Project Documentation: Automated Detection of Tomato Plant Diseases Using Traditional Machine Learning Techniques

Zemedkun Abebe, Dagim Melkie, Wasihun Belay

A Project Documentation Submitted to AAIT

In Partial Fulfillment of the Requirements for Computer Vision Course

1 Introduction

1.1 Motivation

Early and accurate detection of tomato plant diseases is crucial for minimizing yield losses and boosting agricultural productivity. However, traditional methods rely on visual inspection by human experts, a process that can be time-consuming, subjective, and prone to errors. This project aims to develop a reliable and efficient system for automated disease detection in tomato plants using machine learning techniques.

Tomato is a globally significant crop, and is unfortunately susceptible to various diseases caused by fungi, bacteria, viruses, and environmental factors. These diseases can significantly reduce fruit quality and yield, causing substantial economic losses for farmers. Timely identification and intervention are key to mitigating these losses and optimizing production.

While visual inspection by experienced personnel remains the primary method for disease detection, its limitations are evident. Human experts' assessments can be labor-intensive, time-consuming, and susceptible to individual variations and fatigue. Moreover, qualified experts may not be readily available in remote or resource-constrained agricultural settings. Therefore, an automated, objective, and readily deployable system for tomato disease detection is highly desirable.

1.2 Related Work and Contributions

Several recent studies have achieved remarkable results in automated tomato disease detection using deep learning approaches, particularly convolutional neural networks (CNNs). These methods often boast high accuracy but can be computationally expensive and require large datasets for optimal performance. Additionally, their complex nature can hinder interpretability and debugging.

Focusing on traditional machine learning (ML) offers an alternative with some key advantages:

- **Interpretability:** Established algorithms like PCA, LDA, and Random Forest provide clearer insights into the features and relationships influencing disease classification, allowing for better understanding of the model's decision-making process.
- **Computational efficiency:** Traditional ML techniques typically require less computational resources and training time compared to deep learning models, making them potentially more suitable for deployment on resource-constrained devices.
- **Potential for Generalization:** By carefully selecting and combining features, traditional ML models can exhibit better generalizability on unseen data, reducing the risk of overfitting and improving real-world applicability.

Our Contributions:

This project bridges the gap between advanced deep learning and interpretable ML by proposing a novel approach focusing on the following key contributions:

- **Feature Selection and Combination:** We employ techniques like Random Forest feature importance and ANOVA to identify the most relevant and discriminatory features from a combination of color, shape, and texture features. This reduces model complexity, enhances generalizability, and improves interpretability.

- **Robust Ensemble Learning:** We leverage the strengths of different traditional ML algorithms by employing an ensemble method like Random Forest. This approach enhances robustness, prediction accuracy, and stability compared to single-algorithm methods.

Recent Deep Learning Research

While focusing on traditional ML, we acknowledge the valuable contributions of recent deep learning research in this field. Studies like

1. Shoaib et al. (2022)[4] proposed a unique two-stage approach using deep learning for both segmentation and classification of tomato leaf images. Their InceptionNet model achieved impressive accuracy in identifying diseased regions and classifying them into specific disease categories.

2. Wagle (2021)[5] focused on a deep learning-based model for automatic classification and validation of tomato leaf diseases. Their system demonstrated promising results in accurately identifying various diseases while offering validation capabilities for disease diagnosis.

3. Fuentes et al. (2017)[1] developed a robust deep learning detector capable of real-time recognition of both tomato plant diseases and pests. This efficient model showcased potential for practical use in agricultural settings, enabling quick and accurate identification of threats.

4. Habiba and Islam (2021)[2] explored the use of deep learning classifiers for tomato disease classification directly from leaf images. Their study highlighted the feasibility of deploying such models for disease detection without the need for pre-segmentation, simplifying the overall process.

5. Nawaz et al. (2022)[3] addressed the challenge of precise disease localization along with classification by proposing a deep learning approach. Their model exhibited promising results in pinpointing and identifying diseased areas within tomato leaves, offering valuable information for targeted treatment strategies.

These studies showcase the diverse applications of deep learning in tomato disease detection, opening the door for more efficient, accurate, and automated solutions in the field of agriculture.

It's important to note that these are just a few examples, and a plethora of other exciting research efforts are also contributing to this vital field. As deep learning technology continues to evolve, we can expect even more advancements in automated disease detection with the potential to revolutionize how we approach plant health management in the future.

have significantly advanced the field. Our work aims to complement these efforts by exploring the potential of traditional ML as a viable and interpretable alternative for automated tomato disease detection, offering advantages in efficiency and interpretability while potentially achieving competitive performance.

2 Problem Statement

This project addresses the challenge of accurately detecting tomato plant diseases from digital images using traditional machine learning algorithms. Our aim is to develop a robust system that can effectively classify healthy and diseased leaves with high accuracy, empowering farmers to make informed decisions and implement timely disease management strategies.

3 Methodology

Our methodology for automated tomato disease detection encompasses the following stages:

3.1 Dataset Description and Preparation

The dataset used in this project is the PlantVillage dataset, which is a publicly available dataset of plant leaf images with various diseases. The dataset contains 54,306 images of 14 crop species and 26 diseases, as well as healthy images. The images are in RGB format, with a resolution of 256 x 256 pixels. The images are divided into 38 classes, each representing a crop-disease pair or a healthy crop. The dataset is available in two versions: a segmented version, where the background of the images is removed, and a non-segmented version, where the background of the images is retained.

For this project, we only use the images of tomato leaves, which constitute a subset of the PlantVillage dataset. The tomato leaf subset contains 18,103 images, belonging to 10 classes: 9 diseases and 1 healthy. The 9 diseases are: bacterial spot, early blight, late blight, leaf mold, mosaic virus, septoria leaf spot, spider mites, target spot, and yellow leaf curl virus. The distribution of the images among the 10 classes is shown in Table 1.

Class	**Number of Images**
Bacterial Spot	**2127**
Early Blight	**1000**
Late Blight	**1909**
Leaf Mold	**952**
Mosaic Virus	**373**
Septoria Leaf Spot	**1771**
Spider Mites	**1676**
Target Spot	**1404**
Yellow Leaf Curl Virus	**5357**
Healthy	**1734**
Total	**18103**

Table 1: Distribution of images among the 10 classes of tomato leaf subset.

We use the non-segmented version of the dataset, as we want to test the robustness of our method to different backgrounds and noises. We split the dataset into three subsets: training, validation, and testing, with a ratio of 70:15:15. The training subset is used for training the models, the validation subset is used for tuning the hyperparameters, and the testing subset is used for evaluating the performance. The distribution of the images among the three subsets is shown in Table 2.

3.2 Preprocessing Techniques

Before extracting features from the images, we apply some preprocessing techniques to enhance the quality and reduce the noise of the images. The preprocessing techniques we use are as follows

- **Resizing:** We resize the images to a fixed size of 128 x 128 pixels, to reduce the computational cost and standardize the input size for the feature extraction and classification methods.

Subset	Number of Images
Training	**12672**
Validation	**2716**
Testing	**2715**
Total	**18103**

Table 2: Distribution of images among the three subsets of the tomato leaf subset.

- **Color space conversion:** We convert the images from RGB to HSV color space, as HSV is more suitable for capturing the color information of the plant leaves. HSV stands for hue, saturation, and value, which represent the color, intensity, and brightness of the pixels, respectively.
- **Grayscale conversion and noise reduction:** We convert the images to grayscale, as we only need the intensity information for the texture feature extraction. We also apply a Gaussian blur filter to the grayscale images, to smooth out the edges and reduce the noise.
- **Edge detection and thresholding:** We use the Canny edge detector to detect the edges of the plant leaves, which are important for the shape feature extraction. We also apply an adaptive thresholding method to binarize the edge images, to separate the foreground (leaf) from the background.
- **Contour extraction and shape analysis:** We use the find Contours function of OpenCV to extract the contours of the plant leaves from the binary edge images. We then calculate the area and perimeter of the largest contour, which represent the shape features of the plant leaves.

3.3 Feature Extraction and Selection

Following the preprocessing of images, a comprehensive set of features is extracted to enhance the discriminative capabilities for detecting tomato diseases. The chosen features encompass aspects of color, shape, and texture, carefully selected based on their relevance to the task. The extracted features are detailed below:

3.3.1 Color Features:

The mean and standard deviation of each color channel (hue, saturation, and value) are extracted from the HSV images. Additionally, the ratio of green pixels to non-green pixels is computed. These color features encapsulate the color information present in plant leaves, offering insights into the potential presence or absence of diseases.

3.3.2 Shape Features:

Shape-related characteristics are derived from the binary edge images. The area and perimeter of the largest contour are extracted, along with circularity and solidity measures. Circularity is defined as the ratio of the area to the square of the perimeter, while solidity is defined as the ratio of the area to the convex hull area of the contour. These features capture the shape information of plant leaves, reflecting potential deformations or damage caused by diseases.

3.3.3 Texture Features:

Utilizing the Grey Level Co-occurrence Matrix (GLCM) method, texture features are extracted from grayscale images. This includes contrast, dissimilarity, and homogeneity, which provide insights into the textural patterns or variations in plant leaves. These texture features contribute valuable information for discerning characteristics indicative of tomato diseases.

This meticulous selection and extraction process ensures a comprehensive representation of key attributes, laying the foundation for effective detection and classification of tomato diseases based on the extracted features.

4 Results and Findings

Our investigation into tomato plant disease detection using Random Forest classifiers with various feature extraction and selection techniques yielded insightful results:

4.1 Random Forest with PCA:

- Achieved an impressive accuracy of 85.46% on the testing set.
- Demonstrates the efficacy of Principal Component Analysis (PCA) in reducing dimensionality while preserving crucial information for accurate classification.

4.2 Random Forest with LDA:

- Attained an accuracy of 55.79% on the testing set.
- Reveals that Linear Discriminant Analysis (LDA) might not be as effective as PCA in capturing relevant features for distinguishing between healthy and diseased tomato leaves.

4.3 Random Forest with Selected Features:

- Obtained an accuracy of 72.10% on the testing set.
- Indicates that a careful selection of features can contribute to the model's performance, striking a balance between informativeness and simplicity.

4.4 Random Forest with ANOVA:

- Showcased an accuracy of 77.86% on the testing set.
- Suggests that features selected through Analysis of Variance (ANOVA) play a significant role in enhancing the model's discriminatory power.

These findings underscore the importance of feature extraction and selection techniques in influencing the performance of machine learning models for tomato disease classification. The notable accuracy variations among the different approaches emphasize the need for a thoughtful selection of methods tailored to the specific characteristics of the dataset. Overall, our results contribute valuable insights to the ongoing research in automated plant disease detection, providing a foundation for informed decisions in agricultural practices.

5 System Architecture

The system architecture is designed with three core components, each playing a vital role in the overall functionality:

5.1 Preprocessing Module:

- Responsible for preparing the input image for subsequent processing.
- Key tasks include resizing the image, converting it to grayscale, and performing normalization.
- Ensures that the input image is appropriately formatted and scaled for feature extraction and classification.

5.2 Feature Extraction Module:

- Extracts pertinent features from the preprocessed image, focusing on both texture and color attributes.
- Utilizes advanced techniques such as grey level co-occurrence matrix (GLCM) for texture analysis and color channel statistics for capturing color information.
- Ensures that the extracted features encapsulate relevant characteristics crucial for accurate disease classification.

5.3 Classification Module:

- Leverages the trained Random Forest Classifier model to make predictions regarding the health status of the input image.
- Processes the extracted features through the trained model, utilizing the accumulated knowledge to distinguish between healthy and diseased tomato leaves.
- The culmination of the classification process provides valuable insights into potential plant diseases.

This modular architecture promotes a systematic and efficient workflow. It allows for flexibility in upgrading or replacing individual components without affecting the entire system. The clear delineation of responsibilities enhances maintainability, scalability, and the overall robustness of the automated tomato plant disease detection system.

6. Implementation Details

- **Programming Language:** Python
- **Libraries Used:** OpenCV, NumPy, Pandas, Scikit-learn, Streamlit
- **Machine Learning Framework:** Scikit-learn
- **Hardware Platform:** Standard personal computer

7. Model and User Interface

- The trained Random Forest Classifier model is saved for future use.
- A user interface was developed using the Streamlit library, allowing users to upload images and receive disease predictions.

6 Conclusion

In conclusion, this project successfully addresses the task of automated detection of tomato plant diseases using traditional machine learning techniques. The implementation leverages a Random Forest Classifier model trained on a diverse set of features extracted from tomato leaf images. The feature extraction process includes shape, color, and texture characteristics, providing a comprehensive understanding of the plant's health.

The user interface, built with the Streamlit library, enhances accessibility by allowing users to conveniently upload images and obtain predictions regarding potential diseases affecting tomato plants. This user-friendly interface facilitates the broader applicability of the model by making it accessible to individuals, farmers, or professionals interested in monitoring and diagnosing tomato plant health.

The integration of Python, OpenCV, NumPy, Pandas, Scikit-learn, and Streamlit ensures a robust and efficient implementation. The codebase is well-documented and modular, making it adaptable for future improvements or integration into larger agricultural systems.

As a result, this project contributes to the advancement of automated plant disease detection, offering a reliable tool for the early identification and management of tomato plant diseases. The combination of a powerful machine learning model and an intuitive user interface creates a valuable resource for agricultural practitioners, fostering sustainable crop management practices and ultimately promoting global food security.

7 Future Work

While the current implementation has achieved notable success in automating the detection of tomato plant diseases, there are several avenues for future work and enhancements. Some potential areas for improvement and expansion include:

1. **Model Fine-Tuning:** Further refinement of the Random Forest model could be explored by fine-tuning hyperparameters or experimenting with alternative machine learning algorithms. This may lead to improved accuracy and robustness across a wider range of scenarios.
2. **Dataset Augmentation:** Enlarging the training dataset through augmentation techniques, such as rotation, flipping, or changes in lighting conditions, can enhance the model's ability to generalize to diverse environmental conditions.
3. **Integration of Deep Learning:** Exploring the integration of deep learning architectures, such as convolutional neural networks (CNNs), may provide opportunities for more complex feature learning and improved accuracy in disease detection.

4. **Real-time Monitoring:** Adapting the system for real-time monitoring of tomato plants in agricultural settings could be valuable. This would involve optimizing the model for quick predictions and integrating it into a live monitoring system.

5. **Disease Classification Expansion:** Extend the model to detect and classify a broader range of tomato plant diseases. This could involve incorporating additional classes and training the model on a more extensive dataset with diverse disease manifestations.

6. **Mobile Application Development:** Creating a mobile application for disease detection could increase accessibility for farmers and enthusiasts. This would involve adapting the user interface for mobile platforms and ensuring seamless integration with smartphone cameras.

7. **Crowdsourced Data Collection:** Implementing a system for collecting and incorporating crowdsourced data on tomato plant diseases could contribute to a more comprehensive and up-to-date dataset, improving the model's generalization capabilities.

8. **Collaboration with Agricultural Experts:** Collaborating with agricultural experts and researchers to validate the model's predictions and gather domain-specific insights can enhance the system's reliability and relevance in real-world agricultural practices.

9. **Localization and Multilingual Support:** Tailoring the system to specific geographical regions and providing multilingual support could make the tool more accessible to a global audience, considering the diverse farming practices and languages used in agriculture.

10. **Continuous Maintenance and Updates:** Regularly updating the model with new data and features, along with maintaining compatibility with the latest libraries and frameworks, ensures the system remains effective and reliable over time.

Addressing these areas of future work will contribute to the ongoing development of an advanced, adaptable, and impactful tool for tomato plant disease detection in agricultural settings.

References

[1] Alvaro Fuentes, Sook Yoon, Sang Cheol Kim, and Dong Sun Park. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9):2022, 2017.

[2] Sultana Umme Habiba and Md Khairul Islam. Tomato plant diseases classification using deep learning based classifier from leaves images. In *2021 international conference on information and communication technology for sustainable development (ICICT4SD)*, pages 82–86. IEEE, 2021.

[3] M Nawaz, T Nazir, A Javed, M Masood, J Rashid, J Kim, and A Hussain. A robust deep learning approach for tomato plant leaf disease localization and classification. sci. rep. 12, 18568 (2022).

[4] Muhammad Shoaib, Tariq Hussain, Babar Shah, Ihsan Ullah, Sayyed Mudassar Shah, Farman Ali, and Sang Hyun Park. Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Frontiers in Plant Science*, 13:1031748, 2022.

[5] Shivali Amit Wagle et al. A deep learning-based approach in classification and validation of tomato leaf disease. *Traitement du signal*, 38(3), 2021.