Model Optimization and Tuning Phase Template

Date	24 April 2024
Team ID	**738194**
Project Title	RIPE-SENSE: MANGO QUALITY GRADING WITH IMAGE ANALYSIS AND DEEP LEARNING.
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyper parameter Tuning Documentation (8 Marks):

Model	**Tuned Hyperparameters**
VGG16	· **loss='categorical_crossentropy'**: This is a common loss function for multi-class classification problems, not a hyperparameter of VGG16 itself. · **optimizer='adam'**: Adam is a popular optimizer for training neural networks, again not specific to VGG16. · **Learning rate**: This is not explicitly set in code .Adam optimizer typically uses a default learning rate, but we can adjust it during compilation (e.g., Adam(learning_rate=0.001)). · **Metrics=**['accuracy']:['accuracy'] is the chosen metric to monitor during training. We can add other metrics like precision or recall.
CNN(Sequential)	· **Optimizer:** rmsprop (stands for Root Mean Square Prop) is the chosen optimizer for updating model weights during training. This is a hyperparameter because you could have chosen a different optimizer like Adam, SGD, etc. · **Loss:** categorical_crossentropy is the function used to measure the error between the model's predictions and the true labels (multi-class classification). While this is a common choice, it technically isn't a hyperparameter you can tune in this specific context. · **Metrics:** ["accuracy"] is a list containing a single metric, accuracy, to track model performance during training. You could consider this a hyperparameter as you could choose different metrics (e.g., precision, recall) or even a combination.

Final Model Selection Justification (2 Marks):

Final Model	**Reasoning**
VGG16	I decided to utilize VGG16 because it is a great choice for our CNN architecture based on its success in image classification. Its strength lies in its convolutional layers that expertly extract features from images, allowing for precise recognition during testing. VGG16's proven ability to achieve high validation accuracy scores makes it a strong contender for our project.