Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27 October 2023
Team ID	PNT2022TMID592620
Project Name	Project - xxx
Maximum Marks	4 Marks

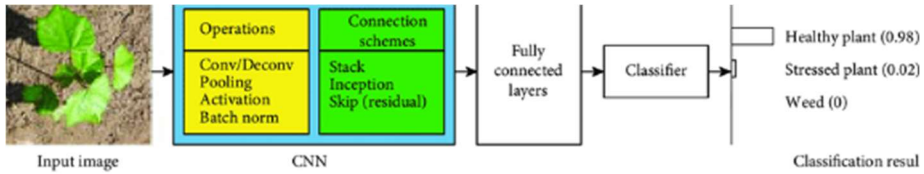
Technical Architecture:

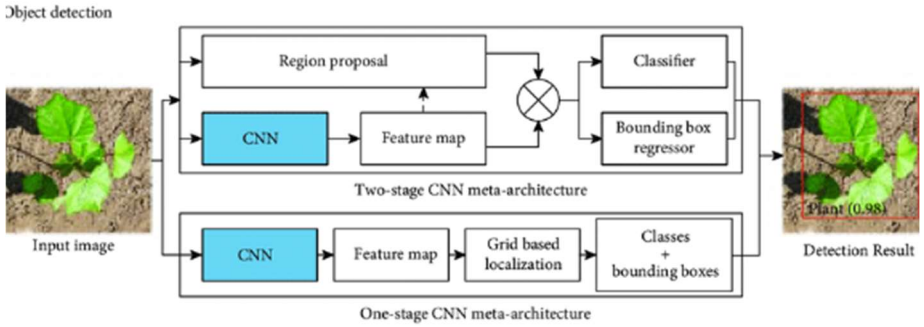
Green Classify: Deep Learning-based Approach for Vegetable Image Classification:

Reference:

Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)


Operations
Conv/Deconv
Pooling
Activation
Batch norm
Connection schemes
Stack
Inception
Skip (residual)
Fully connected layers
Classifier
Healthy plant (0.98)
Stressed plant (0.02)
Weed (0)
Input image
CNN
Classification resul


Object detection
Region proposal
Classifier
CNN
Feature map
Bounding box regressor
Two-stage CNN meta-architecture
Input image
CNN
Feature map
Grid based localization
Classes + bounding boxes
One-stage CNN meta-architecture
Plant (0.98)
Detection Result

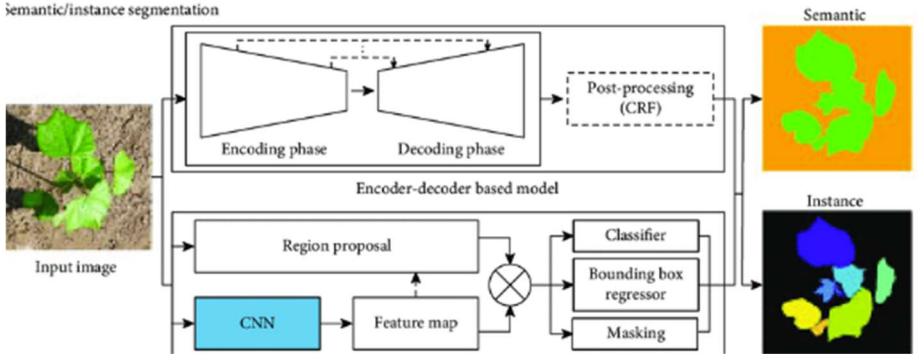

Semantic/instance segmentation
Post-processing (CRF)
Encoding phase
Decoding phase
Encoder-decoder based model
Semantic
Instance
Input image
Region proposal
Classifier
Bounding box regressor
CNN
Feature map
Masking

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Image Data Source	This is where the vegetable images are collected from, which can be local storage or a cloud-based repository.	
2.	Preprocessing	The preprocessing component will clean and prepare the images for the classification model. Preprocessing tasks include resizing images to a consistent dimension, normalizing pixel values, and applying data augmentation techniques (e.g., rotation, flipping).	
3.	Deep Learning Model	The core of the application is a deep learning model, such as a Convolutional Neural Network (CNN), for image classification. The model is loaded and initialized with pre-trained to benefit from transfer learning.	CNN, PyTorch, TensorFlow
4.	Training Infrastructure	Training hyperparameters and the dataset are configured, and the training process begins. During training, the model's weights are updated based on the training data, and checkpoints are saved.	AWS EC2 , Azure VMs
5.	Training Data	The training data is a labeled dataset containing images of various vegetables, with corresponding class labels. Data is partitioned into training, validation, and test sets to evaluate model performance.	
6.	Inference Engine	The inference engine is responsible for serving the trained model to make predictions on new, unlabeled vegetable images. It exposes an API that accepts image data as input and returns the predicted class label along with a confidence score. The API may use a RESTful endpoint or a WebSocket for real-time interactions.	RESTFUL API's ,Websockets
7.	User Interface	The user interface (UI) allows users to interact with the application.	

		Users can upload images through the UI for classification or provide image URLs. The UI displays the classification results, including the predicted vegetable type and confidence score.	
8.	Training Data Storage	Relational Database Management System (RDBMS) suitable for structured data. NoSQL Database for training dataset is semi-structured or unstructured.	MYSQL, PostgreSQL, Microsoft SQL MongoDB or Cassandra
9.	Cloud Object Storage	Cloud Storage are excellent options for storing images in the cloud. They offer scalability and data durability.	Amazon S3 Google Cloud Storage Azure Blob
10.	Machine Learning Model	Convolutional Neural Networks (CNNs) are a common choice for image classification tasks,	Object Recognition Model, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Scalability	The architecture should support horizontal scaling, especially if it's a cloud-based solution to handle increasing user demands.	
2	Data Storage	Data storage components or services are required for storing training data, user-uploaded images, and model checkpoints. These can be databases or cloud storage solutions.	
3	External Interfaces	The application may use third-party APIs for tasks like image preprocessing, image hosting, or feedback collection.	
4	Deployment	The application can be deployed in the cloud or locally, depending on scalability requirements.	

S.No	Characteristics	Description	Technology
5	Machine Learning Interface	There is an interface between the application and the machine learning model. This interface allows the application to send images for classification and receive classification results.	
6	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	

References:

TensorFlow Documentation: https://www.tensorflow.org/

PyTorch Documentation: https://pytorch.org/docs/stable/index.html

AWS Documentation: https://docs.aws.amazon.com/

Azure Documentation: https://docs.microsoft.com/en-us/azure/

Google Cloud Documentation: https://cloud.google.com/docs

scikit-learn Documentation: https://scikit-learn.org/stable/documentation.html

Keras Documentation: https://keras.io/

fastai Documentation: https://docs.fast.ai/

documentation for Google Cloud Vision API: https://cloud.google.com/vision/docs