

```

# -A-Deep-Learning-Odyssey-in-Rice-Type-Classification
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt

# Simulated dataset parameters
# Define the parameters for the simulated dataset
NUM_CLASSES = 5 # Example: 5 types of rice
IMG_HEIGHT = 128
IMG_WIDTH = 128
NUM_SAMPLES = 1000 # Number of training samples
BATCH_SIZE = 32 # Define BATCH_SIZE here

num_classes = NUM_CLASSES # e.g., Basmati, Jasmine, Arborio, Brown,
White
img_height, img_width = IMG_HEIGHT, IMG_WIDTH
num_train = NUM_SAMPLES
num_val = 100

# Generate random synthetic image data and labels
X_train = np.random.rand(num_train, img_height, img_width, 3)
y_train = tf.keras.utils.to_categorical(np.random.randint(0, num_classes,
size=(num_train,)), num_classes)

X_val = np.random.rand(num_val, img_height, img_width, 3)
y_val = tf.keras.utils.to_categorical(np.random.randint(0, num_classes,
size=(num_val,)), num_classes)

# Build the CNN model
model = models.Sequential([
    layers.Input(shape=(img_height, img_width, 3)),
    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, validation_data=(X_val, y_val),
epochs=5, batch_size=BATCH_SIZE)

# Plot training & validation accuracy
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Rice Type Classification (Simulated)')
plt.xlabel('Epoch')

```

```
plt.ylabel('Accuracy')  
plt.legend()  
plt.show()
```