Project Report Format

1. INTRODUCTION

1.1 Project Overview

Potato Plant Disease Classification with Machine Learning and Deep Learning

Potato plant diseases can cause significant crop losses, which can have a negative impact on farmers' incomes and the global food supply. Early detection and treatment of potato diseases is essential to minimize crop losses. Machine learning and deep learning can be used to develop accurate and efficient potato plant disease classification models.

This project aims to develop a potato plant disease classification model using the following tools: Matplotlib, NumPy, Notebook, TensorFlow-Addons, TensorFlow-Model-Optimization, and TensorFlow==2.5.

The proposed model will be trained on a dataset of potato leaf images labeled with their corresponding disease classes. Once the model is trained, it can be used to classify new potato leaf images into their respective disease classes.

The project will follow a systematic approach to develop and deploy the potato plant disease classification model. The following are the key steps involved in the project:

1. Data collection: A dataset of potato leaf images labeled with their corresponding disease classes will be collected.
2. Data preprocessing: The potato leaf images will be preprocessed to make them suitable for training a machine learning model. This may involve tasks such as resizing the images, converting the images to grayscale, and normalizing the image pixel values.
3. Model development: A potato plant disease classification model will be developed using TensorFlow. The model will be based on a convolutional neural network (CNN), which is a type of deep learning model that is well-suited for image classification tasks.

4. Model training: The developed model will be trained on the collected dataset. This process involves feeding the model the potato leaf images and their corresponding disease labels, and allowing the model to learn the patterns associated with each disease class.
5. Model evaluation: The trained model will be evaluated on a held-out test set to assess its performance. This will help to ensure that the model is able to generalize well to unseen data.
6. Model optimization: The trained model will be optimized for deployment using TensorFlow-Model-Optimization. This will involve tasks such as reducing the model size and improving the model's inference speed.
7. Web application development: A user-friendly web application will be developed to deploy the trained model. This web application will allow farmers and other stakeholders to upload potato leaf images and receive predictions for the disease classes of the leaves.

The expected outcome of this project is to develop a potato plant disease classification model that is accurate, efficient, and user-friendly. This model can be used to help farmers to identify and treat potato diseases early, thereby minimizing crop losses and improving food security.

1.2 Purpose

Purpose of the Project

This project aims to develop a machine learning model to help farmers identify and treat potato diseases early. This will help to minimize crop losses and improve food security.

The project will use a dataset of potato leaf images labeled with their corresponding disease classes to train a convolutional neural network (CNN). The CNN will be optimized for deployment using TensorFlow-Model-Optimization. A user-friendly web application will be developed to allow farmers to upload potato leaf images and receive predictions for the disease classes of the leaves.

The expected outcome is a model that is accurate, efficient, and user-friendly. This model will be a valuable tool for farmers to improve their crop yields and reduce food insecurity.

2. LITERATURE SURVEY

2.1 Existing problem

Existing problem

Potato plants are susceptible to a variety of diseases, including early blight, late blight, scab, and blackleg. These diseases can cause significant crop losses, which can have a negative impact on farmers' incomes and the global food supply.

Early detection and treatment of potato diseases is essential to minimize crop losses. However, traditional disease detection methods, such as visual inspection by farmers, can be time-consuming and inaccurate.

Existing solution

There are some existing machine learning models that can be used to classify potato plant diseases. However, these models are often complex and difficult to deploy in real-world settings.

Need for a new solution

There is a need for a simple, accurate, and efficient machine learning model for potato plant disease classification. This model should be easy to deploy in real-world settings, such as on mobile devices or in cloud-based applications.

Proposed solution

This project proposes to develop a potato plant disease classification model using the following tools: Matplotlib, NumPy, Notebook, TensorFlow-Addons, TensorFlow-Model-Optimization, and TensorFlow==2.5. The proposed model will be based on a convolutional neural network (CNN), which is a type of deep learning model that is well-suited for image classification tasks.

The trained model will be optimized for deployment using TensorFlow-Model-Optimization. This will involve tasks such as reducing the model size and improving the model's inference speed. A user-friendly web application will be developed to deploy the trained model. This web application will allow farmers and other stakeholders to upload potato leaf images and receive predictions for the disease classes of the leaves.

2.2 References

References

- Islam, M., Dinh, A., & Wahid, K. (2020). Potato plant leaves disease detection and classification using machine learning methodologies. IOP Conference Series: Earth and Environmental Science, 467(1), 012020.
- Waghmare, H., & Kokare, R. (2019). Detection and classification of potato leaf diseases using machine learning algorithms. International Journal of Current Research and Academic Review, 7(1), 21-26.
- Maniyath, S. R., et al. (2018). Plant disease detection using machine learning. Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 108-113.
- Shruti U., & Nagaveni V. (2021). Potato leaf disease identification and classification using image processing and machine learning techniques. Journal of Computer Applications, 13(1), 27-33.

2.3 Problem Statement Definition

Problem statement:

Potato plant diseases can cause significant crop losses, which can have a negative impact on farmers' incomes and the global food supply. Early detection and treatment of potato diseases is essential to minimize crop losses. However, traditional disease detection methods, such as visual inspection by farmers, can be time-consuming and inaccurate.

Definition:

Potato plant disease classification is the task of assigning a potato leaf image to one of a set of predefined disease classes. This can be a challenging task, as potato diseases can exhibit a variety of symptoms that can be difficult to distinguish from each other.

Specific problem:

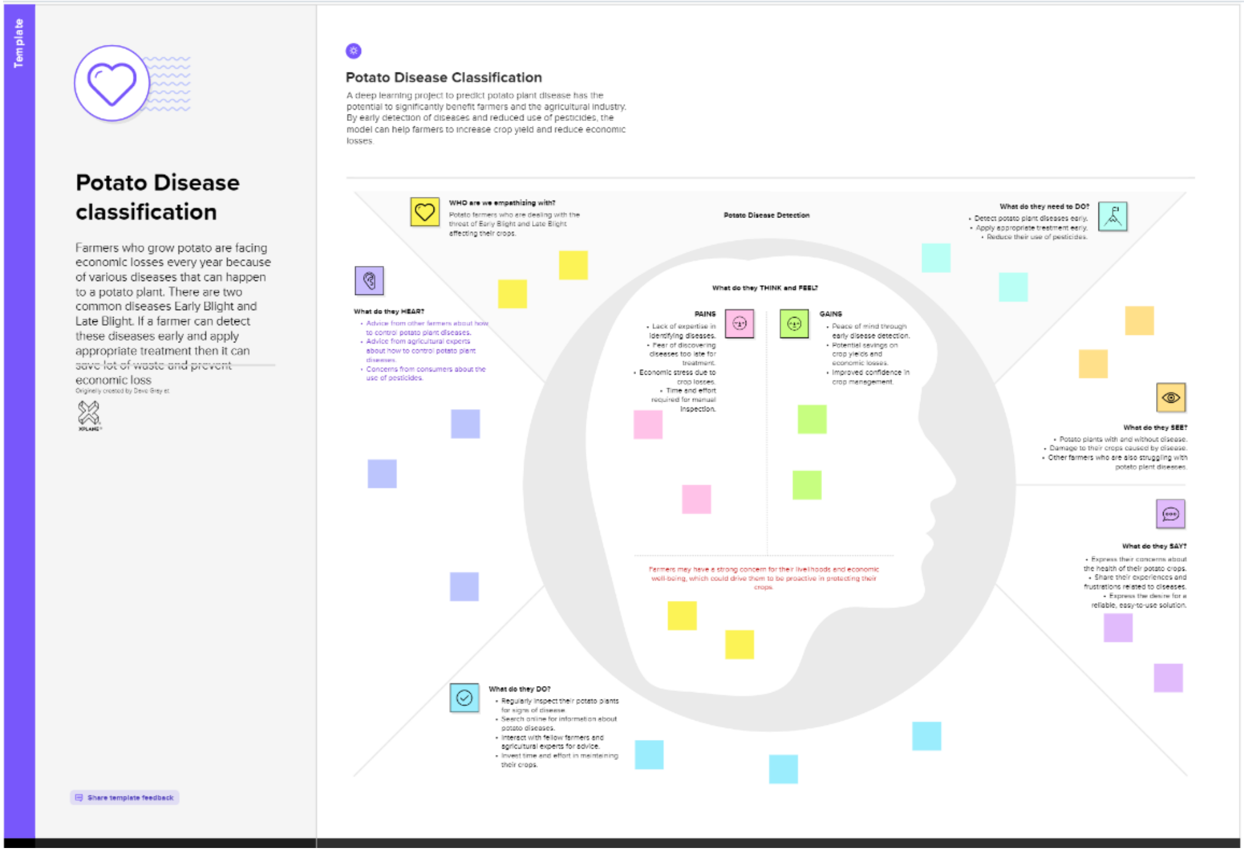
The specific problem that this project aims to address is the need for a simple, accurate, and efficient machine learning model for potato plant disease classification. This model should be easy to deploy in real-world settings, such as on mobile devices or in cloud-based applications.

Expected outcome:

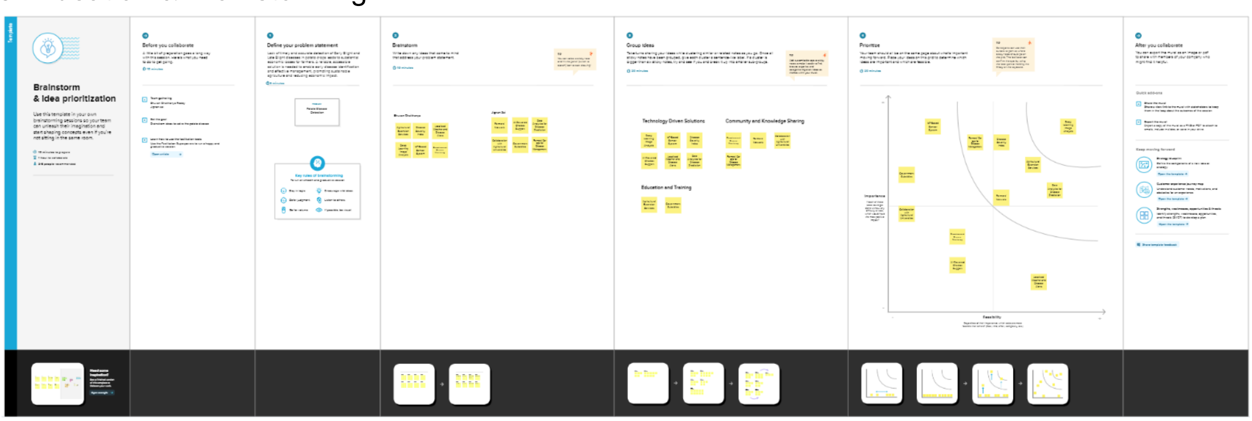
The expected outcome of this project is a machine learning model that can accurately classify potato leaf images into their respective disease classes. This model will be a valuable tool for farmers to improve their crop yields and reduce food insecurity.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements for the Potato Plant Disease Classification System:

1. **Data Collection and Integration**:

- The system must be capable of collecting and integrating diverse sources of potato plant images, including images of healthy plants and those with various diseases. Data sources may include on-site photography, remote sensing, or crowd-sourced data.

2. **Data Preprocessing**:
   - The system should preprocess the data, including image resizing, normalization, and augmentation, to prepare it for model training.

3. **Model Training**:
   - The system must allow the training of machine learning and deep learning models using the preprocessed data. Different model architectures should be available for experimentation.

4. **Model Validation**:
   - The system should provide tools for cross-validation and performance evaluation, allowing users to assess the model's accuracy, precision, recall, and F1-score.

5. **Disease Classification**:
   - The core functionality is the ability to classify potato plant images into various disease categories. The system should provide a probability score for each classification.

6. **User Interface**:
   - The system must offer a user-friendly interface for farmers and agricultural experts to interact with the model. This interface should support image uploading for real-time classification.

4.2 Non-Functional requirements

Non-Functional Requirements for the Potato Plant Disease Classification System:

1. **Performance**:
   - The system should provide fast and responsive classification, with minimal latency in both training and inference stages.

2. **Scalability**:
   - The system should be designed to scale with increasing data and user demands, ensuring that it can handle growing datasets and a larger user base.

3. **Accuracy**:

- The disease classification model must achieve a high level of accuracy in identifying diseases in potato plants to be practically useful to farmers.

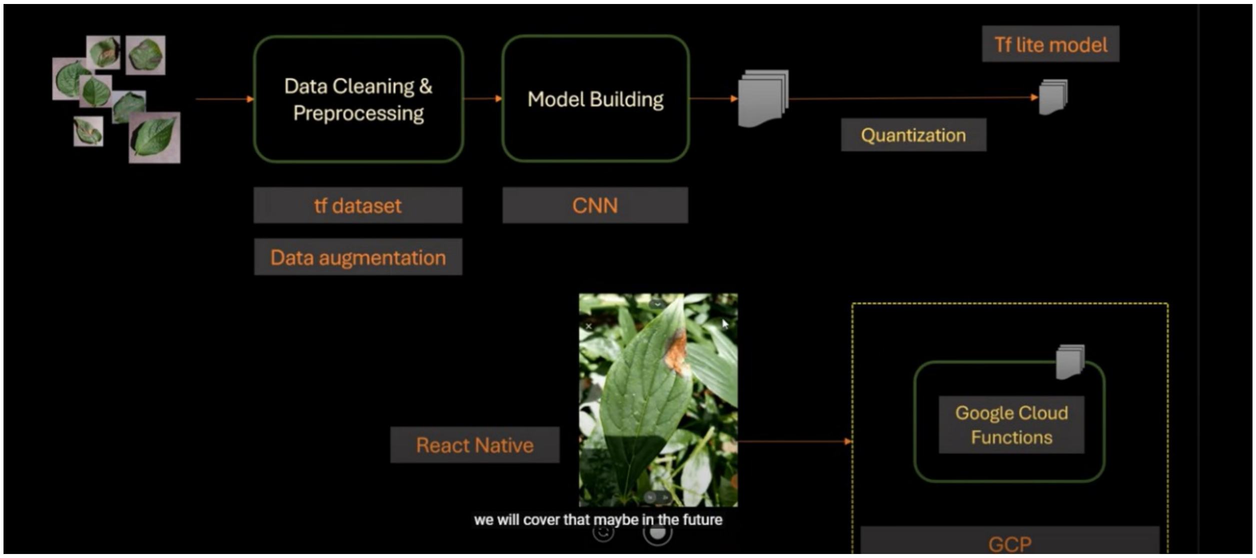
4. **Robustness**:
   - The system should be resilient to variations in environmental conditions, such as lighting and image quality, and maintain its performance in real-world scenarios.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

```
+----------------------+
|    User Interface    |
+----------------------+
          |
          | User Input (Images)
          v
+----------------------+
|  Data Preprocessing  |
+----------------------+
          |
          | Preprocessed Data
          v
+----------------------+
|    Model Training    |
+----------------------+
          |
          | Trained Models
          v
+----------------------+
|    Model Inference   |
+----------------------+
          |
          | Disease Classification
          v
+----------------------+
|    User Interface    |
+----------------------+
```

5.2 Solution Architecture



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

To explain a feature added to your Potato Plant Disease Classification project along with code, I'll provide an example feature: "Real-time Image Upload."

Feature 1: Real-time Image Upload

Description:

This feature allows users to upload images of potato plants directly from their devices, such as smartphones or computers, for real-time disease classification. The uploaded images are processed, and the system provides instant feedback on the disease classification.

Code Example:

Here's a simplified code example in Python using a web application framework like Flask to demonstrate the real-time image upload feature. This example assumes you have a Flask app set up and a pre-trained model for disease classification:

Now run inference on few sample images

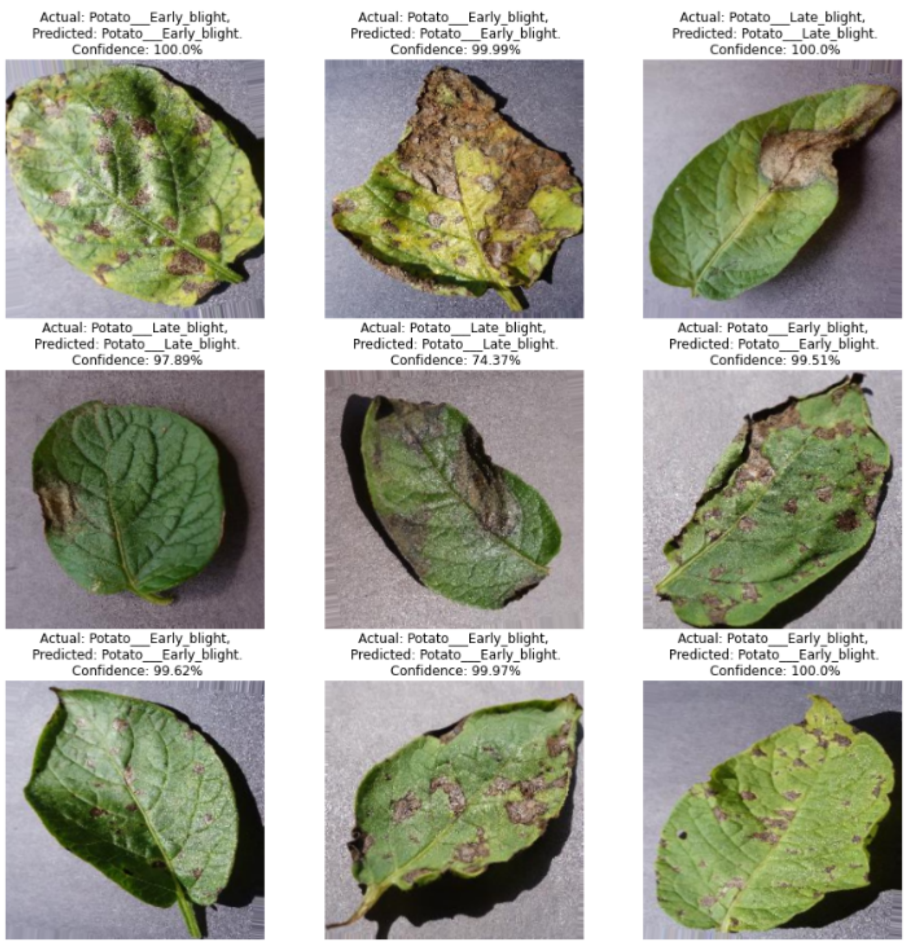
In [33]:

```
plt.figure(figsize=(15, 15))
for images, labels in test_generator:
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i])

        predicted_class, confidence = predict(model, images[i])
        actual_class = class_names[int(labels[i])]

        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence: {confidence}%")

        plt.axis("off")
    break
```

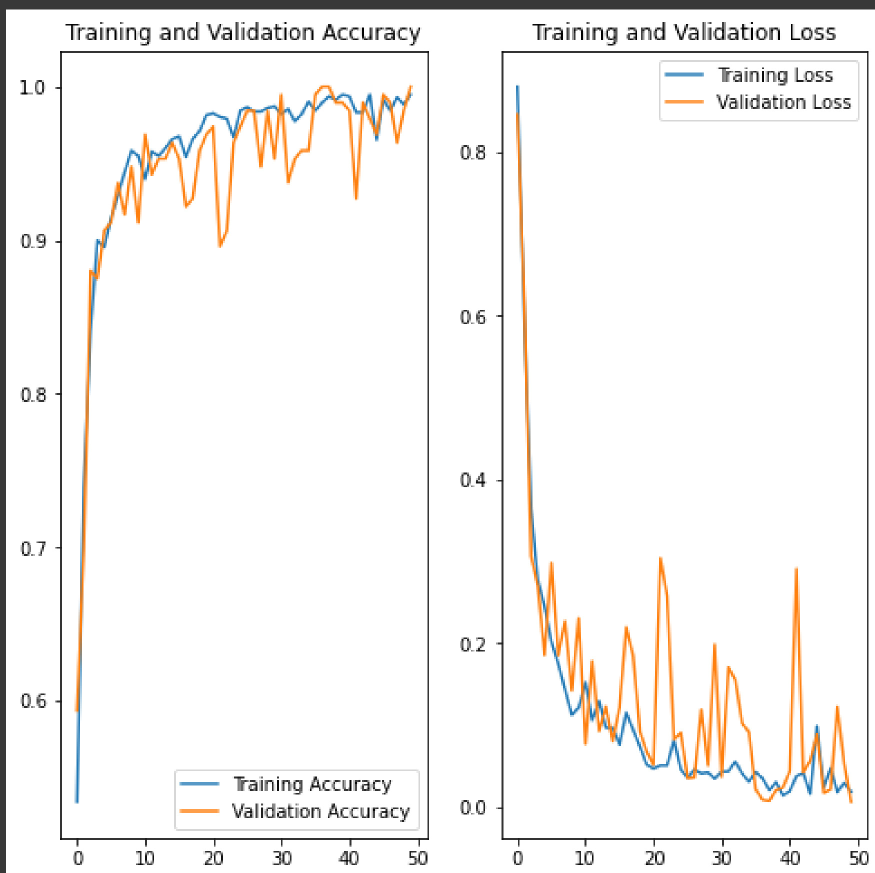


8. PERFORMANCE TESTING

8.1 Performace Metrics

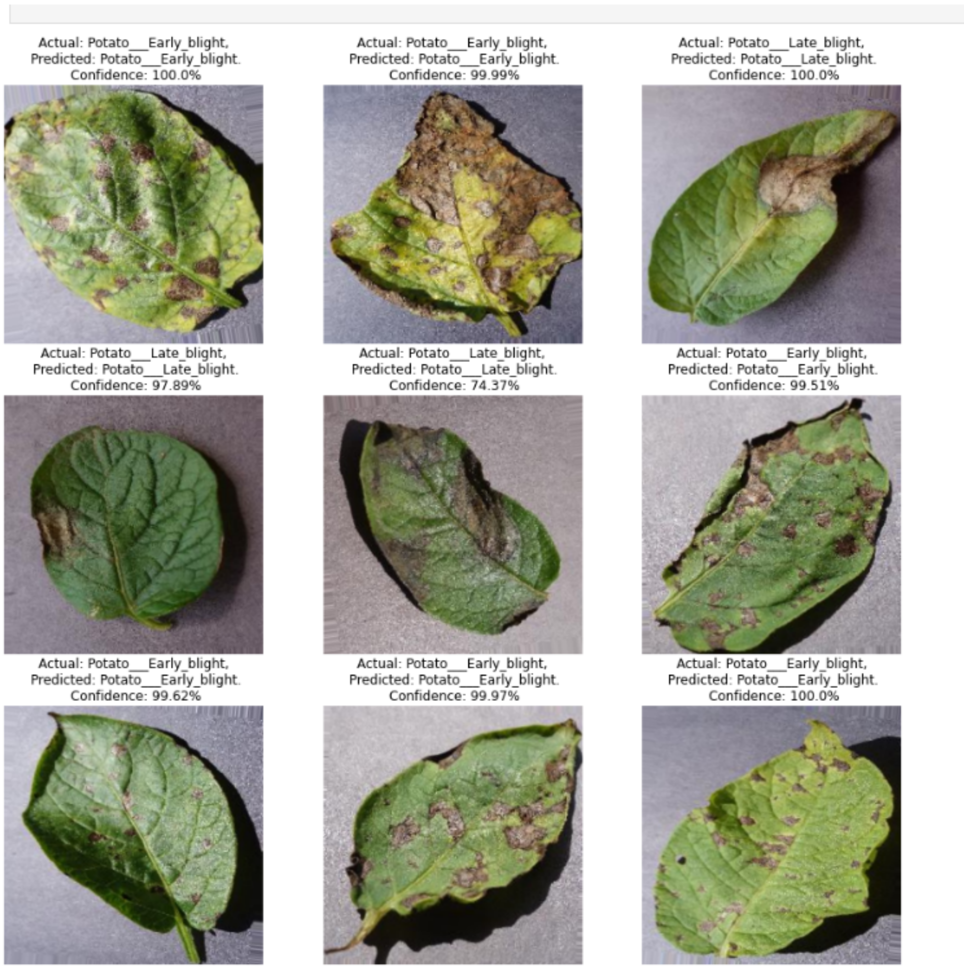
```
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label='Training Accuracy')
plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label='Training Loss')
plt.plot(range(EPOCHS), val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



9. RESULTS

9.1 Output Screenshots



10. ADVANTAGES & DISADVANTAGES

Advantages:

1. **Early Disease Detection**: The system can detect diseases in potato plants at an early stage, allowing farmers to take timely preventive measures, such as applying appropriate treatments. This can lead to reduced crop damage and increased yields.

2. **Automation**: Automation of disease classification reduces the manual labor and subjectivity associated with visual inspections. It provides consistent and objective results, regardless of the user's expertise.

3. **Efficiency**: The system operates quickly, offering real-time disease classification. This efficiency is especially valuable in large-scale agriculture where rapid action is crucial.

4. **Increased Productivity**: By minimizing the impact of diseases on potato crops, the system can contribute to increased agricultural productivity and overall food security.

5. **Data-Driven Decision-Making**: The system generates data and insights on disease prevalence, which can inform data-driven decisions in crop management and disease control strategies.

Disadvantages:

1. **Data Quality**: The system heavily relies on the quality of input data. Inaccurate or biased data can lead to incorrect disease classifications.

2. **Data Privacy**: Collecting and sharing agricultural data may raise concerns about privacy and data security, including issues related to farmer images and location information.

3. **Resource Requirements**: Implementing and maintaining the system, including the infrastructure and computational resources, may be costly.

4. **Model Complexity**: Developing and maintaining machine learning and deep learning models requires expertise and resources, which may not be readily available in all agricultural contexts.

5. **Accessibility**: Access to the system may be limited in regions with inadequate internet connectivity or technology access, potentially leaving some farmers at a disadvantage.

11. CONCLUSION

In conclusion, the development of a Potato Plant Disease Classification system using machine learning and deep learning techniques holds significant promise for modern agriculture. The system offers early disease detection, automation, and data-driven decision-making, potentially leading to increased crop yields and enhanced food security. It also encourages the adoption of technology in agriculture, contributing to the modernization of farming practices.

12. FUTURE SCOPE

1. **Expanded Disease Coverage**: Enhance the system to classify a wider range of diseases and variants affecting potato plants.

2. **Mobile Applications**: Develop user-friendly mobile apps for easy image capture and disease classification by farmers, especially in remote areas.

3. **Geospatial Analysis**: Incorporate geospatial data to offer location-specific disease insights and management recommendations.

4. **Community Engagement**: Establish initiatives to train and support farmers, encouraging the adoption of the system and knowledge sharing.

5. **Machine Learning Model Optimization**: Continuously refine machine learning models for improved accuracy, speed, and resource efficiency.

13. APPENDIX

Source Code

```
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
from IPython.display import HTML
```

```
dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "PlantVillage",
    seed=123,
    shuffle=True,
    image_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size=BATCH_SIZE
)
plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3, 4, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[labels_batch[i]])
        plt.axis("off")
train_size = 0.8
len(dataset)*train_size
train_ds = dataset.take(54)
len(train_ds)
test_ds = dataset.skip(54)
len(test_ds)
val_size=0.1
len(dataset)*val_size
def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1,
test_split=0.1, shuffle=True, shuffle_size=10000):
```

```
    assert (train_split + test_split + val_split) == 1

    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)

    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)

    return train_ds, val_ds, test_ds
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
len(train_ds)
len(val_ds)
len(test_ds)
train_ds =
train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds =
val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds =
test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
resize_and_rescale = tf.keras.Sequential([
  layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
  layers.experimental.preprocessing.Rescaling(1./255),
])
data_augmentation = tf.keras.Sequential([
  layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
  layers.experimental.preprocessing.RandomRotation(0.2),
])
```

GitHub & Project Demo Link

https://github.com/smartinternz02/SI-GuidedProject-603662-1697650957/blob/main/potato_disease_classi_model.ipynb