Image Classification with MobileNet

You will use the MobileNet architecture, a lightweight deep neural network designed for mobile and edge devices, to perform image classification tasks. The goal is to understand the practical aspects of working with pre-trained models, fine-tuning them, and deploying them for a specific image classification task.

## Dataset:

You can choose one of the following datasets for this assignment:

* CIFAR-10
* CIFAR-100

## Understanding MobileNet Architecture

* Research and write a summary (300-500 words) on the MobileNet architecture, focusing on its design principles, advantages, and use cases.

## Setting Up the Environment

* Ensure TensorFlow is installed in your Python environment.
* If not installed, use pip install Tensorflow to install it.

## Data Preprocessing

* Load your chosen dataset.
* Perform necessary preprocessing steps like normalization, resizing images to match the input size of MobileNet, and data augmentation techniques.

## Model Building and Transfer Learning

* Load the pre-trained MobileNet model from TensorFlow's Keras applications module.
* Remove the top layer and add custom layers suitable for your dataset. Include at least one fully connected layer and the output layer with appropriate activation functions.
* Freeze the weights of the pre-trained layers and train only the custom layers initially.

## Model Training and Fine-Tuning

* Train your model on the training dataset.
* After a few epochs, unfreeze some of the top layers of MobileNet and continue training. This process is known as fine-tuning.
* Experiment with different numbers of layers unfrozen and observe the impact on performance.

## Model Evaluation

* Evaluate the performance of your model on the test dataset.
* Provide a detailed analysis of the results, including accuracy and loss plots.

## Discussion

* Discuss the challenges faced during the assignment.
* Compare the performance of the MobileNet model with a more complex model (like ResNet50 or VGG16) on the same dataset, if applicable.

**Deliverables:**

* A comprehensive report including all the above tasks.
* Source code in a Jupyter Notebook or Python script format.