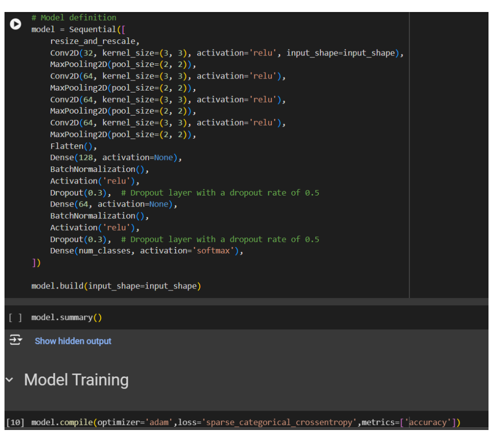
Model Optimization and Tuning Phase Template

Date	20 July 2024
Team ID	SWTID1720082030
Project Title	Hydration Essentials: Classifying Water Bottle Images
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

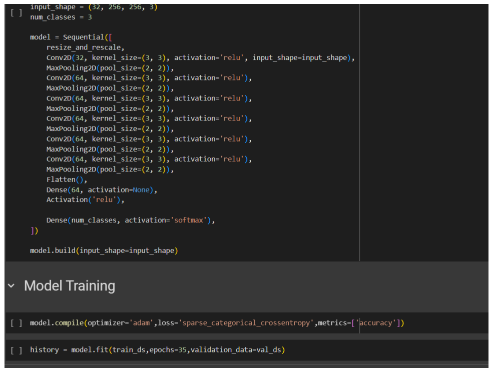
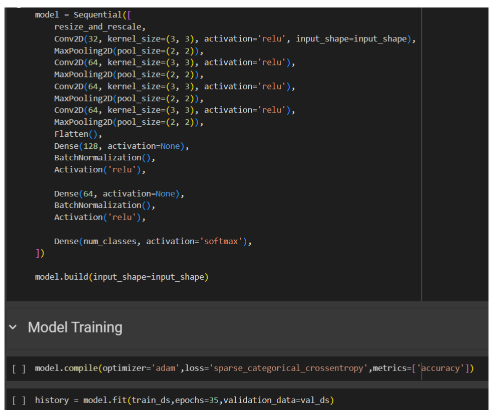
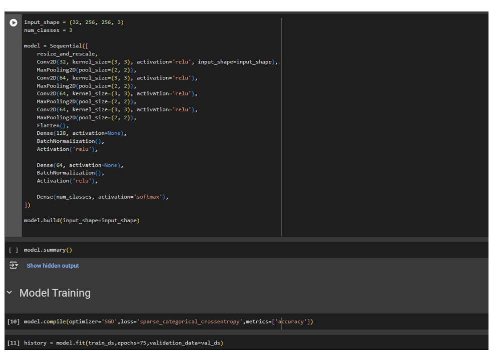
The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (8 Marks):

<table>
<tr><th>Model</th><th>Tuned Hyperparameters</th></tr>
<tr><td>CNN (optimiser: Adam, batch normalisation and dropout)</td><td>It uses Adam optimiser and batch normalisation along with dropout .

<pre><code># Model definition
model = Sequential([
    resize_and_rescale,
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation=None),
    BatchNormalization(),
    Activation('relu'),
    Dropout(0.3), # Dropout layer with a dropout rate of 0.5
    Dense(64, activation=None),
    BatchNormalization(),
    Activation('relu'),
    Dropout(0.3), # Dropout layer with a dropout rate of 0.5
    Dense(num_classes, activation='softmax'),
])

model.build(input_shape=input_shape)

[ ] model.summary()
Show hidden output
Model Training
[10] model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])</code></pre></td></tr>
</table>

<table>
<tr>
<td>CNN (optimizer : Adam, 2 fully connected layers)</td>
<td>It uses Adam optimiser and 2 fully connected layers

</td>
</tr>
<tr>
<td>CNN (optimiser: adam with 3 fully connected layers)</td>
<td>It uses Adam optimiser and 3 fully connected layers

</td>
</tr>
<tr>
<td>CNN (with optimiser: SDG, 3 layers)</td>
<td>It uses SDG optimiser with 3 layers

</td>
</tr>
</table>

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
CNN (with optimiser: SDG, 3 layers)	A CNN with three layers optimized using SGD is a basic yet effective choice for image classification. SGD offers computational efficiency, making it suitable for large datasets. The three-layer architecture provides a balance between model complexity and performance, often yielding good results for simpler image classification tasks. While deeper architectures and more advanced optimizers might outperform this model on complex datasets, its simplicity and efficiency make it a strong baseline.