Sprint-1
Dataset Collection & Image Preprocessing

Date	23 October 2022
Team ID	PNT2022TMID52880
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Tasks

There are two tasks:

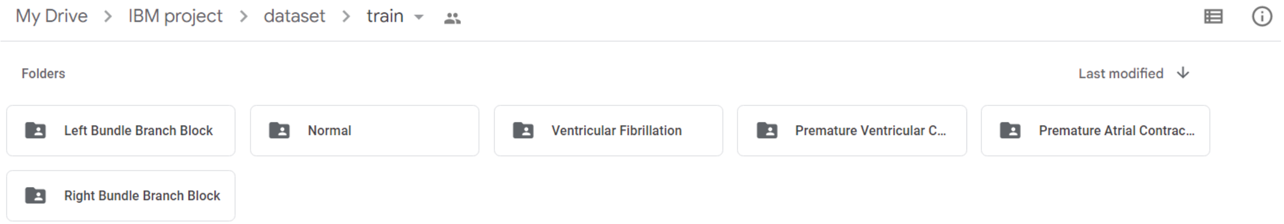
1. Dataset Collection
2. Image Preprocessing

Dataset Collection:

The dataset contains six classes:

1. Left Bundle Branch Block
2. Normal
3. Premature Atrial Contraction
4. Premature Ventricular Contractions
5. Right Bundle Branch Block
6. Ventricular Fibrillation

Training folder:


My Drive > IBM project > dataset > train
Folders
Last modified
Left Bundle Branch Block
Normal
Ventricular Fibrillation
Premature Ventricular C...
Premature Atrial Contrac...
Right Bundle Branch Block

Test folder:

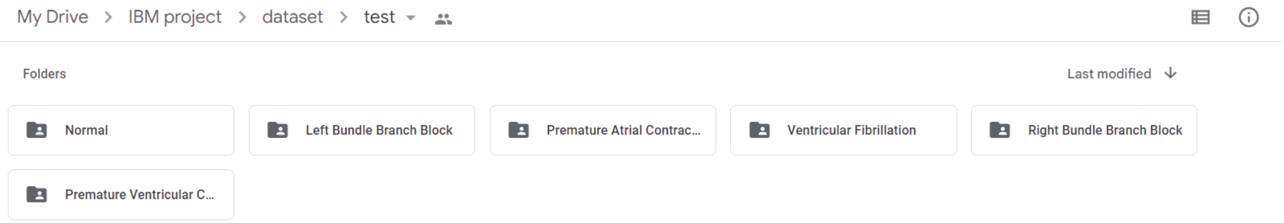

My Drive > IBM project > dataset > test
Folders
Last modified
Normal
Left Bundle Branch Block
Premature Atrial Contrac...
Ventricular Fibrillation
Right Bundle Branch Block
Premature Ventricular C...

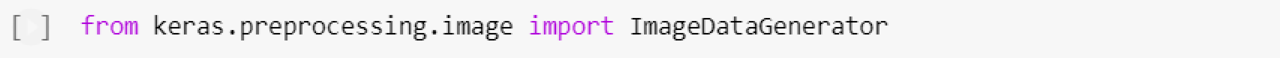
Image Preprocessing:

Image Pre-processing includes the following main tasks

1. Import ImageDataGenerator Library
2. Configure ImageDataGenerator Class
3. Apply ImageDataGenerator functionality to the trainset and testset

Import 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.

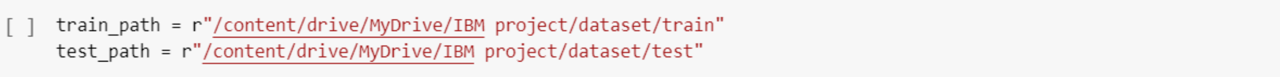


```
[ ]  from keras.preprocessing.image import ImageDataGenerator
```

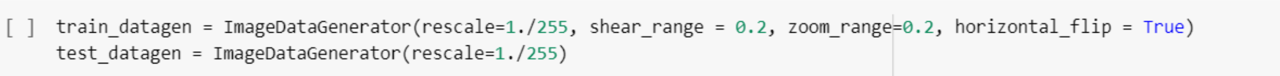
Configure ImageDataGenerator Class:

There are five main types of data augmentation techniques for image data, specifically:

1. Image shifts via the width_shift_range and height_shift_range arguments.
2. Image flips via the horizontal_flip and vertical_flip arguments.
3. Image rotates via the rotation_range argument.
4. Image brightness via the brightness_range argument.
5. Image zooms via the zoom_range argument.



```
[ ]  train_path = r"/content/drive/MyDrive/IBM project/dataset/train"
     test_path = r"/content/drive/MyDrive/IBM project/dataset/test"
```



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

An instance of the ImageDataGenerator class can be constructed for train and test.

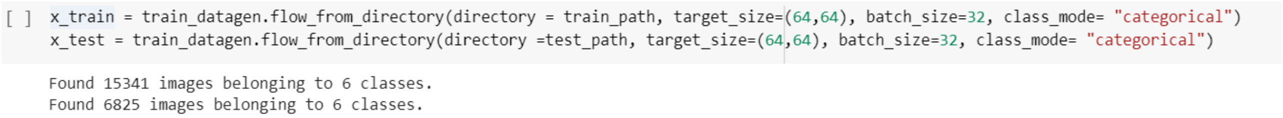
Apply ImageDataGenerator functionality to the trainset and testset:

We will apply ImageDataGenerator functionality to Trainset and Testset by using the following code.

This function will return batches of images from the subdirectories Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, Right Bundle Branch Block and Ventricular Fibrillation, together with labels 0 to 5

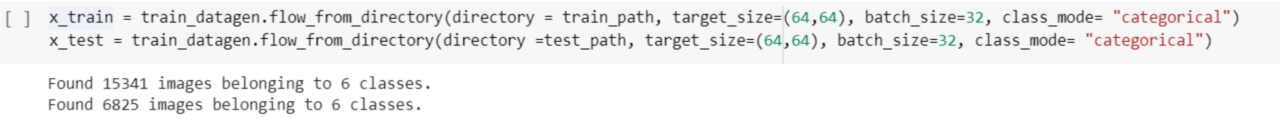
{'Left Bundle Branch Block': 0, 'Normal': 1, 'Premature Atrial Contraction': 2, 'Premature Ventricular Contractions': 3, 'Right Bundle Branch Block': 4, 'Ventricular Fibrillation': 5}

We can see that for training there are 15341 images belonging to 6 classes and for testing there are 6825 images belonging to 6 classes.



```
[ ]  x_train = train_datagen.flow_from_directory(directory = train_path, target_size=(64,64), batch_size=32, class_mode= "categorical")
     x_test = train_datagen.flow_from_directory(directory =test_path, target_size=(64,64), batch_size=32, class_mode= "categorical")

     Found 15341 images belonging to 6 classes.
     Found 6825 images belonging to 6 classes.
```



```
[ ]  x_train = train_datagen.flow_from_directory(directory = train_path, target_size=(64,64), batch_size=32, class_mode= "categorical")
     x_test = train_datagen.flow_from_directory(directory =test_path, target_size=(64,64), batch_size=32, class_mode= "categorical")

     Found 15341 images belonging to 6 classes.
     Found 6825 images belonging to 6 classes.
```