```
# STEP 2: Import Libraries
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
# STEP 3: Load and Preprocess Dataset (Limited to 5,000 Samples)
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

#  Use only 5,000 samples to reduce load time
x_train, y_train = x_train[:5000], y_train[:5000]

# Resize images from 32x32 to 96x96
x_train = tf.image.resize(x_train, (96, 96)) / 255.0
x_test = tf.image.resize(x_test, (96, 96)) / 255.0

# One-hot encode labels
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
# STEP 4: Load Pretrained MobileNetV2
base_model = MobileNetV2(
    input_shape=(96, 96, 3),
    include_top=False,
    weights='imagenet'
)
base_model.trainable = False  # Freeze base layers
# STEP 5: Add Custom Layers
model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(10, activation='softmax')
])
# STEP 6: Compile and Train (Only 3 Epochs)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(
    x_train, y_train,
    epochs=3,              #  Reduced epochs
    batch_size=64,
    validation_data=(x_test, y_test),
    verbose=2
)
```

OUTPUT

```
Epoch 1/3
79/79 - 115s - 1s/step - accuracy: 0.5034 - loss: 1.4663 - val_accuracy: 0.6782 -
val_loss: 0.9523
Epoch 2/3
79/79 - 118s - 1s/step - accuracy: 0.6870 - loss: 0.9312 - val_accuracy: 0.7206 -
val_loss: 0.8136
Epoch 3/3
79/79 - 117s - 1s/step - accuracy: 0.7374 - loss: 0.7702 - val_accuracy: 0.7330 -
val_loss: 0.7789
<keras.src.callbacks.history.History at 0x79e7cd258990>
```