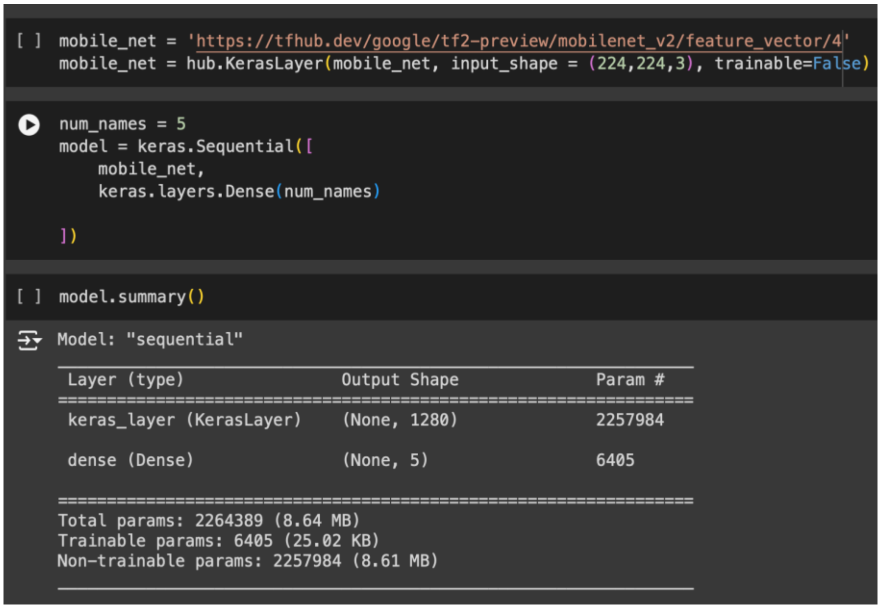
# Model Optimization and Tuning Phase Template

| | |
|---|---|
| Date | 08 July 2024 |
| Team ID | SWTID1720201335 |
| Project Title | Rice Type Classification Using CNN |
| Maximum Marks | 10 Marks |

## Model Optimization and Tuning Phase

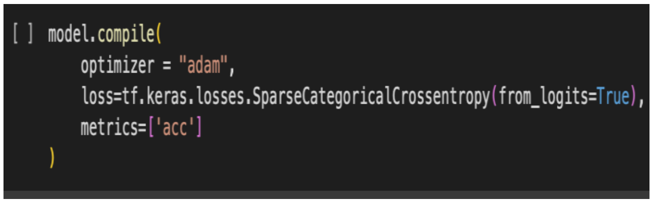
The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

## Hyperparameter Tuning Documentation (8 Marks):

| Model | Tuned Hyperparameters |
|---|---|
| MobileNetV2 | Fully connected layers parameters:<br> |

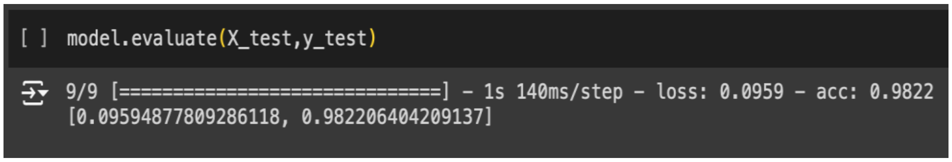
```
[ ]  mobile_net = 'https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4'
     mobile_net = hub.KerasLayer(mobile_net, input_shape = (224,224,3), trainable=False)

     num_names = 5
     model = keras.Sequential([
         mobile_net,
         keras.layers.Dense(num_names)

     ])

[ ]  model.summary()

     Model: "sequential"
     _________________________________________________________________
      Layer (type)                Output Shape              Param #
     =================================================================
      keras_layer (KerasLayer)    (None, 1280)              2257984

      dense (Dense)               (None, 5)                 6405

     =================================================================
     Total params: 2264389 (8.64 MB)
     Trainable params: 6405 (25.02 KB)
     Non-trainable params: 2257984 (8.61 MB)
     _________________________________________________________________
```

Learning process parameters:



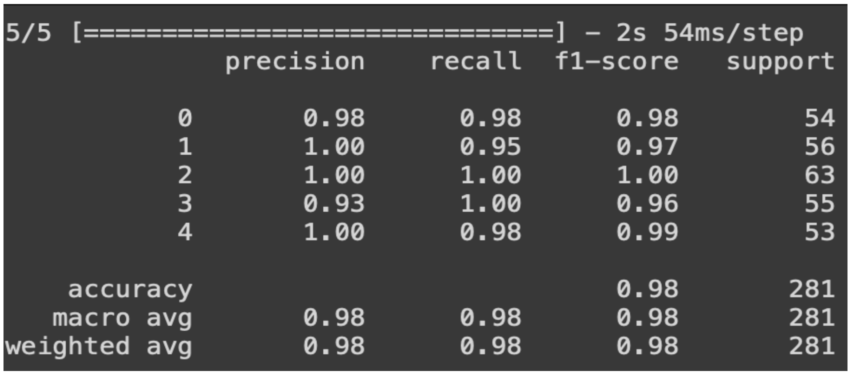
```
[ ] model.compile(
        optimizer = "adam",
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=['acc']
    )
```

Accuracy and loss:

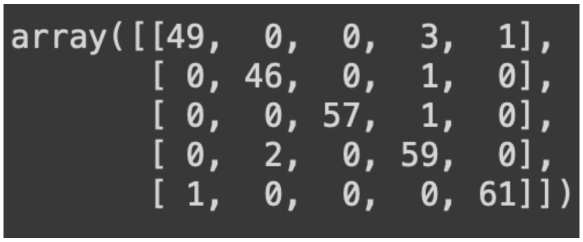


```
[ ] model.evaluate(X_test,y_test)

    9/9 [==============================] - 1s 140ms/step - loss: 0.0959 - acc: 0.9822
    [0.09594877809286118, 0.982206404209137]
```

Optimized metrics:



```
5/5 [==============================] - 2s 54ms/step
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        54
           1       1.00      0.95      0.97        56
           2       1.00      1.00      1.00        63
           3       0.93      1.00      0.96        55
           4       1.00      0.98      0.99        53

    accuracy                           0.98       281
   macro avg       0.98      0.98      0.98       281
weighted avg       0.98      0.98      0.98       281
```



```
array([[49,  0,  0,  3,  1],
       [ 0, 46,  0,  1,  0],
       [ 0,  0, 57,  1,  0],
       [ 0,  2,  0, 59,  0],
       [ 1,  0,  0,  0, 61]])
```

| | |
|---|---|
| Model 2 | -- |

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

| Final Model | Reasoning |
|---|---|
| MobileNetV2 | The MobileNetV2 model was chosen due to its remarkable efficiency and lightweight architecture, making it highly suitable for deployment in environments with limited computational resources. MobileNetV2 achieves a fine balance between accuracy and performance, delivering robust predictions without sacrificing speed. Its use of depth-wise separable convolutions significantly reduces the number of parameters and computational load, ensuring fast inference times while maintaining high accuracy. This makes MobileNetV2 an ideal choice for applications where real-time processing and low latency are critical, perfectly aligning with the project's requirements. |