

In [1]:

```
#Image Augmentation
from tensorflow.keras.preprocessing.image import ImageDataGenerator
rain_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
```

In [2]:

```
#Create Model
#Importing req library
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

In [3]:

```
#Add Layers
model = Sequential() # Initializing sequential model
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(64,64,3))) # convolution
layer
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(300, activation='relu')) # Hidden layer 1
model.add(Dense(150, activation='relu')) # Hidden layer 2
model.add(Dense(4, activation='softmax')) # Output layer
```

In [4]:

```
#Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

In [5]:

```
#Save the model
model.save('ECG.h5')
```

In [7]:

```
#Test the model
#Testing 1
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img('/content/fig_2113.png', target_size=(64,64)) # Reading image
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
pred = np.argmax(model.predict(x))
op = ['Normal', 'Left Bundle Branch Block', 'Right Bundle Branch Block', 'Premature Atrial
Contraction', 'Premature Ventricular Contraction', 'Ventricular Fibrillation']
op[pred]
```

1/1 [=====] - 0s 25ms/step

Out[7]:

'Normal'

In [9]:

```
#Test the model
#Testing 2
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img('/content/fig_5898.png', target_size=(64,64)) # Reading image
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
pred = np.argmax(model.predict(x))
op = ['Left Bundle Branch Block', 'Normal', 'Right Bundle Branch Block', 'Premature Atrial
```

```
Contraction', 'Premature Ventricular Contraction', 'Ventricular Fibrillation']  
op[pred]
```

1/1 [=====] - 0s 27ms/step

Out[9]:

'Left Bundle Branch Block'

In [10]:

```
#Test the model  
#Testing 3  
from tensorflow.keras.preprocessing import image  
import numpy as np  
img = image.load_img('/content/fig_12.png', target_size=(64,64)) # Reading image  
x = image.img_to_array(img)  
x = np.expand_dims(x, axis=0)  
pred = np.argmax(model.predict(x))  
op = ['Right Bundle Branch Block', 'Normal', 'Left Bundle Branch Block', 'Premature Atrial  
Contraction', 'Premature Ventricular Contraction', 'Ventricular Fibrillation']  
op[pred]
```

1/1 [=====] - 0s 25ms/step

Out[10]:

'Right Bundle Branch Block'

In [12]:

```
#Test the model  
#Testing 4  
from tensorflow.keras.preprocessing import image  
import numpy as np  
img = image.load_img('/content/fig_24.png', target_size=(64,64)) # Reading image  
x = image.img_to_array(img)  
x = np.expand_dims(x, axis=0)  
pred = np.argmax(model.predict(x))  
op = ['Premature Atrial Contraction', 'Normal', 'Left Bundle Branch Block', 'Right Bundle B  
ranch Block', 'Premature Ventricular Contraction', 'Ventricular Fibrillation']  
op[pred]
```

1/1 [=====] - 0s 24ms/step

Out[12]:

'Premature Atrial Contraction'

In [13]:

```
#Test the model  
#Testing 5  
from tensorflow.keras.preprocessing import image  
import numpy as np  
img = image.load_img('/content/fig_5656.png', target_size=(64,64)) # Reading image  
x = image.img_to_array(img)  
x = np.expand_dims(x, axis=0)  
pred = np.argmax(model.predict(x))  
op = ['Premature Ventricular Contraction', 'Normal', 'Left Bundle Branch Block', 'Right Bun  
dle Branch Block', 'Premature Atrial Contraction', 'Ventricular Fibrillation']  
op[pred]
```

1/1 [=====] - 0s 25ms/step

Out[13]:

'Premature Ventricular Contraction'

In [14]:

```
#Test the model  
#Testing 6  
from tensorflow.keras.preprocessing import image
```

```
import numpy as np
img = image.load_img('/content/VFEfig_122.png',target_size=(64,64)) # Reading image
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['Ventricular Fibrillation','Normal','Left Bundle Branch Block','Right Bundle Branch Block','Premature Atrial Contraction','Premature Ventricular Contraction']
op[pred]
```

1/1 [=====] - 0s 31ms/step

Out[14]:

'Ventricular Fibrillation'

In [15]:

```
#Train the model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLRonPlateau
```

In [16]:

```
early_stop = EarlyStopping(monitor='val_accuracy',
                           patience=5)

lr = ReduceLRonPlateau(monitor='val_accuaracy',
                      factor=0.5,
                      min_lr=0.00001)

callback = [early_stop,lr]
```