AI in Potato Agriculture

Step 1: Prototype Selection

Problem Statement

Traditional methods of potato farming involved manual disease detection and remedies, which were labor-intensive and often unreliable. Advancements in technology now allow for more accurate disease detection by analyzing potato leaves. This method uses deep learning techniques like Convolutional Neural Networks and machine learning techniques like K-means algorithms. These models are trained on extensive datasets of healthy and diseased potato leaves, and then deployed into a mobile or Web application. This app allows farmers to take a picture of a potato leaf, detecting any signs of disease, enabling timely interventions and supporting economic stability in farming communities.

Market/Customer/Business Need Assessment

Market/Customer/Business Need Assessment for Potato Disease Detection

Market Need:

- High disease prevalence of potato crops.
- Large scale of production in India.
- Inefficient traditional methods for disease detection.

Customer Need:

- Farmers need simple, accurate, and efficient disease monitoring tools.
- Agricultural consultants benefit from advanced tools for disease diagnosis and treatment recommendations.
- Agribusiness companies need to ensure crop quality and yield for profitability.

Business Need:

- Improved crop yields.
- Technological advancements in agriculture.
- Market expansion through mobile or web application development.

Target Specifications

Mobile or Web Application for Detecting Potato Diseases

- High Accuracy: Identifies common potato diseases with over 90% accuracy.
- Rapid Results: Provides real-time analysis within seconds of image capture.

• Ease of Use: Features an intuitive, user-friendly interface accessible to all technological skill levels.

• Scalability: Supports a large number of users and handles substantial volumes of image data efficiently.

• Offline Functionality: Usable without an internet connection for remote farmers.

• Affordability: Cost-effective solution with minimal usage and maintenance costs.

• Cross-Platform Compatibility: Works on various smartphone operating systems, including Android and iOS.

External Search

- → Dataset
- → CNN

Let's import the dataset and have a look at it!

Dependencies

In [1]:
```
import os
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter
from sklearn.metrics import confusion_matrix, classification_report

import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array

print(tf.__version__)
############ settings ############
data_dir = r"..\data"

batch_size = 32
img_height = 128
img_width = 128
```

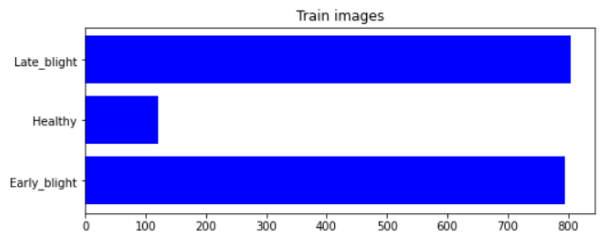
```
2.6.0
```

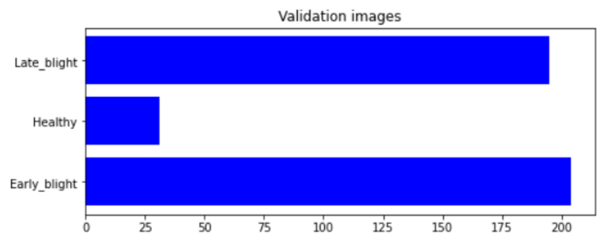
Data preprocessing

- Loading and splitting data
- Train:80% and Validation:20%

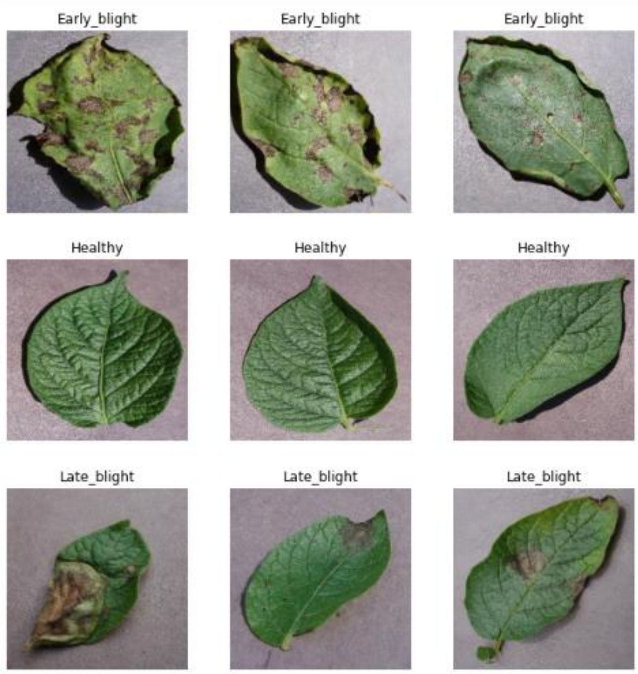
In [2]:
```
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="training",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

```
Found 2152 files belonging to 3 classes.
Using 1722 files for training.
Found 2152 files belonging to 3 classes.
Using 430 files for validation.
```





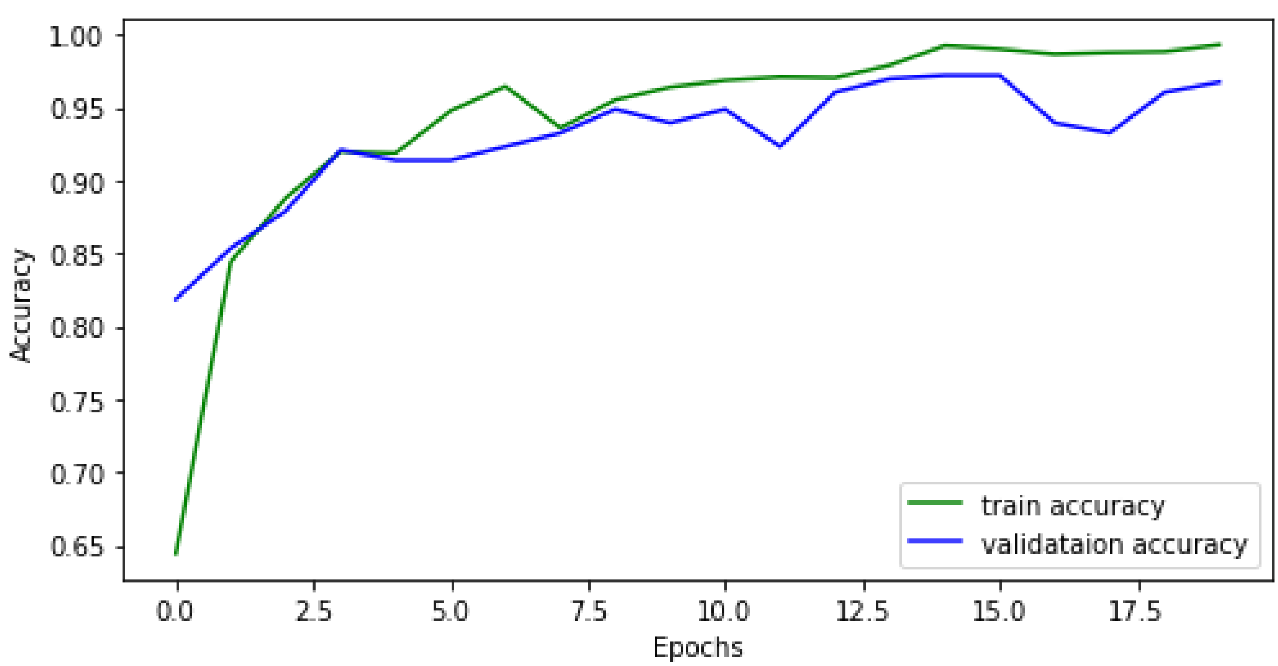
In [5]:
```
## Sample images
class_names = train_ds.class_names

for class_name in class_names:
    imgs = os.listdir(os.path.join(data_dir, class_name))[:3]
    plt.figure(figsize=(10, 10))
    for i, img in enumerate(imgs):
        ax = plt.subplot(3, 3, i+1)
        plt.imshow(plt.imread(os.path.join(data_dir, class_name, img)))
        plt.title(class_name)
        plt.axis('off')
```

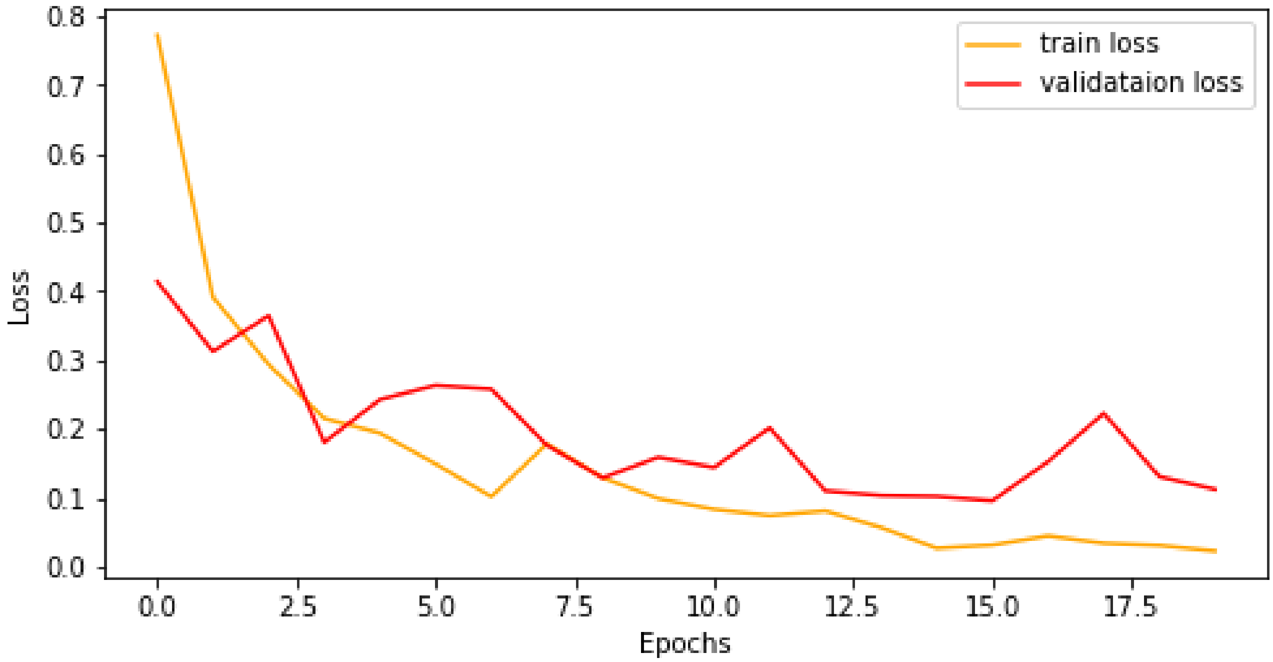


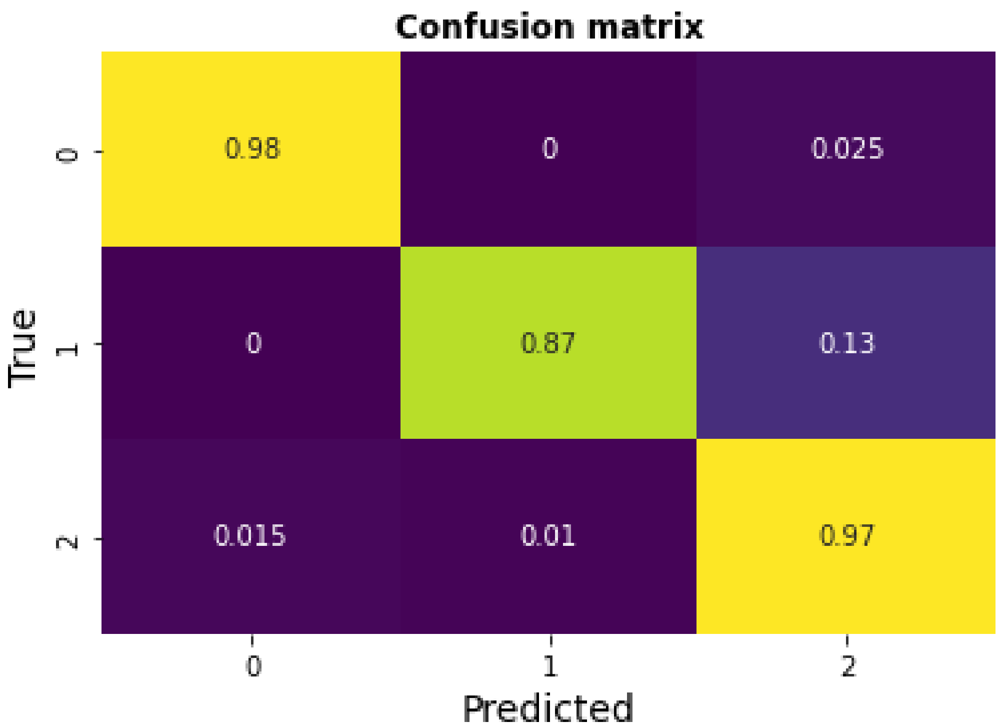
Model Training

In [6]:
```
# Configure the dataset for performance

AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

In [7]:
```
# Model architecture

num_classes = 3

model = tf.keras.Sequential([
    tf.keras.layers.InputLayer(input_shape=(img_height, img_width, 3)),
    tf.keras.layers.experimental.preprocessing.Rescaling(1./255),

    tf.keras.layers.Conv2D(16, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Conv2D(32, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Conv2D(64, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Dense(num_classes, activation='softmax')

])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

earlystop_callback = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy',
                                                      min_delta=0.0001,
                                                      patience=5)

history = model.fit(train_ds,
                    validation_data=val_ds,
                    epochs=20,
                    callbacks=[earlystop_callback])
```







In [25]:
```
print(classification_report(correct_labels, predicted_labels))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       204
           1       0.93      0.87      0.90        31
           2       0.95      0.97      0.96       195

    accuracy                           0.97       430
   macro avg       0.96      0.94      0.95       430
weighted avg       0.97      0.97      0.97       430
```

Applicable Regulations

Regulatory Compliance for Potato Disease Detection App in India

Data Privacy and Protection:

- Personal Data Protection Bill (PDPB): App must secure explicit user consent for data collection and ensure transparency and security.

Agricultural Regulations:

- Insecticides Act, 1968: Recommendations for disease treatments must comply with approved pesticide use.
- Seeds Act, 1966: App should help farmers identify disease-free seeds and offer best practices for seed treatment and storage.

Environmental Regulations:

- Environment (Protection) Act, 1986: Promote sustainable disease management practices.
- Water and Air Pollution Acts: Prevent environmental pollution.

Health and Safety Regulations:

- FSSAI: Ensure disease treatments align with food safety standards.
- Occupational Safety Code, 2020: Provide safe usage guidelines for pesticides and treatments.

Intellectual Property Rights:

- Indian Patent Act, 1970: Ensure technologies do not infringe on existing patents.
- Copyright Act, 1957: Ensure all app content is original or properly licensed.

Electronic and Information Technology Regulations:

- Information Technology Act, 2000: Ensure secure data transmission and storage.
- Digital India Initiative: Promote digital literacy and reliable tools for farmers.

Applicable Constraints

Developing a Mobile or Web Application for Potato Disease Detection in India

Space Constraints:

- Efficient storage for high-resolution images, user data, and machine learning models.
- Network optimization for offline functionality in rural areas.

Budget Constraints:

- Development Costs: Investment in software development, machine learning model training, and maintenance.
- Operational Costs: Budget for cloud services, server maintenance, cybersecurity, and user support.
- Hardware: Compatibility with a wide range of smartphones, including low-cost devices.
- Marketing and Training: Allocate resources for app promotion and farmer training programs.

Expertise Constraints:
• Technical Expertise: Skilled professionals in software development, machine learning, and agricultural science.
• User Training: Design user-friendly interfaces and conduct training sessions.
• Support and Maintenance: Provide ongoing technical support, updates, and model retraining.

Business Opportunity

Smartphone or web Application for Detecting Potato Diseases in India's Agriculture

• Fills a crucial need in India's agriculture industry.

• Offers significant commercial opportunity.

• Improves crop health management, positively impacting India's GDP.

• Provides fast, accurate disease diagnosis and treatment suggestions.

• Monetization ideas include premium features, subscription models, and alliances with agricultural groups.

• Potential expansion to include more crops and geographical areas.

• Promotes agricultural sustainability and food security.

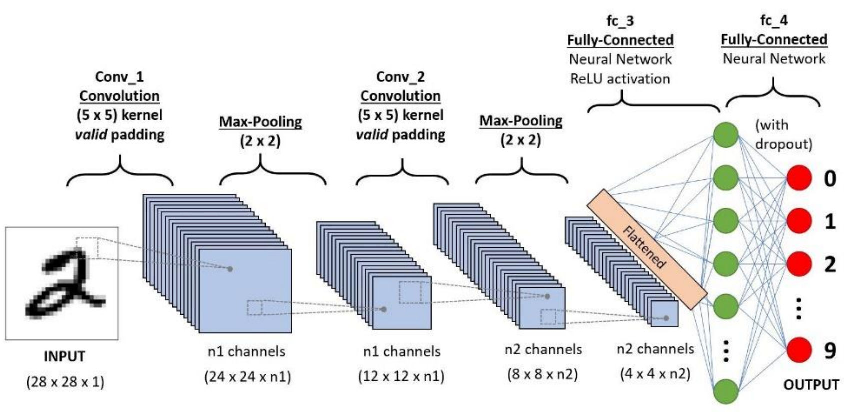
Concept Generation

For successful implementation, the proposed service will require the following algorithms, tools and experts.

Algorithms:

CNN

CNNs for Classification and Recognition of Images
• Deep learning algorithms for categorization and recognition of images.
• Use backpropagation to discover the spatial hierarchies of features.
• Employ fully connected layers, pooling, and several convolutional layers.
• Good at spotting features and patterns in photos.
• Robust feature extraction; • Able to handle substantial amounts of picture data.



Tools:

- → Python: It’s a programming language that will be used for building the service.
- → Pandas: Pandas is a library mainly used for handling, manipulating and transforming data.
- → Scikit-learn: It is the gold standard library for machine learning which comes with plenty of algorithms to perform different tasks such as regression, classification etc.
- → Matplotlib and Seaborn: Both of these libraries are used for visualization purposes.

Team:

• **Project Manager**: Oversees project milestones and budget allocation.
• **Machine Learning Engineers**: Develop and refine disease detection models.
• **Mobile and WebApp Developers**: Responsible for app creation and ML model integration.
• **Backend Developers**: Handle data management and server-side functionalities.
• **UI/UX Designers**: Design intuitive interfaces and ensure user-friendly experiences.
• **Agricultural Experts**: Provide domain-specific insights and validate model accuracy.
• **Quality Assurance Testers**: Conduct thorough testing to ensure app functionality and user satisfaction.

Links:

Github

Business Model for Potato Disease Classification System Using Deep Learning Algorithms

Adwait Gore

The Potato Disease Prediction System (PDPS) leverages advanced deep learning algorithms to identify and predict diseases in potato crops. This innovative solution aims to improve crop yield, reduce losses, and optimize the use of pesticides. The PDPS will serve farmers, agricultural enterprises, and aggrotech companies by providing accurate and timely disease predictions, allowing for proactive management and intervention.

Market Analysis

1. **Target Market**:
    - Small to large-scale potato farmers.
    - Agricultural cooperatives.
    - Aggrotech companies.
    - Government and non-governmental agricultural bodies.

2. **Competitive Landscape**:
    - Existing solutions include manual inspection, traditional software tools, and emerging AI-based applications.
    - The competitive edge of PDPS lies in its accuracy, real-time analysis, and ease of use.

Value Proposition

1. **Precision Agriculture**:
    - Real-time disease detection and prediction.
    - Reduced crop losses and improved yield.
    - Optimized pesticide use, leading to cost savings and environmental benefits.

2. **Economic Benefits**:
    - Increased profitability for farmers through better crop management.
    - Lower operational costs due to reduced need for extensive manual inspections.

3. **Sustainability**:
    - Supports sustainable farming practices.
    - Reduces environmental impact through targeted pesticide application.

Technology

1. **Deep Learning Algorithms**:
    - Convolutional Neural Networks (CNNs) for image recognition and classification.
    - Recurrent Neural Networks (RNNs) for time-series analysis and prediction.

2. **Data Sources**:
    - High-resolution images from drones or smartphones.
    - Weather data and soil health indicators.

3. **Platform Features**:
    - User-friendly mobile and web applications.
    - Real-time alerts and recommendations.
    - Integration with existing farm management systems.

Revenue Model

1. **Subscription-Based Model**:
    - Monthly or annual subscription fees based on the size of the farm and number of users.
2. **Freemium Model**:
    - Basic features available for free with premium features accessible through a paid subscription.
3. **Consulting Services**:
    - Expert consulting and custom solutions for large agricultural enterprises.

Marketing and Sales Strategy

1. **Digital Marketing**:
    - Social media campaigns targeting farmers and aggrotech enthusiasts.
    - Content marketing through blogs, webinars, and online demos.

2. **Partnerships**:
   - Collaborations with agricultural cooperatives and government bodies.
   - Partnerships with drone and farm equipment manufacturers.

3. **Sales Channels**:
   - Direct sales through a dedicated sales team.
   - Online sales via the company's website and aggrotech marketplaces.

Operations Plan

1. **Development Team**:
   - Software engineers, data scientists, and agronomists to develop and maintain the platform.
2. **Customer Support**:
   - 24/7 support through chat, email, and phone.
   - Online resources such as FAQs, tutorials, and user manuals.
3. **Infrastructure**:
   - Cloud-based servers for scalable and secure data storage and processing.

Risks and Mitigation

1. **Technical Challenges**:
   - Continuous improvement and testing of algorithms to ensure accuracy.
2. **Market Adoption**:
   - Extensive user training and support to facilitate adoption.
3. **Regulatory Risks**:
   - Compliance with agricultural and data privacy regulations.

Conclusion

The Potato Disease Prediction System using deep learning algorithms represents a significant advancement in agricultural technology. By providing accurate and timely disease predictions, PDPS will help farmers optimize their crop management practices, reduce losses, and increase profitability. The proposed business model outlines a sustainable path to market entry and growth, leveraging cutting-edge technology and strategic partnerships.

Financial Modeling, Machine Learning, and Data Analysis Report for Potato Disease Detection System

SOWMIYA V

1. Financial Modeling

1.1 Revenue Projections

- **Revenue Streams:** Define primary revenue sources: subscription fees, consulting services, and premium features.
- **Pricing Strategy:** Detail pricing structure for different tiers: basic and premium subscriptions.
- **Customer Segmentation:** Identify target segments: small to large-scale farmers, agricultural cooperatives, agritech companies.
- **Revenue Forecast:** Develop a revenue forecast based on market research and assumptions about customer acquisition and retention.

1.2 Cost Structure

- **Fixed Costs:** Identify fixed expenses: salaries, rent, software licenses, hardware.
- **Variable Costs:** Detail variable costs: product development, marketing, operations.
- **Cost Per Acquisition (CPA):** Estimate the cost to acquire a new customer.
- **Customer Lifetime Value (CLTV):** Calculate projected revenue from a customer over their lifetime.

1.3 Profit and Loss (P&L) Statement

- **Revenue and Cost Summary:** Combine revenue and cost data to create a P&L statement.
- **Profit Margin Analysis:** Calculate gross and net profit margins to assess profitability.

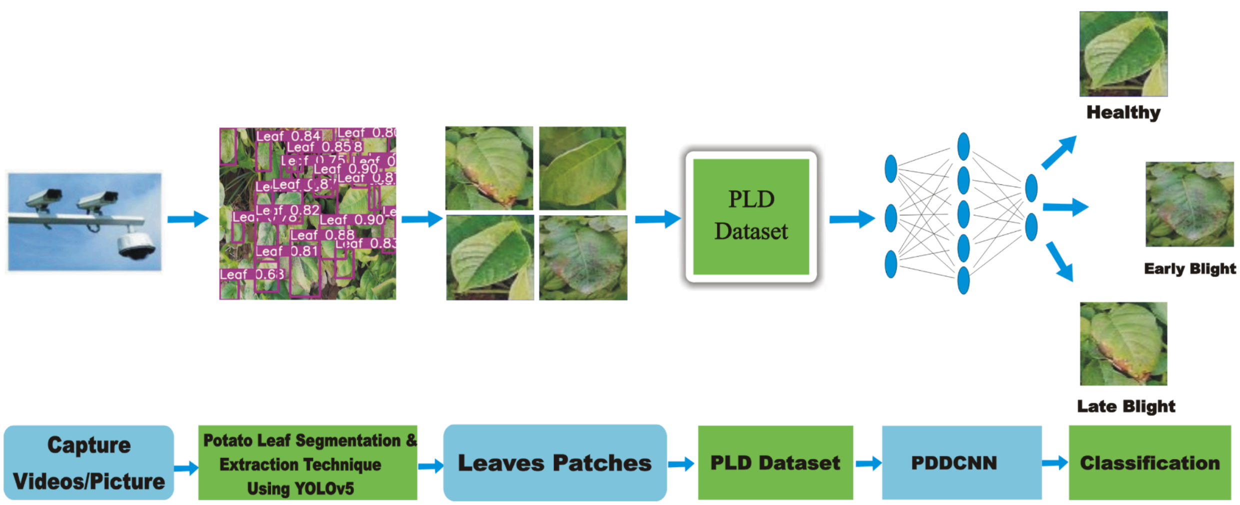
1.4 Cash Flow Statement

- **Cash Inflows and Outflows:** Identify cash inflows (revenue, investments) and outflows (expenses, investments).
- **Cash Burn Rate:** Determine the rate at which the company spends cash.
- **Break-Even Analysis:** Calculate the revenue required to cover costs.

1.5 Sensitivity Analysis

- **Key Variable Identification:** Identify critical variables affecting financial performance (e.g., customer acquisition cost, churn rate).
- **Scenario Analysis:** Evaluate the impact of different scenarios on financial outcomes.

2. Machine Learning



2.1 Data Acquisition and Preparation

- **Data Sources:** Describe sources of data (e.g., image datasets, weather data, soil data).
- **Data Cleaning:** Outline data cleaning processes (e.g., handling missing values, outliers, inconsistencies).
- **Data Preprocessing:** Explain data transformations (e.g., normalization, feature scaling).
- **Data Splitting:** Describe how the data is divided into training, validation, and testing sets.

2.2 Model Development and Training

- **Model Selection:** Justify choice of machine learning algorithms (e.g., CNN, SVM).
- **Model Architecture:** Detail model architecture and hyperparameters.
- **Training Process:** Describe training methodology, including optimization algorithms and loss functions.
- **Model Evaluation:** Explain evaluation metrics used (e.g., accuracy, precision, recall, F1-score).

2.3 Model Deployment

- **Deployment Platform:** Specify platform for deploying the model (e.g., cloud, mobile app).
- **Integration with Application:** Describe how the model is integrated into the potato disease detection application.
- **Model Monitoring:** Outline plans for monitoring model performance in production.

3. Data Analysis

3.1 Exploratory Data Analysis (EDA)

- **Data Visualization:** Describe visualization techniques used to explore data patterns.
- **Data Insights:** Summarize key findings from EDA.

3.2 Feature Engineering

- **Feature Creation:** Explain the process of creating new features from existing data.
- **Feature Selection:** Describe methods used to select relevant features.

3.3 Model Performance Analysis

- **Performance Metrics:** Detail metrics used to evaluate model performance.
- **Error Analysis:** Analyze model errors to identify potential improvements.

4. Integration of Financial Modeling and Machine Learning

4.1 Model Accuracy Impact: Assess the impact of model accuracy on revenue and costs.

4.2 Data-Driven Decision Making: Demonstrate how data and model insights inform business decisions.

5. Conclusion

This report provides a comprehensive overview of the financial modeling, machine learning, and data analysis aspects of the Potato Disease Detection System project. The insights gained from this analysis will be instrumental in making informed business decisions and optimizing the project's performance.

Reference : Github Link