Model Development Phase Template

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

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code demonstrates the code for the CNN model that we have used .

Initial Model Training Code (5 marks):

```
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'),

    Dense(64, activation=None),
    BatchNormalization(),
    Activation('relu'),

    Dense(num_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

Model Training

```
[ ] model.compile(optimizer='SGD',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

```
[ ] history = model.fit(train_ds,epochs=75,validation_data=val_ds)
```

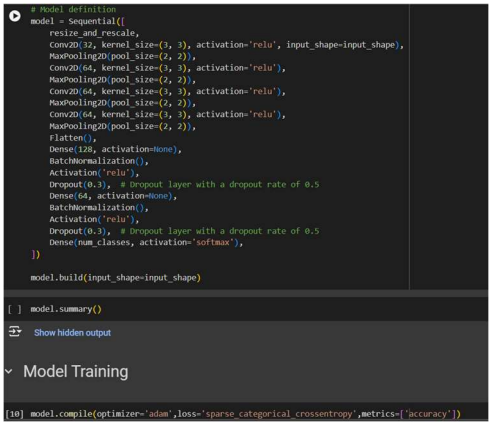
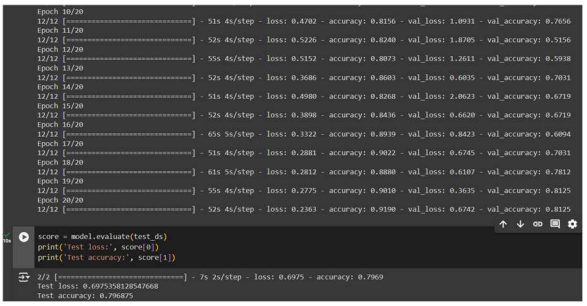
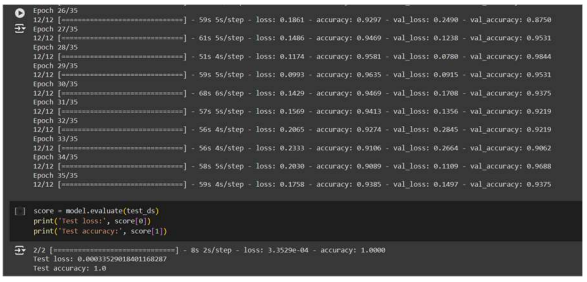
```
Epoch 1/75
12/12 [==============================] - 56s 4s/step - loss: 0.9289 - accuracy: 0.5866 - val_loss: 0.9941 - val_accuracy: 0.6094
Epoch 2/75
12/12 [==============================] - 53s 4s/step - loss: 0.6754 - accuracy: 0.7151 - val_loss: 0.8896 - val_accuracy: 0.7031
Epoch 3/75
12/12 [==============================] - 52s 4s/step - loss: 0.6126 - accuracy: 0.7542 - val_loss: 0.9081 - val_accuracy: 0.6250
Epoch 4/75
12/12 [==============================] - 52s 4s/step - loss: 0.5209 - accuracy: 0.7905 - val_loss: 0.9017 - val_accuracy: 0.5781
Epoch 5/75
12/12 [==============================] - 65s 5s/step - loss: 0.5197 - accuracy: 0.8156 - val_loss: 0.8909 - val_accuracy: 0.6579
Epoch 6/75
12/12 [==============================] - 52s 4s/step - loss: 0.5746 - accuracy: 0.7682 - val_loss: 0.9422 - val_accuracy: 0.5781
Epoch 7/75
12/12 [==============================] - 55s 4s/step - loss: 0.4760 - accuracy: 0.8359 - val_loss: 0.8388 - val_accuracy: 0.5938
Epoch 8/75
12/12 [==============================] - 54s 4s/step - loss: 0.4946 - accuracy: 0.8045 - val_loss: 0.8811 - val_accuracy: 0.5469
```

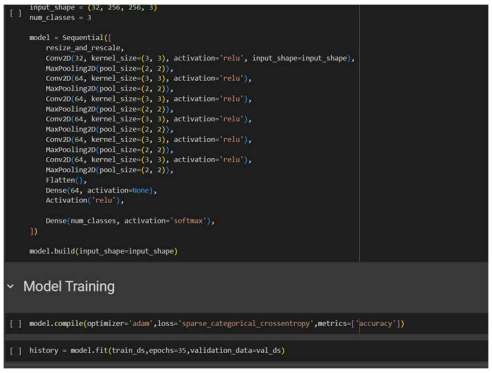
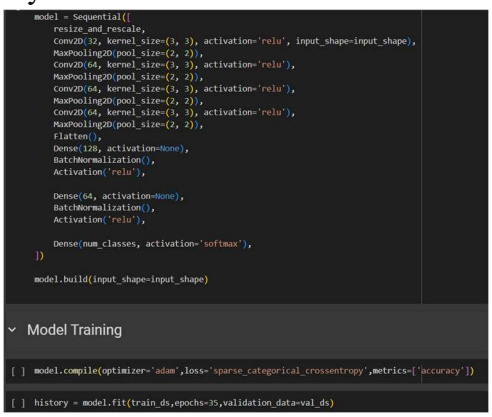
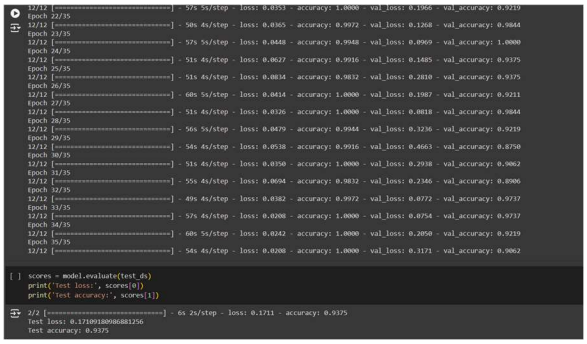
```
Epoch 73/75
12/12 [==============================] - 55s 4s/step - loss: 0.0211 - accuracy: 1.0000 - val_loss: 0.0546 - val_accuracy: 0.9844
Epoch 74/75
12/12 [==============================] - 62s 5s/step - loss: 0.0193 - accuracy: 0.9972 - val_loss: 0.8371 - val_accuracy: 0.7344
Epoch 75/75
12/12 [==============================] - 55s 4s/step - loss: 0.0273 - accuracy: 0.9974 - val_loss: 0.0823 - val_accuracy: 0.9844

[ ] scores = model.evaluate(test_ds)
    print('Test loss:', scores[0])
    print('Test accuracy:', scores[1])

2/2 [==============================] - 5s 1s/step - loss: 0.0427 - accuracy: 1.0000
Test loss: 0.0427468940615654
Test accuracy: 1.0
```

Model Validation and Evaluation Report (5 marks)

Model	Summary	Training and Validation Performance Metrics
CNN	Used batch normalisation and dropout(optimizer:adam) 	
CNN	(optimizer:adam) Only 2 fully connected layers	

		
CNN	(optimizer:adam) 3 fully connected layers 	
CNN	Optimiser: SGD 3 layers