```
import zipfile
import os

# Correct dataset path
dataset_zip = "/content/dataset.zip"  # This matches your Colab file location
extract_path = "/content/brain_tumor_dataset"  # Extracted folder

# Extract dataset
with zipfile.ZipFile(dataset_zip, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print("Dataset extracted successfully!")
```

```
Dataset extracted successfully!
```

```
import os
import cv2
import numpy as np

# Define dataset path (Modify this based on your dataset location)
dataset_path = "/content/brain_tumor_dataset"  # Update based on your dataset location

# Define categories based on your dataset folder structure
categories = ["yes", "no"]  # Update to match your dataset structure

# Initialize empty lists to store images and labels
images = []
labels = []

# Loop through each category folder
for category in categories:
    path = os.path.join(dataset_path, category)  # Create full path
    class_index = categories.index(category)  # Assign index (0 for "no", 1 for "yes")

    # Check if the folder exists to avoid errors
    if not os.path.exists(path):
        print(f"Warning: {path} does not exist. Skipping...")
        continue

    # Process all images in the folder
    for img_name in os.listdir(path):
        img_path = os.path.join(path, img_name)  # Full path to image

        # Read the image in grayscale
        img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

        # Ensure the image is read correctly
        if img is None:
            print(f"Warning: Unable to read {img_path}. Skipping...")
            continue

        # Resize the image to 128x128
        img = cv2.resize(img, (128, 128))

        # Append to lists
        images.append(img)
        labels.append(class_index)

# Convert lists to numpy arrays
images = np.array(images).reshape(-1, 128, 128, 1)  # Reshape for CNN input
images = images / 255.0  # Normalize pixel values
labels = np.array(labels)

# Print summary
print(f"Dataset Loaded: {len(images)} images from {len(categories)} categories")
print(f"Image shape: {images.shape}")
print(f"Labels shape: {labels.shape}")
```

```
Dataset Loaded: 253 images from 2 categories
Image shape: (253, 128, 128, 1)
Labels shape: (253,)
```

Start coding or generate with AI.