

✓ Split_dataset_augmented_images1_restorable_spilt

This block of code splits a dataset of images into training, validation, and test sets. The split ratios are 70% for training, 20% for validation, and 10% for testing.

✓ Explanation

1. Importing Libraries:

- `os`: Used for handling file and directory operations.
- `shutil`: Used for high-level file operations like copying and moving files.
- `random`: Used for shuffling the list of images.

2. Defining Paths:

- `input_dir`: Path to the directory containing the original dataset of images.
- `output_dir`: Path to the directory where the split dataset will be stored.
- `train_dir`, `val_dir`, `test_dir`: Paths to the subdirectories for training, validation, and testing datasets.

3. Creating Directories:

- A loop is used to create the `train_dir`, `val_dir`, and `test_dir` directories if they do not already exist.

4. Listing and Shuffling Images:

- `all_images`: A list of all image filenames in the `input_dir`.
- `random.shuffle(all_images)`: Shuffles the list of image filenames to ensure random distribution.

5. Defining Split Points:

- `train_split`: Index where the training set ends (70% of the dataset).
- `val_split`: Index where the validation set ends (90% of the dataset), leaving the last 10% for testing.

6. Splitting the Images:

- `train_images`: List of image filenames for the training set.
- `val_images`: List of image filenames for the validation set.
- `test_images`: List of image filenames for the test set.


7. Moving Images:

- The images are moved from the `input_dir` to the respective directories (`train_dir`, `val_dir`, `test_dir`) based on the splits.

Comments in the Code

I added comments in the code to explain each step, providing clarity on the purpose of each block and line of code. The comments describe the process of defining paths, creating directories, shuffling and splitting the dataset, and moving files to the appropriate directories.

```
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive

```
import os # For file and directory handling
import shutil # For moving files
import random # For shuffling the dataset
```

- ✓ Define paths for the input directory and the output directories for training, validation, and testing datasets

```
input_dir = '/content/drive/MyDrive/A new/BK Non restorable final'
output_dir = '/content/drive/MyDrive/dataset_update_augment/final N_update_split'
train_dir = os.path.join(output_dir, 'train')
val_dir = os.path.join(output_dir, 'validation')
test_dir = os.path.join(output_dir, 'test')
```

- ✓ Create directories if they do not exist

```
for dir_path in [train_dir, val_dir, test_dir]:
    if not os.path.exists(dir_path):
        os.makedirs(dir_path)
```

- ✓ List all images in the input directory and shuffle them

```
# Split data # Move images to the respective directories
all_images = os.listdir(input_dir)
random.shuffle(all_images)

train_split = int(0.7 * len(all_images))
val_split = int(0.9 * len(all_images))

train_images = all_images[:train_split]
val_images = all_images[train_split:val_split]
test_images = all_images[val_split:]

for image in train_images:
    shutil.move(os.path.join(input_dir, image), train_dir)
for image in val_images:
    shutil.move(os.path.join(input_dir, image), val_dir)
for image in test_images:
    shutil.move(os.path.join(input_dir, image), test_dir)
```

✓ Split_dataset_augmented_images1_Non_restorable_spilt

```
import os
import shutil
import random
```

```
# Define paths
input_dir = '/content/drive/MyDrive/dataset_update_augment/BK Non restorable fi
output_dir = '/content/drive/MyDrive/dataset_update_augment/BK Non restorable f
train_dir = os.path.join(output_dir, 'train')
val_dir = os.path.join(output_dir, 'validation')
test_dir = os.path.join(output_dir, 'test')
```

```
# Create directories
for dir_path in [train_dir, val_dir, test_dir]:
    if not os.path.exists(dir_path):
        os.makedirs(dir_path)
```

```
# Split data
all_images = os.listdir(input_dir)
random.shuffle(all_images)

train_split = int(0.7 * len(all_images))
val_split = int(0.9 * len(all_images))

train_images = all_images[:train_split]
val_images = all_images[train_split:val_split]
test_images = all_images[val_split:]

for image in train_images:
    shutil.move(os.path.join(input_dir, image), train_dir)
for image in val_images:
    shutil.move(os.path.join(input_dir, image), val_dir)
for image in test_images:
    shutil.move(os.path.join(input_dir, image), test_dir)
```