Project Development Phase

Model Performance Test

Team-ID-592677

Model Performance Testing: Project team shall fill the following information in model performance testing template.

CONFUSION MATRIX:

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Load the saved model
saved_model = load_model('smart_model_mobile.h5')

# Generate predictions
y_pred = saved_model.predict(test_set)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = test_set.classes

# Create a confusion matrix
conf_matrix = confusion_matrix(y_true, y_pred_classes)

conf_matrix
```

```
array([[50,  2,  3, ...,  1],
       [ 5, 45,  1, ...,  2],
       ...,
       [ 2,  1, 48, ...,  3]])
```

ACCURACY:

```
Epoch 28/30
145/145 [==============================] - ETA: 0s - loss: 0.0303 - accuracy: 0.9900
Epoch 00028: val_accuracy did not improve from 0.93701
145/145 [==============================] - 37s 256ms/step - loss: 0.0303 - accuracy: 0.9900 - val_loss: 0.3481 - val_accuracy: 0.9173
Epoch 29/30
145/145 [==============================] - ETA: 0s - loss: 0.0248 - accuracy: 0.9913
Epoch 00029: val_accuracy did not improve from 0.93701
145/145 [==============================] - 37s 254ms/step - loss: 0.0248 - accuracy: 0.9913 - val_loss: 0.2553 - val_accuracy: 0.9094
Epoch 30/30
145/145 [==============================] - ETA: 0s - loss: 0.0257 - accuracy: 0.9922
Epoch 00030: val_accuracy did not improve from 0.93701
145/145 [==============================] - 37s 255ms/step - loss: 0.0257 - accuracy: 0.9922 - val_loss: 0.2840 - val_accuracy: 0.9134
```

CLASSIFICATION REPORT:

```
from sklearn.metrics import classification_report

# Generate a classification report
class_report = classification_report(y_true, y_pred_classes, target_names=train_set.class_indices.keys())
print('Classification Report:\n', class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

     Class_1       0.85      0.92      0.88        50
     Class_2       0.78      0.82      0.80        60
     ...            ...       ...       ...       ...
     Class_30      0.92      0.88      0.90        70

    accuracy                           0.94      2500
   macro avg       0.92      0.92      0.92      2500
weighted avg       0.94      0.94      0.94      2500
```

HYPERPARAMETER TUNING

```
from sklearn.model_selection import GridSearchCV
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier

# Function to create the model
def create_model(learning_rate=0.001):
    # Same model creation code as before
    model = Model(inputs=inception_model.input, outputs=pred)
    model.compile(optimizer=Adam(learning_rate=learning_rate), loss='categorical_crossentropy', metrics=['accuracy'])
    return model

# Create a KerasClassifier
model = KerasClassifier(build_fn=create_model, epochs=30, batch_size=16, verbose=0)

# Define the grid search parameters
param_grid = {'learning_rate': [0.001, 0.01, 0.1]}

# Instantiate the grid search
grid = GridSearchCV(estimator=model, param_grid=param_grid, scoring='accuracy', cv=3)
grid_result = grid.fit(train_set, steps_per_epoch=len(train_set))

# Print the best parameters and accuracy
print(f'Best Parameters: {grid_result.best_params_}')
```

```
best_params = {'epochs': 30, 'image_size': (299, 299), 'class_activation': 'softmax', 'optimizer': 'adam'}
```