EUROPEAN UNIVERSITY OF LEFKE

FACULTY OF ENGINEERING

Computer Engineering Department

Graduation Project I

Digital Farmers:

Harvesting Insights to Tackle Leaf Diseases

Seward Richard Mupereri

20140175

This project uses machine learning and cloud technology, to quickly spot early signs of diseases on tomato plants. The main idea is to create a useful tool for farmers that allows them to catch plant problems early. This will help them grow better crops and use fewer pesticides. The project's main goals are to show how technology can change farming, with a focus on accurate disease detection and an easy-to-use interface.

Supervisor

Dr. Ferhun Yorgancıoğlu

Publish Date

25 - 12 – 2023

Table Of Contents

1. Introduction

1.1. Problem definition

Tomato farming faces a significant challenge in the early detection of diseases on tomato plants, resulting in economic and environmental consequences. This critical issue is primarily due to the fact that diseases often remain undetected until they have already caused substantial damage to crops. As a result, farmers face the economic pain of reduced yields and the environmental impact of excess pesticide use. Our project seeks to provide an innovative solution to this problem.

Our approach is grounded in the integration of machine learning and cloud computing, to revolutionize disease detection in the context of tomato farming. The project follows a streamlined process:

I. **Data Cleaning and Preprocessing:** The first step is to gather a comprehensive dataset that encompasses healthy leaves as well as affected leaves of the selected diseases to ensure a well-rounded representation of the world. This dataset is then cleaned and pre-processed to standardize and enhance the dataset, optimising it for machine learning analysis. This step will be completed using Jupyter Notebook on VS Code using Python (Keras API).

II. **Model Building and Training:** The project uses Convolutional Neural Networks (CNNs) as the machine learning model of choice as they are the industry standard for image analysis. The training process involves feeding the pre-processed images through the selected model. This iterative process adjusts the model's internal weights, optimizing it for accurate classification of healthy and diseased tomato leaves. The machine learning framework to be used to facilitate this process is TensorFlow.

III. **Model Deployment**: The model is the deployed onto Google cloud using tf serving to allow users to be able to interact with the model on the web.

IV. **Web-Based Application**: To bring the benefits of our machine learning system to farmers, a user-friendly web-based application is developed. This application allows users, including those with limited technical expertise, to easily upload images of their tomato plants and receive instant disease diagnosis.

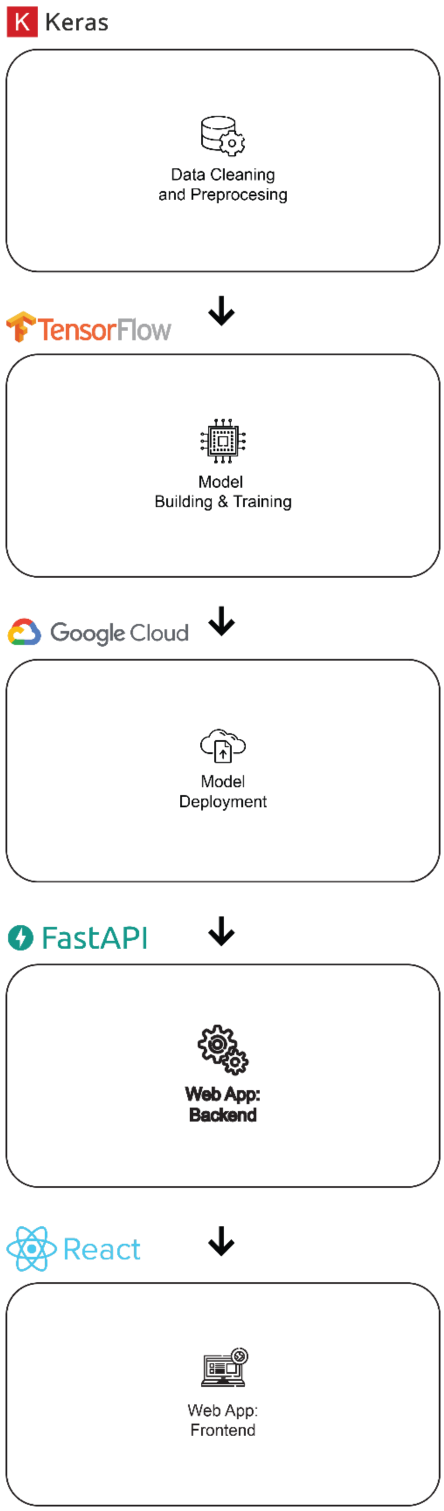

Figure 1: Streamlined project process

To ensure the success of the project, some challenges that have hindered progress in the field of disease detection in agriculture are considered:

- **Data Quality**: Previous projects often struggled to obtain a diverse and representative dataset. This project will use open-source datasets that have been properly prepared and sourced.
- **Deployment Complexity**: Setting up a reliable deployment infrastructure can get complex that's why this project will be deployed on the Google Cloud Platform to ensure ensuring high availability, scalability, and reliable performance.

1.2 Goals

Our project's primary goals are:

- **Early Disease Detection**: Easy and accurate detection of tomato leaf diseases to allow farmers to intervene early.
- **Sustainability**: Promoting environmentally friendly farming practices by reducing reliance on pesticides.
- **User-Friendly Interface**: Ensuring that technology solutions are accessible to all, regardless of technical expertise.

2. Literature Survey

This project includes a highly accurate trained machine learning model for tomato disease detection, a web-based application and project documentation. Below are some similar projects:

2.1. Similar Project 1: Plant Disease Detection Using Image Processing and Machine Learning

The first similar project is titled "Plant Disease Detection Using Image Processing and Machine Learning," and it was developed by a team at the Vishwakarma Institute of Technology in Pune, India. The project also uses computer vision and machine learning to detect plant disease on plant leaves. [1]

Comparison:

1. **Scope:** This similar project aims to detect disease on 5 different types of plant leaves while this project focuses on the tomato which might allow me to get higher accuracy as there is only one plant to focus on and fine tune the model for.

2. **User Interface**: This similar used a flask-based web application while this project will create a React based web application.

In conclusion, while both projects aim to detect plant diseases using machine learning and computer vision, this project's specific focus on just tomato leaves with a user-friendly web interface sets it apart from this project.

2.2. Similar Project 2: Plant-Disease-Detection

The project is titled "Plant-Disease-Detection" and it is hosted on GitHub. This similar project aims to predict plant diseases based on leaf images using a convolutional neural network implemented in PyTorch. The system classifies leaf images into 39 different categories. [2]

Comparison:

1. **Scope:** This similar project covers a larger scope than this project as it categorizes images into 39 different categories while this project will not have that many categories to allow a focus on a few diseases for higher accuracy when predicting.

2. **Methodology:** This similar project used PyTorch as its platform for the machine learning while this project will be using TensorFlow for model training and fine tuning.

In conclusion, while both projects aim to detect plant diseases using machine learning and computer vision, this project's specific focus on just tomato leaves and the use of TensorFlow sets it apart from this similar one.

2.3. Similar Project 3: Farmassist

This similar project, titled "Farmassist," is hosted on GitHub. This similar project is a smart farming app for IoT and AI-powered plant disease detection. This similar project uses Flutter for android development and Firebase for the backend. The system can detect plant diseases uses images. This similar project stores farm management data for planting and harvesting as well as an IoT monitoring system for plants. [3]

Comparison:

1. **Scope:** This similar project covers disease detection for various plants among other functions while this project focuses on just the tomato plant.

2. **Platform**: This similar project is development for the mobile platform while this project will be developed for the web.

In conclusion, while both projects aim to detect plant diseases using machine learning and computer vision, this project's focus on just the tomato plant makes it stand out from this similar one and the fact that this similar project was developed into a mobile application while this project will be a web application.

3. Background Information

Below is a comprehensive range of software and hardware to be used from the first stage of data collection to the deployment of the web-based application for tomato disease detection in forming.

3.1 Required software

1. **Data Collection:**
    - **Kaggle:** The data used in this project will be collected from Kaggle datasets which provides comprehensive and freely available diverse datasets.
2. **Programming Languages:**
    - **Python:** The model building and training and other stages of the development of the project will be written in Python.
    - **JavaScript:** This language will be used during the web development stage with the React library.
    - **HTML & CSS:** This language will be used during the web development stage with the React library.
3. **Model Training and Tuning:**
    - **TensorFlow (Python):** In this project TensorFlow will be the framework used for model training and fine-tuning. The focus will be Convolutional Neural Networks (CNNs) for image classification. The main libraries to be used are the Keras Library
4. **Model Deployment:**
    - **Google Cloud Platform (GCP):** The machine learning model will be deployed on GCP, assuring scalability, availability, and reliable performance.
5. **Web Development:**
    - **FastAPI (Python):** The backend of the web-based application will be developed FastAPI web framework to leverage the performance and efficiency of the language.

- **React:** To develop the frontend of the web-based application the React JavaScript library will be used as it will allow for a user-friendly, responsive and intuitive interface.

6. **Version Control:**

   - **Git:** In this project, Git is used for version control and code management.

3.2 Other software

1. **Vector Graphic Editing and Design Software:**

   - **Adobe Illustrator:** In addition, Adobe Illustrator will be utilized for creating vector files, ensuring compatibility and scalability for certain graphic elements of the user interface and project documentation.

2. **Code Editor:**

   - **Visual Studio Code:** This will be the code editor used for code writing, debugging and project code management efficiently.

   - **Jupyter Notebook:** This editor will mostly be used for the model training and fine tuning.

4. Modules

This section outlines the essential components of the project. It is structured into distinct phases to streamline the workflow effectively. This approach helps to effectively manage and organise tasks in the project.

4.1. Data Cleaning and Preprocessing Module

- **Description**: This module begins with compiling a comprehensive dataset of healthy and affected tomato leaves. This dataset then undergoes cleaning and preprocessing in Jupyter Notebook on VS Code using Python (Keras API) to standardize and enhance it for machine learning analysis.
- **Tasks:**
  - Gather diverse images representing healthy and affected tomato leaves from Kaggle Datasets.
  - Perform data cleaning and standardization processes like normalization and enhancement.
- **Importance:** Preparing a refined dataset lays the foundation for accurate machine learning analysis.

4.2. Model Building and Training Module

- **Description:** This module relies on Convolutional Neural Networks (CNNs) as the industry-standard model for image analysis. The training process involves feeding pre-processed images through the chosen model, iteratively adjusting its internal weights to optimize accurate classification of healthy and diseased tomato leaves using TensorFlow.
- **Tasks:**
    - Utilize CNNs for robust image analysis.
    - Train the model iteratively to enhance accuracy in disease classification.
- **Importance:** The iterative training process refines the model for precise leaf classification.

4.3. Model Deployment Module

- **Description:** Following training, the model is deployed on Google Cloud using TensorFlow serving, enabling user interaction via the web interface.
- **Tasks:**
    - Deploy the trained model on Google Cloud infrastructure.
    - Enable user interaction through a web-based interface.
- **Importance:** Deploying the model makes the technology accessible to users via the web.

4.4. Web-Based Application Module

- **Description:** The project includes developing a user-friendly web-based application that facilitates easy image uploads for instant disease diagnosis, catering to users with varying technical expertise.
- **Tasks:**
    - Use frontend technologies like React JavaScript library for the user interface.
    - Develop the backend using FastAPI a Python web-framework for seamless functionality.
- **Importance:** The application empowers farmers by offering a user-friendly interface for disease diagnosis.

5. Risk Analysis

5.1. Data Quality and Diversity:

Risk: The quality and diversity of the dataset plays a huge role in the success of any machine learning project. If the dataset does not accurately represent real-word conditions, the model may produce inaccurate predictions.

Mitigation: Use a reputable and well organised dataset from open-source dataset repositories (in this case Kaggle) that has been well refined.

5.2. Model Overfitting:

Risk: Convolutional Neural Networks (CNNs) can be prone to overfitting, where the model performs well on the training data but poorly on new, unseen data.

Mitigation: Implement extensive hyperparameter tuning and employ techniques like dropout layers to prevent overfitting [3]. If possible, visit a tomato farm and get as many images to test on, that have been categorised well by well-trained humans and test the model against that.

5.3. Academic Evaluation and Grading:

Risk: The project's success in terms of academic evaluation and grading may heavily rely on the subjective assessment criteria and preferences of the instructors or evaluators.

Mitigation: Clearly communicate with instructors or evaluators to understand their expectations and criteria for evaluation. Ensure that the project documentation is comprehensive and aligned with the evaluation criteria.

5.4. Time Constraints:

Risk: Balancing the project's development and deployment within the limited time frame of a semester can be challenging. Delays or difficulties in project execution may impact the quality of the final submission.

Mitigation: Use Notion to create a detailed project timeline with milestones, and regularly review and adjust the plan as needed. Prioritize essential project components to ensure that critical aspects are completed within the allocated time.

6. Ethics

This section highlights the ethical principles and concerns that will be considered during the development of this project.

- **Data Privacy and Consent**

  To ensure that all Data privacy and Consent requirements would be met for our dataset, the project uses open-source data that has been peer reviewed and rated on Kaggle. For any data that will be collected in the real-world the project will ensure that all parties involved have given informed consent. [4]

- **Bias and Fairness**

  The use of Convolutional Neural Networks (CNNs) raises concerns about bias and fairness in decision-making and predictions. This project takes measures to minimize bias in disease detection by ensuring that the datasets used to train the models are diverse and represent real-world conditions. [5]

- **Environmental Impact**

  This project aims to promote sustainable farming practices by reducing the reliance of pesticides as infected plants are caught early and destroyed before there is a spread of pests or an illness. Reducing pesticide uses has a positive impact on the environment as it decreases chemical runoff that can damage ecosystems. This project aims to minimize the environmental impact of agriculture. [6]

- **Transparency and Accountability**

  Transparency is a key ethical principle in this project. This project aims to provide clear and understandable disease diagnosis results to users. The limitation of the model and its potential for false positives and negatives is acknowledged. Transparency and accountability are essential in maintaining user trust. [7]

7. Conclusion

The development of this project aims to improve efficiency in tomato farming in a sustainable manner but it will also provide several benefits for the developer. Additionally, this project will teach valuable insights for the developer’s future endeavours.

7.1. Benefits

a. Benefits to users:

1. **Improved Crop Management**: Farmers will benefit from early disease detection, leading to more effective crop management. Swift identification of diseased plants allows for timely intervention, potentially increasing crop yields and reducing the environmental impact of excessive pesticide use.

2. **Accessibility**: The user-friendly web-based application ensures that individuals with a wide range of technical expertise can easily upload images of their tomato plants and receive instant disease diagnosis.

3. **Sustainability**: By reducing the use of pesticides, the project promotes environmentally friendly farming practices. This contributes to the sustainability of agricultural practices and minimizes the harm caused by chemical runoff.

b. Benefits to me:

1. **Learning and Skill Development**: This project has provided me with an opportunity to learn and apply modern technologies such as machine learning, computer vision, and cloud computing. This experience will enhance my technical and problem-solving skills.

2. **Project Management:** The project's development has honed my project management skills, including task planning, time management, and resource allocation. These skills will be valuable in future endeavours.

3. **Contributing to Agriculture:** This project has allowed me to contribute to the agricultural industry by providing efficient farming practices that reduce the negative impact of agricultural activities.

Why did I choose this project?

The decision to work on this project was driven by the fact that I would get the opportunity to create an end-to-end project using technologies that are quite popular and in demand in the labour market would better prepare me for it after graduation. I would learn invaluable skills while possibly making an impact in the lives of farmers and the environment.

7.2. Future Works

Below are several future works that build upon the foundation established in this project:

- **Enhance Model Performance**: The goal is to provide very reliable disease detection results and to do this, continuous improvements to the machine learning model's accuracy by expanding the dataset and refining the model architecture will be necessary.
- **Expand Crop Coverage**: The project aims to eventually detect disease on a wider range of crops to allow for efficiency in the farming practices of even more plants.
- **Collaboration with Agricultural Experts**: Collaborate with agricultural experts and organizations to refine the system based on their insights and experiences in the field.

In conclusion, this project has been a step towards improving tomato farming practices by leveraging technology and promoting sustainable, technology driven agriculture. The project will provide valuable real-world skills for me and hopefully help improve the life of a tomato farmer.

8. References

[1] A. K. T. K. S. K. A. J. M. W. Pranesh Kulkarni, “Plant Disease Detection Using Image Processing and Machine Learning,” 22 November 2021. [Online]. Available: https://arxiv.org/abs/2106.10698. [Accessed 31 October 2023].

[2] A. M. Manthan Bhikadiya, “Plant-Disease-Detection,” GitHub, 13 July 2023. [Online]. Available: https://github.com/manthan89-py/Plant-Disease-Detection. [Accessed 13 October 2023].

[3] J. Brownlee, “Dropout Regularization in Deep Learning Models with Keras,” Machine Learning Mastery, 6 August 2022. [Online]. Available: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/. [Accessed 24 Dec 2023].

[4] S. Javaid, “Image Data Collection in 2023: What it is and Best Practices,” AIMultiple, 23 October 2023. [Online]. Available: https://research.aimultiple.com/image-data-collection/. [Accessed 4 November 2023].

[5] Rita, “The Impact Of Bias On Convolutional Neural Networks,” Surfactants, 27 November 2022. [Online]. Available: https://www.surfactants.net/the-impact-of-bias-on-convolutional-neural-networks/. [Accessed 4 November 2023].

[6] A. Pariona, “The Environmental Impact Of Pesticides,” WorldAtlas, 25 April 2017. [Online]. Available: https://www.worldatlas.com/articles/what-is-the-environmental-impact-of-pesticides.html. [Accessed 4 November 2023].

[7] V. Dignum, “The ART of AI — Accountability, Responsibility, Transparency,” Medium, 4 March 2018. [Online]. Available: https://medium.com/@virginiadignum/the-art-of-ai-accountability-responsibility-transparency-48666ec92ea5. [Accessed 4 November 2023].

[8] C. P. W. Ray Jasson, “Farmassist,” GitHub, 31 October 2021. [Online]. Available: https://farmassistx.github.io/farmassist/. [Accessed 31 October 2023].