

DATA COLLECTION

Date	10 November 2022
Team ID	PNT2022TMID50713
Project Name	A gesture-based tool for sterile browsing of radiology images

ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

Download The Dataset

The dataset contains six classes:

0, 1, 2, 3, 4 and 5

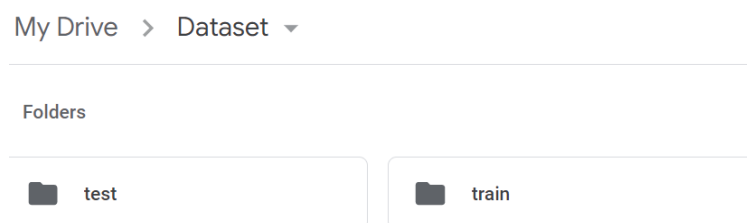


Fig. 1.1 Dataset containing Test and Train

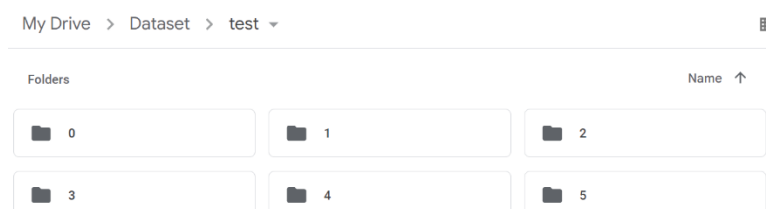


Fig. 1.2 Test Dataset

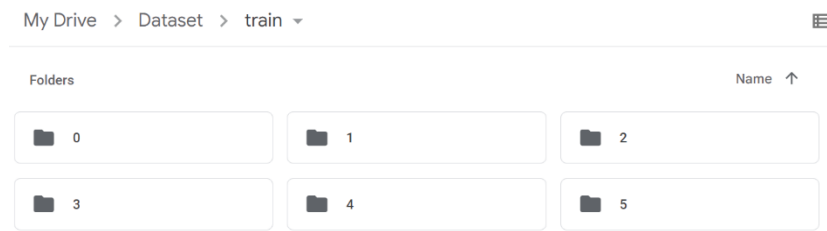


Fig. 1.3 Train Dataset

Image Pre-processing

In this milestone we will be improving the image data that suppresses unwilling distortions or enhances some image features important for further processing, although perform some geometric transformations of images like rotation, scaling, translation etc.

Import The ImageDataGenerator Library

- Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.
- The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.
- Let us import the ImageDataGenerator class from keras

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Configure ImageDataGenerator Class

- ImageDataGenerator class is instantiated and the configuration for the types of data augmentation.
- There are five main types of data augmentation techniques for image data; specifically:
 - Image shifts via the width_shift_range and height_shift_range arguments.
 - Image flips via the horizontal_flip and vertical_flip arguments.
 - Image rotations via the rotation_range argument
 - Image brightness via the brightness_range argument.
 - Image zoom via the zoom_range argument.
 - An instance of the ImageDataGenerator class can be constructed for train and test.

```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,  
zoom_range=0.2, horizontal_flip=True)  
test_datagen=ImageDataGenerator(rescale=1./255)
```

Apply ImageDataGenerator Functionality To Trainset and Testset

- Let us apply ImageDataGenerator functionality to Trainset and Testset by using the following code
- For Trainingset using flow_from_directory function.

- This function will return batches of images from the subdirectories 0, 1, 2, 3, 4, 5 together with labels 0 to 5 {'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}.

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/  
Dataset/train',
```

```
target_size = (64,64), batch_size = 5, class_mode = 'categorical')
```

```
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/  
Dataset/test',
```

```
target_size = (64,64), batch_size = 8, class_mode = 'categorical')
```

Found 594 images belonging to 6 classes.

Found 30 images belonging to 6 classes.