In [1]:

```
# a. Load libraries
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import MobileNetV2
```

In [2]:

```
# Load an existing dataset
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
```

In [3]:

```
# Normalize values
x_train = x_train / 255.0
x_test = x_test / 255.0
```

In [4]:

```
# b. Load a pre-trained CNN model
base_model = MobileNetV2(
    weights="imagenet",
    include_top=False,
    input_shape=(96, 96, 3)
# a. Load libraries
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import MobileNetV2
# Load an existing dataset
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()

)
```

In [5]:

```
# Resize CIFAR-10 images to MobileNet size
x_train = tf.image.resize(x_train, (96, 96))
x_test = tf.image.resize(x_test, (96, 96))
```

In [6]:

```
# Freeze lower layers
base_model.trainable = False
```

In [7]:

```
model = models.Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(10, activation='softmax')
])
```

In [8]:

```
# d. Compile and train classifier layers
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

In [9]:

```
model.fit(x_train, y_train, epochs=3, validation_split=0.2)
```

```
Epoch 1/3
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 258s 199ms/step - accuracy: 0.6580 - loss: 1.0156 - val_a
ccuracy: 0.7754 - val_loss: 0.6734
Epoch 2/3
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 292s 233ms/step - accuracy: 0.7411 - loss: 0.7504 - val_a
ccuracy: 0.7819 - val_loss: 0.6310
Epoch 3/3
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 248s 198ms/step - accuracy: 0.7673 - loss: 0.6658 - val_a
ccuracy: 0.8010 - val_loss: 0.5900
Out[9]:
<keras.src.callbacks.history.History at 0x19feb814ad0>
```

In [10]:

```
# e. Fine-tune: unfreeze some of the deeper layers
base_model.trainable = True
```

In [11]:

```
model.compile(optimizer=tf.keras.optimizers.Adam(1e-5),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

In [12]:

```
model.fit(x_train, y_train, epochs=2, validation_split=0.2)
```

```
Epoch 1/2
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 961s 731ms/step - accuracy: 0.7169 - loss: 0.9758 - val_a
ccuracy: 0.8188 - val_loss: 0.5553
Epoch 2/2
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 909s 728ms/step - accuracy: 0.8053 - loss: 0.6202 - val_a
ccuracy: 0.8597 - val_loss: 0.4390
Out[12]:
<keras.src.callbacks.history.History at 0x19fea166850>
```

In [13]:

```
test_loss, test_acc = model.evaluate(x_test, y_test)
print("Test Accuracy:", test_acc)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 53s 169ms/step - accuracy: 0.8537 - loss: 0.4563
Test Accuracy: 0.8536999821662903
```

In [14]:

```
import numpy as np
from tensorflow.keras.preprocessing import image
```

In [15]:

```
class_names = ["airplane", "car", "bird", "cat", "deer",
               "dog", "frog", "horse", "ship", "truck"]
```

In [19]:

```
img_path = "horse.jpg"
img = image.load_img(img_path, target_size=(96, 96))
img_array = image.img_to_array(img)
img_array = img_array / 255.0
img_array = np.expand_dims(img_array, axis=0)

# Predict
pred = model.predict(img_array)
pred_class = np.argmax(pred)
```

```
print("Prediction Vector:", pred)
print("Predicted Class Index:", pred_class)
print("Predicted Label:", class_names[pred_class])
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step
Prediction Vector: [[5.8336701e-02 2.2478106e-04 1.5201515e-01 1.3921638e-01 1.6248260e-
01
  1.5945232e-01 1.0623984e-02 3.1492248e-01 1.9645065e-03 7.6108793e-04]]
Predicted Class Index: 7
Predicted Label: horse
```

In [ ]: