Model Optimization and Tuning Phase Template

Date	16 March 2024
Team ID	SWTID1720418653
Project Title	Crystal Clear Vision: Revolutionizing Cataract Prediction through Transfer Learning Mastery
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	• **Adam Optimizer:** Adapts the learning rate for each parameter, making it effective for many types of neural networks and data. • **Learning Rate:** A very small learning rate (0.00001) indicates cautious adjustments to the model weights, which is useful if the model is very complex or if you want to fine-tune it gently. • **Binary Cross-entropy Loss:** Suitable for binary classification tasks, ensuring that the model's predicted probabilities are close to the true binary labels. • **Accuracy Metric:** Provides an easy-to-understand measure of the model's performance in terms of the proportion of correctly classified samples. `vgg16.compile(loss='binary_crossentropy', optimizer=Adam(learning_rate=0.00001), m`
VGG19	• **Adam Optimizer:** Adapts the learning rate for each parameter, making it effective for many types of neural networks and data. • **Learning Rate:** A very small learning rate (0.00001) indicates cautious adjustments to the model weights, which is useful if the model is very complex or if you want to fine-tune it gently.

	• **Binary Cross-entropy Loss:** Suitable for binary classification tasks, ensuring that the model's predicted probabilities are close to the true binary labels. • **Accuracy Metric:** Provides an easy-to-understand measure of the model's performance in terms of the proportion of correctly classified samples. `vgg19.compile(loss='binary_crossentropy', optimizer=Adam(learning_rate=0.00001), m`
InceptionNet	• **Adam Optimizer:** Adapts the learning rate for each parameter, making it effective for many types of neural networks and data. • **Learning Rate:** A very small learning rate (0.00001) indicates cautious adjustments to the model weights, which is useful if the model is very complex or if you want to fine-tune it gently. • **Binary Cross-entropy Loss:** Suitable for binary classification tasks, ensuring that the model's predicted probabilities are close to the true binary labels. • **Accuracy Metric:** Provides an easy-to-understand measure of the model's performance in terms of the proportion of correctly classified samples. `model.compile(optimizer=Adam(learning_rate=0.00001), loss='binary_crossentropy', m`
XceptionNet	• **Adam Optimizer:** Adapts the learning rate for each parameter, making it effective for many types of neural networks and data. • **Learning Rate:** A very small learning rate (0.00001) indicates cautious adjustments to the model weights, which is useful if the model is very complex or if you want to fine-tune it gently. • **Binary Cross-entropy Loss:** Suitable for binary classification tasks, ensuring that the model's predicted probabilities are close to the true binary labels. • **Accuracy Metric:** Provides an easy-to-understand measure of the model's performance in terms of the proportion of correctly classified samples. `model.compile(optimizer=Adam(learning_rate=0.00001), loss='binary_crossentropy', m`
ResNet50	• **Optimizer:** Adam with a learning rate of 0.00001, which adjusts weights to minimize the loss function gradually. • **Loss Function:** Categorical crossentropy, used for multi-class classification tasks to measure prediction accuracy.

	• **Metrics:** Accuracy, to monitor the fraction of correct predictions made by the model during training and evaluation.
	
	Categorical cross-entropy: This is the loss function used for multi-class classification problems where each sample can belong to one of many possible classes. It measures the performance of a classification model whose output is a probability value between 0 and 1. This loss function is suitable for one-hot encoded labels.
	
	`# Recompile the model with a lower learning rate`
	`model.compile(optimizer=Adam(learning_rate=0.00001), loss='categorical_crossentropy', m`

MobileNet	• **Adam Optimizer:** Adapts the learning rate for each parameter, making it effective for many types of neural networks and data. • **Learning Rate:** A very small learning rate (0.00001) indicates cautious adjustments to the model weights, which is useful if the model is very complex or if you want to fine-tune it gently. • **Binary Cross-entropy Loss:** Suitable for binary classification tasks, ensuring that the model's predicted probabilities are close to the true binary labels. • **Accuracy Metric:** Provides an easy-to-understand measure of the model's performance in terms of the proportion of correctly classified samples. `model.compile(optimizer=Adam(learning_rate=0.00001), loss='binary_crossentropy', m`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
VGG19	After Finetuning the VGG19 model t reached an accuracy of 98.35% with test loss of 0.082. It is comparitable efficient than other models. So we choose VGG19 has our final model to web deployment.