

Dataset: Rock Paper Scissors Image Dataset from Kaggle Link :

<https://www.kaggle.com/drgfreeman/rockpaperscissors>

Classes: paper, rock, scissors

Model: CNN

Split: 80% Training, 10% Validation, 10% Testing

Augmentation: Flip, Rotation, Zoom

```
from google.colab import files
```

```
uploaded = files.upload()
```

archive (7).zip

archive (7).zip(application/x-zip-compressed) - 320716788 bytes, last modified: 7/4/2026 - 100% done
Saving archive (7).zip to archive (7).zip

Un Zipfile

```
!unzip -q "archive (7).zip" -d dataset
```

```
dataset_path = "/content/dataset/rps-cv-images"
```

```
!!ls dataset/rps-cv-images
```

```
paper  README_rpc-cv-images.txt  rock  scissors
```

Import Required Libraries

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

Create Dataset Folders

```
base_dir = "/content/split_dataset"

train_dir = os.path.join(base_dir, "train")
val_dir = os.path.join(base_dir, "validation")
test_dir = os.path.join(base_dir, "test")

for folder in [train_dir, val_dir, test_dir]:
    os.makedirs(folder, exist_ok=True)
```

```
print("Folders created successfully.")
```

```
Folders created successfully.
```

Define the Image Classes

```
classes = ["paper", "rock", "scissors"]
```

```
print(classes)
```

```
['paper', 'rock', 'scissors']
```

Shuffle the Images

```
random.shuffle(images)
```

```
print("Images shuffled successfully.")
```

```
Images shuffled successfully.
```

Split the Dataset

```
train_size = int(len(images) * 0.8)
```

```
val_size = int(len(images) * 0.1)
```

```
train_images = images[:train_size]
```

```
val_images = images[train_size:train_size + val_size]
```

```
test_images = images[train_size + val_size:]
```

Copy Images to the New Folders

```
for folder, image_list in zip(
    [train_dir, val_dir, test_dir],
    [train_images, val_images, test_images]
):

    class_folder = os.path.join(folder, cls)

    os.makedirs(class_folder, exist_ok=True)

    for image in image_list:

        shutil.copy(
            os.path.join(source, image),
            os.path.join(class_folder, image)
        )
```

Split All Classes

```
for cls in classes:

    source = os.path.join(dataset_path, cls)

    images = os.listdir(source)

    random.shuffle(images)

    train_size = int(len(images) * 0.8)
    val_size = int(len(images) * 0.1)

    train_images = images[:train_size]
    val_images = images[train_size:train_size + val_size]
    test_images = images[train_size + val_size:]

    for folder, image_list in zip(
        [train_dir, val_dir, test_dir],
        [train_images, val_images, test_images]
    ):

        class_folder = os.path.join(folder, cls)

        os.makedirs(class_folder, exist_ok=True)

        for image in image_list:

            shutil.copy(
                os.path.join(source, image),
                os.path.join(class_folder, image)
            )

print("Dataset Split Completed Successfully!")
```

Dataset Split Completed Successfully!

```
print("Dataset Split Completed Successfully!")
```

Dataset Split Completed Successfully!

