

Project Development Phase

Model Performance Test

Date 17 November 2022
Team ID PNT2022TMID33736
Project Name Project - AI-Powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks 10 marks

Model Performance Testing:

S.No	Parameter Values	Screenshots
------	------------------	-------------

1.	Model Summary -	 <pre> Model: "unsequential_1" Layer (type) Output Shape Param # ----- conv2d (Conv2D) (None, 62, 62, 32) 896 max_pooling2d (MaxPooling2D) (None, 31, 31, 32) 0 conv2d_1 (Conv2D) (None, 29, 29, 32) 9280 max_pooling2d_1 (MaxPooling2D) (None, 14, 14, 32) 0 flatten (Flatten) (None, 6272) 0 dense (Dense) (None, 128) 802944 dense_1 (Dense) (None, 5) 643 Total params: 811,733 Trainable params: 811,733 Non-trainable params: 0 </pre>
----	-----------------	---

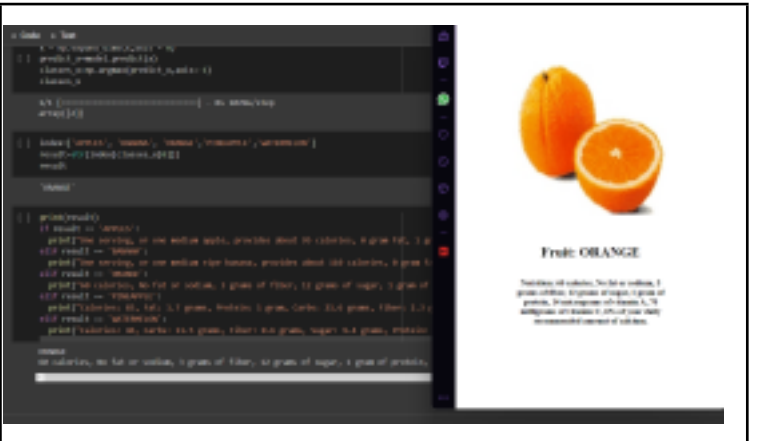
2. Accuracy Training Accuracy - 90.42

Validation Accuracy - 90.42

```

Epoch 10/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 11/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 12/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 13/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 14/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 15/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 16/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 17/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 18/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 19/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 20/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 21/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 22/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 23/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 24/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 25/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 26/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 27/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 28/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 29/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042
Epoch 30/50: 40.000000 - loss: 0.0000 - accuracy: 0.9042 - val_loss: 0.0000 - val_accuracy: 0.9042

```

3.	<p>Confidence Score Class Detected - Apples, Banana, Orange, Pineapple, Watermelon</p> <p>Confidence Score - 90.42 in testing model</p>	 <pre> def get_class_names(class_indices): class_names = ['apple', 'banana', 'orange', 'pineapple', 'watermelon'] return [class_names[i] for i in class_indices] def predict_class(image_path): model = tf.keras.models.load_model('model.h5') image = tf.keras.preprocessing.image.load_img(image_path, target_size=(224, 224)) image = tf.keras.preprocessing.image.array_to_numpy(image) image = image.reshape((1, 224, 224, 3)) image = image.astype('float32') / 255.0 predictions = model.predict(image) predicted_class_index = np.argmax(predictions) predicted_class_name = get_class_names([predicted_class_index]) confidence_score = predictions[predicted_class_index] return predicted_class_name, confidence_score image_path = 'orange.jpg' predicted_class_name, confidence_score = predict_class(image_path) print(predicted_class_name, confidence_score) </pre> <p>orange 0.9042</p>
----	---	---