**AI-Powered Nutrition Analyzer For Fitness Enthusiasts**

| | |
|---|---|
| Date | |
| Team ID | SWTID1749893823 |
| Project Title | AI-Powered Nutrition Analyzer For Fitness Enthusiasts |
| Maximum Marks | 6 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 | In the model, include_top=False removes VGG16's original classifier to allow custom layers. base_model.trainable=False freezes pre-trained layers to retain learned features. Dropout(0.3) prevents overfitting, and Dense(128) controls model complexity.<br><br>`from tensorflow.keras.applications import VGG16`<br><br>`# Load VGG16 without top layer`<br>`base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))`<br>`base_model.trainable = False  # Freeze base`<br><br>`# Add classification head`<br>`model = Sequential([`<br>`  base_model,`<br>`  GlobalAveragePooling2D(),`<br>`  Dropout(0.3),`<br>`  Dense(128, activation='relu'),`<br>`  Dropout(0.3),`<br>`  Dense(5, activation='softmax')`<br>`])`<br><br>`model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])`<br>`model.summary()` |
| **MobileNetV2** | Here's a short paragraph explaining the key parameters in your MobileNetV2-based model:<br>In this model, include_top=False removes MobileNetV2's original classification layers so custom ones can be added. Setting base_model.trainable=False freezes the pre-trained layers to keep learned features intact. Dropout(0.3) helps reduce overfitting, while Dense(128) defines the custom layer's complexity for learning fruit classes effectively. |

```python
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam

# Number of classes (fruits)
NUM_CLASSES = 5

# Load MobileNetV2 without the top layer (include_top=False)
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model.trainable = False  # Freeze the base model

# Build the model
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dropout(0.3),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(NUM_CLASSES, activation='softmax'),
])


# Compile the model
model.compile(optimizer=Adam(),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

**Final Model Selection Justification (2 Marks):**

| Final Model | Reasoning |
|---|---|
| MobileNetV2 | This model was chosen as the final optimized model due to its lightweight architecture, faster training time, and strong performance on image classification tasks with limited computational resources. |