Model Development Phase Template

Date	15 March 2024
Team ID	SWTID1720033149
Project Title	Visual Diagnostics: Detecting Tomato Plant Diseases With Leaf Image Analysis
Maximum Marks	5 Marks

Model Selection Report

In the model selection report for future deep learning and computer vision projects, various architectures, such as CNNs or RNNs, will be evaluated. Factors such as performance, complexity, and computational requirements will be considered to determine the most suitable model for the task at hand.

Model Selection Report:

Model	Description
ResNet152V2	It is based on the ResNet152V2 architecture, a deep convolutional neural network pre-trained on ImageNet. It has been adapted for a tomato leaf classification task using transfer learning. The model includes **global average pooling**, **batch normalization**, and **dense layers** for classification. Training involves **data augmentation**, **normalization**, and **validation** using a subset of the data for evaluation.
Custom Layers	The custom layers added on top of the ResNet152V2 base model serve to adapt its learned features to the specific task of classifying tomato plant diseases from leaf images. The `GlobalAveragePooling2D` layer is crucial for reducing the spatial dimensions of the feature maps generated by the base model, aggregating spatial information across each feature map. Following this, the `Dense` layer with 1000 units and

	ReLU activation facilitate learning of intricate patterns in the flattened feature vectors, enhancing the model's ability to discern fine-grained distinctions among different disease classes. Finally, the output layer uses a `Dense` layer with 10 units and softmax activation to produce probabilities for each disease class, enabling the model to predict the most likely disease category for a given input image. Together, these custom layers optimize the transfer learning process, tailoring the powerful feature extraction capabilities of ResNet152V2 to effectively address the specific challenges posed by tomato plant disease classification.
Model Initialization	The `Model` class from Keras is used to define the final model architecture, specifying the input (from the base model) and output layers (custom layers added).