◈ Rice Grain Classification - Jupyter Notebook Version

This notebook walks through the process of building a rice grain classifier using transfer learning (MobileNetV2).

◈ Step 1: Install Required Libraries

In [ ]:
```
!pip install tensorflow pillow numpy matplotlib
```

◈ Step 2: Download Dataset

1. Download the dataset from Kaggle: ◈ https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset
2. Extract the downloaded ZIP file.
3. Rename the extracted folder to `rice_dataset/` and place it in the same directory as this notebook.

◈ Step 3: Load & Preprocess Dataset

In [ ]:
```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = (224, 224)
BATCH_SIZE = 32

datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_gen = datagen.flow_from_directory('rice_dataset',
                                        target_size=IMG_SIZE,
                                        batch_size=BATCH_SIZE,
                                        subset='training')

val_gen = datagen.flow_from_directory('rice_dataset',
                                      target_size=IMG_SIZE,
                                      batch_size=BATCH_SIZE,
                                      subset='validation')

class_names = list(train_gen.class_indices.keys())
```

```
print("Classes:", class_names)
```

◈ Step 4: Create and Train the Model

In [ ]:
```
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras import layers, models
from tensorflow.keras.callbacks import EarlyStopping

base_model = MobileNetV2(input_shape=IMG_SIZE + (3,), include_top=False, weigh
base_model.trainable = False

model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(len(class_names), activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc

model.fit(train_gen, validation_data=val_gen, epochs=5,
          callbacks=[EarlyStopping(patience=2, restore_best_weights=True)])
```

◈ Step 5: Save the Model

In [ ]:
```
model.save("rice_model.h5")
```

◈ Step 6: Upload Image and Predict

In [ ]:
```
from PIL import Image
import numpy as np
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
from IPython.display import display
import ipywidgets as widgets

uploader = widgets.FileUpload(accept='image/*', multiple=False)
display(uploader)
```

In [ ]:
```
upload_key = list(uploader.value.keys())[0]
img_bytes = uploader.value[upload_key]['content']
```

```
with open("test.jpg", "wb") as f:
    f.write(img_bytes)

img_path = "test.jpg"
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img) / 255.0
x = np.expand_dims(x, axis=0)

pred = model.predict(x)[0]
idx = np.argmax(pred)

plt.imshow(img)
plt.axis('off')
plt.title(f"Prediction: {class_names[idx]} ({round(pred[idx]*100, 2)}%)")
plt.show()
```