Model Optimization and Tuning Phase Template

Date	8 July 2024
Team ID	SWTID1720076593
Project Title	Visual Diagnostics: Detecting Tomato Plant Diseases through Leaf Image Analysis
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
Model 1	**Accuracy**: 0.9871 **Tuned Hyperparameters**: 1. **Learning Rate**: 0.0001 2. **Batch Size**: 32 3. **Number of Epochs**: 15 **Description**: • **Learning Rate**: A smaller learning rate (0.0001) means the model learns slowly but accurately, making better adjustments to improve over time. • **Batch Size**: The model updates its learning every 32 samples, which balances the learning process and computing power needed. • **Number of Epochs**: The model goes through the entire training data 15 times, which helps it learn well without learning too much noise from the data.

	Screenshot of Code:
	
	Batch Size: 32
	
	`batch_size=32,`
	`image_size=(128, 128),`
	`shuffle=True,`
	
	Learning Rate: 0.0001
	
	`model.compile(optimizer=tf.keras.optimizers.Adam(`
	`learning_rate=0.0001),loss='categorical_crossentropy',metrics=['accuracy'])`
	
	Number of Epochs: 15
	
	`training_history = model.fit(x=training_set,validation_data=validation_set,epochs=15)`

Model 2	**Accuracy: 0.9359**
	
	Tuned Hyperparameters:
	
	1. **Learning Rate**: 0.001
	2. **Batch Size**: 64
	3. **Dropout Rate**: 0.3
	
	Description:
	
	• **Learning Rate**: Adjusted to 0.001 to allow finer updates to the model weights, improving convergence.
	• **Batch Size**: Increased to 64 to improve the stability of the gradient estimates.
	• **Dropout Rate**: Set to 0.3 to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training.
	
	Screenshot of Code:
	
	Batch Size: 64
	
	`color_mode="rgb",`
	`batch_size=64,`
	`image_size=(128, 128),`
	`shuffle=True,`

<table>
<tr><td></td><td>Learning Rate: 0.001

<pre>model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.001),loss='categorical_crossentropy',metrics=['accuracy'])</pre>
Dropout Rate: 0.3

<pre>model.add(Dropout(0.30))</pre></td></tr>
<tr><td>Model 3</td><td>Accuracy: 0.8942

Tuned Hyperparameters:

1. Learning Rate: 0.00005
2. Batch Size: 128
3. Number of Layers: 4

Description:

• Learning Rate: Reduced to 0.00005 to make very precise adjustments, helping the model learn with high accuracy.
• Batch Size: Increased to 128, allowing the model to use more data at once, which helps stabilize and improve learning.
• Number of Layers: Increased to 5, enabling the model to learn more detailed and complex features.

Screenshot of Code:

Batch Size: 128

<pre>color_mode="rgb",
batch_size=128,
image_size=(128, 128),</pre>
Learning Rate: 0.00005

<pre>model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.00005),loss='categorical_crossentropy',metrics=['accuracy'])</pre></td></tr>
</table>

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model 1	**Accuracy**: Model 1 was the best at identifying tomato plant diseases, with an accuracy of 98.71%. **Tuned Hyperparameters**: 1. **Learning Rate (0.0001)**: A smaller learning rate means the model learns more slowly and accurately, making better adjustments. 2. **Batch Size (32)**: The model updates its learning every 32 samples, balancing learning stability and computing power. 3. **Number of Epochs (15)**: The model goes through all the training data 15 times, which helps it learn well without overfitting. **Additional Factors**: • **Generalization**: The small learning rate and moderate epochs help the model learn effectively without overfitting to the training data. • **Training Stability**: The chosen batch size keeps the learning updates stable, avoiding drastic changes. **Conclusion**: Model 1, with its highest accuracy and well-chosen settings, is the best at detecting tomato plant diseases from leaf images. Its balanced settings make it a strong and reliable choice for the project "Visual Diagnostics: Detecting Tomato Plant Diseases through Leaf Image Analysis."