

Model Optimization and Tuning Phase Template

Date	11 July 2024
SWTID1720080895	SWTID1720080895
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.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
VGG16	<ul style="list-style-type: none"> - <u>loss='categorical_crossentropy'</u>: Commonly used for multi-class classification tasks, this loss function is not a specific hyperparameter of VGG16 itself. - <u>optimizer='adam'</u>: Adam optimizer is widely employed for neural network training, offering adaptive learning rates, but it's not unique to VGG16. - <u>Learning rate</u>: Typically not explicitly set in code; Adam optimizer uses a default learning rate, adjustable during compilation (e.g., Adam(learning_rate=0.001)). - <u>Metrics=['accuracy']</u>: 'accuracy' is selected as the primary metric for monitoring training performance, though additional metrics like precision or recall can be included.

EfficientNet	<ul style="list-style-type: none"> - <u>loss='categorical_crossentropy'</u>: Commonly used for multi-class classification tasks, this loss function is not a specific hyperparameter of the model itself. - <u>optimizer='adamax'</u>: Adamax optimizer, similar to Adam but with potentially better performance on very large datasets or in models with large parameter spaces, although not unique to the model. - <u>Learning rate</u>: Set explicitly to 0.01, impacting the speed and convergence of training within the model. - <u>epochs = 100</u>: Defines the number of complete passes through the training dataset during model training, influencing the depth of learning. - <u>callbacks :</u> <ul style="list-style-type: none"> - ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=1, verbose=1): Adjusts the learning rate when validation loss plateaus, potentially enhancing model performance and convergence speed. - EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True, verbose=1): Halts training when validation loss fails to improve for a specified number of epochs, preserving the model's best weights.
--------------	---

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
EfficientNet	Our EfficientNet implementation leverages callback mechanisms to dynamically adjust training parameters like learning rates, incorporating features such as compound scaling to optimize model depth, width, and resolution. These advancements contribute to achieving impressive accuracies, with reported results reaching up to 96.6% accuracy in testing data.