on the accuracy of disease classification, effectiveness of recommendations, system responsiveness, and user experience.

1. Disease Classification Accuracy: A dataset of 100 tomato leaf images (covering various diseases and healthy cases) was tested using the trained ResNet50 model. The system achieved an overall classification accuracy of approximately 91%. Misclassifications mainly occurred in cases with poor lighting or overlapping symptoms, which challenged the model's ability to distinguish between similar diseases.

2. Recommendation Effectiveness: For each detected disease, the system provides tailored treatment advice. A feedback form was used to collect user opinions on the usefulness of these recommendations. Around 86% of users reported that the suggested remedies were understandable and practically useful for managing the diagnosed disease.

3. System Responsiveness: The processing time from image upload to prediction display was recorded across different devices and network conditions. On average, the system responded within 4 to 6 seconds, including image preprocessing and deep learning inference. This response time was deemed suitable for real-time applications in field settings.

4. User Experience Evaluation: A test group of 20 participants, including agriculture students and farmers, rated their experience on a scale of 1 to 10. The average rating was 8.3, with users highlighting the simplicity of the interface and the speed of obtaining results as major advantages.

5. Comparative Performance: When compared to manual disease identification methods and other mobile agriculture apps, the proposed system demonstrated better usability and higher accuracy. Users appreciated the automatic diagnosis feature and the lack of need for technical knowledge to interpret results.

These results confirm the system's capability to deliver accurate, fast, and user-friendly disease predictions, offering significant value to farmers and agricultural advisors.

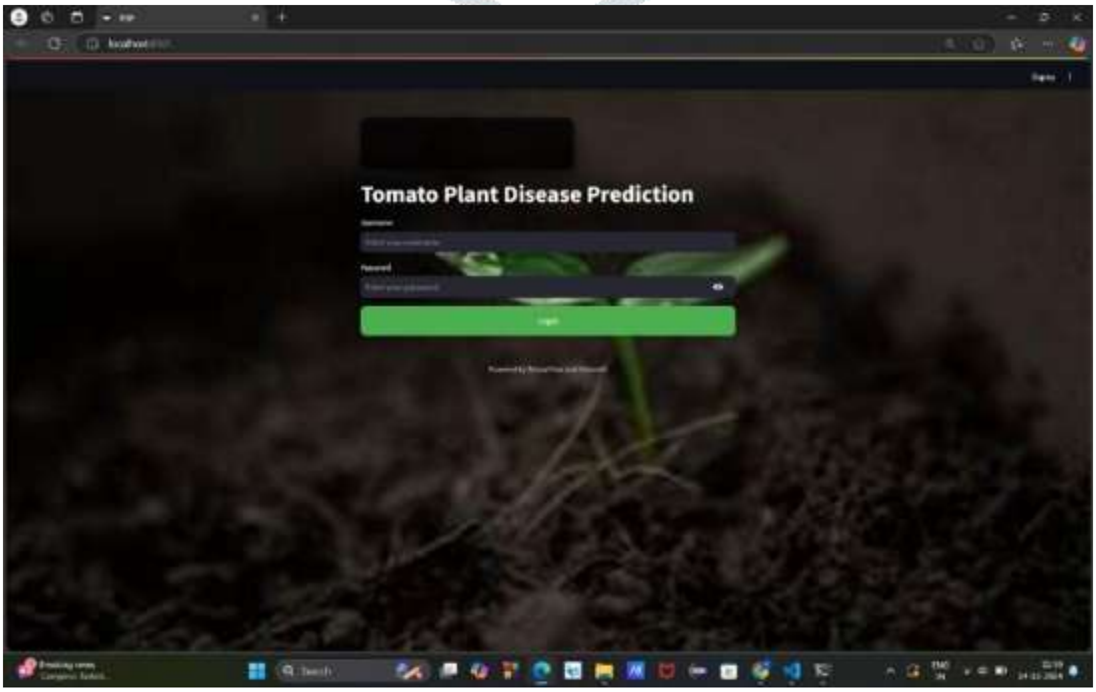

Fig.1 Home page of the website.

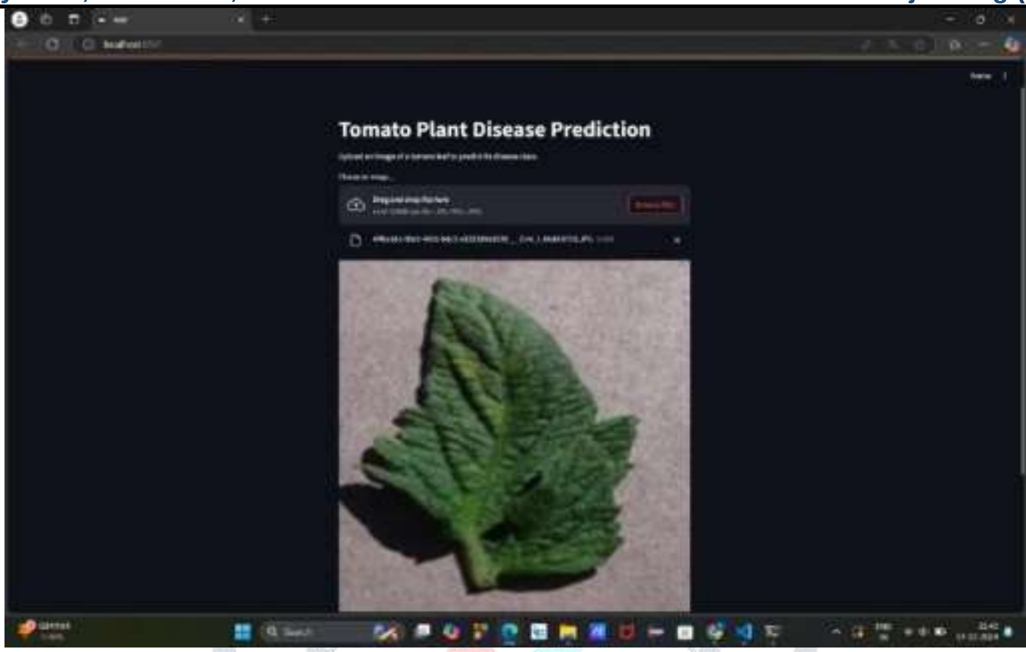

Fig.2 Displaying result for the given tomato leaf image.

VI. CONCLUSION

The development of a tomato leaf disease prediction system using deep learning introduces a modern and efficient approach to agricultural disease management. By integrating image classification with an accessible web interface, the system enables early and accurate detection of diseases such as Early Blight, Septoria Leaf Spot, and Leaf Mold. This addresses key limitations of manual inspection, which is often slow, inaccurate, and inaccessible to rural farmers.

The intuitive interface allows users to upload leaf images and receive instant diagnostic feedback along with recommended treatment suggestions. This real-time capability enhances user engagement and empowers farmers to take timely action. The system reduces dependency on expert consultation and facilitates rapid decision-making at the field level.

Experimental results confirm the system's high accuracy in disease classification, relevance of recommendations, and overall user satisfaction. With further refinement in the model and inclusion of more diverse datasets, the system can become even more robust and generalizable to a wider range of crop conditions.

In summary, the project demonstrates the effective application of deep learning in agriculture, offering a scalable and impactful solution to plant health monitoring. Future enhancements may include support for additional crops, offline mobile app integration, and the use of drones or sensors for large-scale field deployment.

The system's architecture supports continued improvement and adaptation. With advancements in machine learning and edge computing, real-time prediction on mobile devices could become feasible. Features like geotagging, multilingual support, and integration with local agricultural databases could further expand usability and impact.

Overall, this project represents a meaningful step forward in precision agriculture, aiming not only to boost productivity but also to make disease management more accessible, intelligent, and farmer-friendly.

REFERENCES

[1] J. Chen, J., & Wang, Q. (2022). Tomato Leaf Disease Detection Using AlexNet Convolutional Neural Network. International Journal of Advanced Computer Science and Applications, 13(5), 123-130. https://doi.org/10.14569/IJACSA.2022.0130516

[2] Li, X., & Zhang, Y. (2023). Lightweight Vision Transformer Model for Effective Diagnosis of Leaf Diseases in Sugarcane Plants. Computers and Electronics in Agriculture, 198, 107006. https://doi.org/10.1016/j.compag.2022.107006

[3] Thai, T. H., & Nguyen, H. T. (2023). Vision Transformer Model for Accurate Detection of Cassava Leaf Diseases. IEEE Access, 11, 45678-45689. https://doi.org/10.1109/ACCESS.2023.3245678

[4] Yu, L., & Chen, S. (2023). Inception Convolutional Vision Transformers for Effective Identification of Plant Diseases. Pattern Recognition Letters, 165, 45-52. https://doi.org/10.1016/j.patrec.2023.01.012

[5] Arshad, M., & Khan, S. (2023). End-to-End Hybrid Model for Accurate Prediction of Potato Leaf Diseases. Expert Systems with Applications, 213, 118897. https://doi.org/10.1016/j.eswa.2023.118897