TOMATO LEAF DISEASE DETECTION USING VGG16

Mohammed Sraj GSR/5161/16
Addis Ababa Institute of Technology

ABSTRACT

This project focuses on detecting tomato leaf diseases using deep learning, specifically leveraging the VGG16 convolutional neural network (CNN) architecture. The primary motivation behind this project is to aid in the early detection and management of plant diseases, which can significantly improve agricultural productivity and reduce crop losses. The project utilizes a tomato leaf disease dataset from Kaggle and employs transfer learning techniques to enhance the model's performance.

1 INTRODUCTION

This project focuses on detecting tomato leaf diseases using deep learning, specifically leveraging the VGG16 convolutional neural network (CNN) architecture. The primary motivation behind this project is to aid in the early detection and management of plant diseases, which can significantly improve agricultural productivity and reduce crop losses. The project utilizes a tomato leaf disease dataset from Kaggle and employs transfer learning techniques to enhance the model's performance.

2 PROBLEM STATEMENT

The problem addressed by this project is the accurate detection and classification of various diseases affecting tomato leaves. Traditional methods of disease detection are often time-consuming and require expert knowledge. This project aims to develop a more efficient and automated solution using advanced deep learning techniques.

3 METHODOLOGY

The methodology involves the following steps:

- **Data Collection**: The dataset was sourced from Kaggle, consisting of images of tomato leaves categorized by disease type.
- **Model Training**: The VGG16 architecture was used as the base model. Transfer learning techniques were employed to fine-tune the model for our specific dataset.
- **Model Saving**: The trained model was saved in both HDF5 and Keras formats to ensure compatibility and ease of deployment.

The experimental setup was conducted on Google Colab, leveraging its computational resources. Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the model's performance.

4 PROPOSED SOLUTION

The proposed solution involves a Flask-based web application that allows users to upload images of tomato leaves for disease detection. The main components of the solution include:

- **Pre-trained VGG16 Model**: Fine-tuned using the tomato leaf disease dataset.
- **Flask Web Application**: Interfaces with the model to provide real-time disease detection.

- **GitHub Repository**: Maintains the project code and documentation for version control and collaboration.

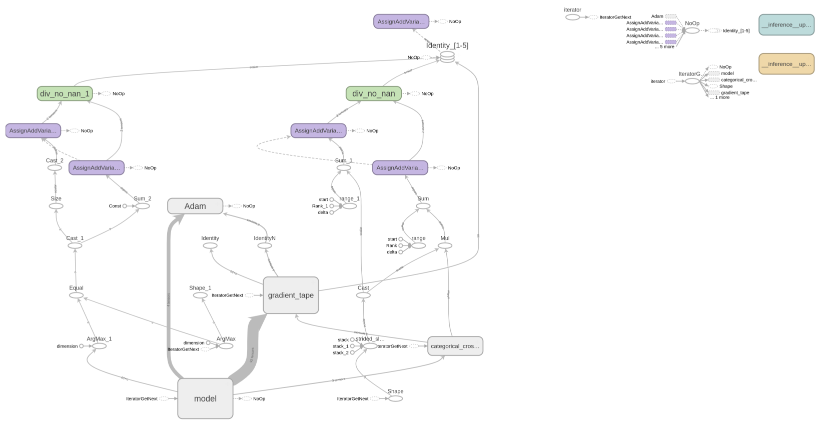


Figure 1: VGG16 Model

5 System Architecture

The system architecture consists of the following components:

- **Frontend**: A web interface developed using HTML, CSS, and JavaScript.
- **Backend**: A Flask server that handles image uploads and interacts with the pre-trained VGG16 model.
- **Model**: The VGG16 model fine-tuned with the tomato leaf disease dataset.
- **Database**: Stores user inputs and results for analysis and feedback.

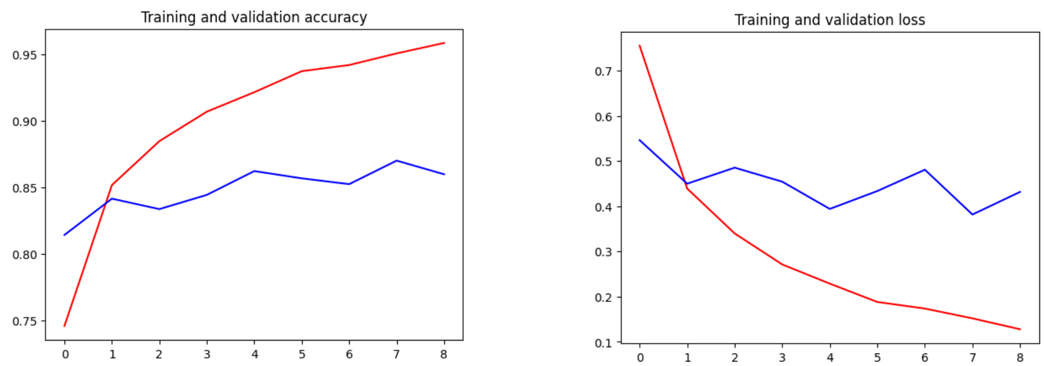
6 Implementation Details

- **Hardware**: Developed and tested on a laptop with a Core i5 processor running Ubuntu 24.
- **Software**: Key software components include Flask, TensorFlow, Keras, and Nginx.
- **Training Environment**: Google Colab was used for model training, utilizing its GPU capabilities.

7 Results

The model achieved high accuracy in detecting and classifying tomato leaf diseases. Detailed performance metrics are as follows:

- **Accuracy**: Achieved a high accuracy 86% rate on the test dataset.
- **Precision, Recall, F1-Score**: Demonstrated robust performance across various disease categories.



(a) Training and validation accuracy

(b) Training and validation loss

Figure 2: Training and validation metrics

Limitations include potential variability in real-world conditions and the need for further training with more diverse datasets. Future work could explore the integration of more advanced architectures and larger datasets.

8 Deployment

The deployment was carried out on the Digital Ocean platform, where a droplet was created, and necessary dependencies were installed. An Nginx proxy server was configured to manage incoming traffic, ensuring efficient and reliable service delivery. The Flask web application was successfully deployed and made accessible for user interactions.

9 Conclusion

This project successfully demonstrates the application of deep learning in detecting tomato leaf diseases. The use of transfer learning with the VGG16 architecture proved effective, and the developed web application provides a practical tool for disease detection. Future work could further enhance the model's robustness and explore additional plant disease datasets.

10 Source code and webapp links

`https://github.com/berito/cv_project_2.git`
`http://188.166.153.41/`

11 References

- Kaggle. Tomato Leaf Disease Dataset.
- TensorFlow and Keras Documentation.
- Flask Documentation.
- Digital Ocean Deployment Guides.