Model Optimization and Tuning Phase Template

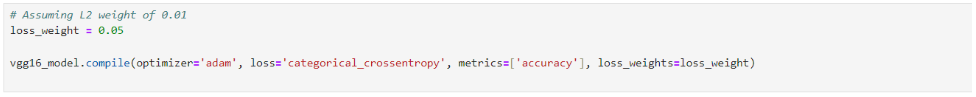
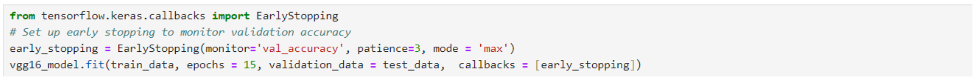
Date	15 March 2024
Team ID	SWTID1720333657
Project Title	Wce Curated Colon Disease Classification Using Deep
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

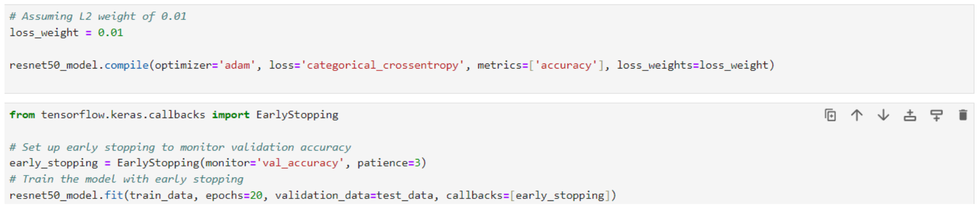
The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

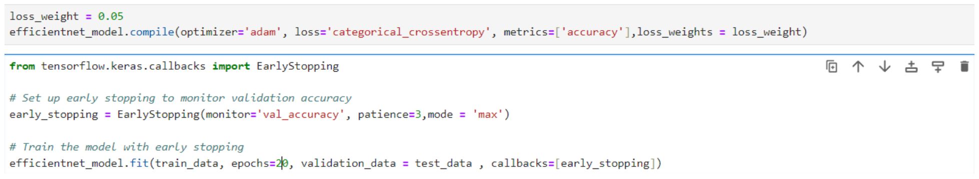
Hyperparameter Tuning Documentation (8 Marks):

Model	**Tuned Hyperparameters**

VGG16	• Loss **('categorical_crossentropy')**: Measures model performance, lower is better.
	• Metrics **(['accuracy'])**: Tracks training progress (percentage of correct predictions).
	• Optimizer **('adam')**: Guides weight updates during training.
	• Epochs **(15)**: Maximum number of training iterations.
	• Early **Stopping**: Stops training if validation accuracy doesn't improve for 3 epochs (prevents overfitting).
	
	`# Assuming L2 weight of 0.01`
	`loss_weight = 0.05`
	
	`vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'], loss_weights=loss_weight)`
	
	`from tensorflow.keras.callbacks import EarlyStopping`
	`# Set up early stopping to monitor validation accuracy`
	`early_stopping = EarlyStopping(monitor='val_accuracy', patience=3, mode = 'max')`
	`vgg16_model.fit(train_data, epochs = 15, validation_data = test_data,  callbacks = [early_stopping])`

ResNet-50	• Loss **('categorical_crossentropy')**: Measures model performance, lower is better.

	• Metrics **(['accuracy'])**: Tracks training progress (percentage of correct predictions).
	• Optimizer **('adam')**: Guides weight updates during training.
	• Epochs **(20)**: Maximum number of training iterations.
	• Early **Stopping**: Stops training if validation accuracy doesn't improve for
	
	3 epochs (prevents overfitting).
	
	`# Assuming L2 weight of 0.01`
	`loss_weight = 0.01`
	
	`resnet50_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'], loss_weights=loss_weight)`
	
	`from tensorflow.keras.callbacks import EarlyStopping`
	
	`# Set up early stopping to monitor validation accuracy`
	`early_stopping = EarlyStopping(monitor='val_accuracy', patience=3)`
	`# Train the model with early stopping`
	`resnet50_model.fit(train_data, epochs=20, validation_data=test_data, callbacks=[early_stopping])`

EfficentNet	• Loss **('categorical_crossentropy')**: Measures model performance, lower is better.
	• Metrics **(['accuracy'])**: Tracks training progress (percentage of correct predictions).
	• Optimizer **('adam')**: Guides weight updates during training.
	• Epochs **(20)**: Maximum number of training iterations.
	• Early **Stopping**: Stops training if validation accuracy doesn't improve for
	
	3 epochs (prevents overfitting).
	
	`loss_weight = 0.05`
	`efficientnet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'],loss_weights = loss_weight)`
	
	`from tensorflow.keras.callbacks import EarlyStopping`
	
	`# Set up early stopping to monitor validation accuracy`
	`early_stopping = EarlyStopping(monitor='val_accuracy', patience=3,mode = 'max')`
	
	`# Train the model with early stopping`
	`efficientnet_model.fit(train_data, epochs=20, validation_data = test_data , callbacks=[early_stopping])`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
EfficientNet	Based on the metrics provided, both ResNet-50 and EfficientNet models exhibit high performance, but each has its strengths. ResNet-50 achieves a final epoch accuracy of 0.9698 and a validation accuracy of 0.9686, slightly surpassing EfficientNet's final epoch accuracy of 0.9695 and validation accuracy of 0.9614. However, EfficientNet demonstrates a significantly lower validation loss of 0.0053 compared to ResNet-50's 0.0158, indicating better performance in minimizing error on the validation set. Given this lower validation loss, EfficientNet appears to generalize better and might perform more reliably on unseen data. Thus, despite ResNet-50's marginally higher accuracy, EfficientNet's superior validation loss suggests it is the preferable model for achieving better overall performance and generalization.