**Team ID :** PNT2022TMID30503

**Date : 19 November 2022**

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.

#Extracting Data

!unzip "/content/drive/MyDrive/IBM Project Development/Classification of Arrhythmia by Using inflating: data/train/Ventricular Fibrillation/VFEfig\_468.png inflating: data/train/Ventricular Fibrillation/VFEfig\_469.png inflating: data/train/Ventricular Fibrillation/VFEfig\_47.png inflating: data/train/Ventricular Fibrillation/VFEfig\_470.png inflating: data/train/Ventricular Fibrillation/VFEfig\_471.png inflating: data/train/Ventricular Fibrillation/VFEfig\_472.png inflating: data/train/Ventricular Fibrillation/VFEfig\_48.png inflating: data/train/Ventricular Fibrillation/VFEfig\_49.png inflating: data/train/Ventricular Fibrillation/VFEfig\_50.png inflating: data/train/Ventricular Fibrillation/VFEfig\_51.png inflating: data/train/Ventricular Fibrillation/VFEfig\_52.png inflating: data/train/Ventricular Fibrillation/VFEfig\_53.png inflating: data/train/Ventricular Fibrillation/VFEfig\_54.png inflating: data/train/Ventricular Fibrillation/VFEfig\_55.png inflating: data/train/Ventricular Fibrillation/VFEfig\_56.png inflating: data/train/Ventricular Fibrillation/VFEfig\_57.png inflating: data/train/Ventricular Fibrillation/VFEfig\_58.png inflating: data/train/Ventricular Fibrillation/VFEfig\_59.png inflating: data/train/Ventricular Fibrillation/VFEfig\_60.png inflating: data/train/Ventricular Fibrillation/VFEfig\_61.png inflating: data/train/Ventricular Fibrillation/VFEfig\_62.png inflating: data/train/Ventricular Fibrillation/VFEfig\_63.png inflating: data/train/Ventricular Fibrillation/VFEfig\_64.png inflating: data/train/Ventricular Fibrillation/VFEfig\_65.png inflating: data/train/Ventricular Fibrillation/VFEfig\_66.png inflating: data/train/Ventricular Fibrillation/VFEfig\_67.png inflating: data/train/Ventricular Fibrillation/VFEfig\_68.png inflating: data/train/Ventricular Fibrillation/VFEfig\_69.png inflating: data/train/Ventricular Fibrillation/VFEfig\_70.png inflating: data/train/Ventricular Fibrillation/VFEfig\_71.png inflating: data/train/Ventricular Fibrillation/VFEfig\_72.png inflating: data/train/Ventricular Fibrillation/VFEfig\_73.png inflating: data/train/Ventricular Fibrillation/VFEfig\_74.png inflating: data/train/Ventricular Fibrillation/VFEfig\_75.png inflating: data/train/Ventricular Fibrillation/VFEfig\_76.png



inflating: data/train/Ventricular Fibrillation/VFEfig\_77.png inflating: data/train/Ventricular Fibrillation/VFEfig\_78.png inflating: data/train/Ventricular Fibrillation/VFEfig\_79.png inflating: data/train/Ventricular Fibrillation/VFEfig\_80.png inflating: data/train/Ventricular Fibrillation/VFEfig\_81.png inflating: data/train/Ventricular Fibrillation/VFEfig\_82.png inflating: data/train/Ventricular Fibrillation/VFEfig\_83.png inflating: data/train/Ventricular Fibrillation/VFEfig\_84.png inflating: data/train/Ventricular Fibrillation/VFEfig\_85.png inflating: data/train/Ventricular Fibrillation/VFEfig\_86.png inflating: data/train/Ventricular Fibrillation/VFEfig\_87.png inflating: data/train/Ventricular Fibrillation/VFEfig\_88.png inflating: data/train/Ventricular Fibrillation/VFEfig\_89.png inflating: data/train/Ventricular Fibrillation/VFEfig\_90.png inflating: data/train/Ventricular Fibrillation/VFEfig\_91.png inflating: data/train/Ventricular Fibrillation/VFEfig\_92.png inflating: data/train/Ventricular Fibrillation/VFEfig\_93.png inflating: data/train/Ventricular Fibrillation/VFEfig\_94.png inflating: data/train/Ventricular Fibrillation/VFEfig\_95.png inflating: data/train/Ventricular Fibrillation/VFEfig\_96.png inflating: data/train/Ventricular Fibrillation/VFEfig\_97.png inflating: data/train/Ventricular Fibrillation/VFEfig\_98.png inflating: data/train/Ventricular Fibrillation/VFEfig\_99.png



# Image Augmentation / Preprocessing :

#Import req. Lib.

from tensorflow.keras.preprocessing.image import ImageDataGenerator

#Augmentation On Training Variable train\_datagen = ImageDataGenerator(rescale= 1./255,

zoom\_range=0.2, horizontal\_flip =True)

#Augmentation On Testing Variable

test\_datagen = ImageDataGenerator(rescale= 1./255)

#Augmentation On Training Variable

ftrain = train\_datagen.flow\_from\_directory('/content/data/train',

target\_size=(64,64), class\_mode='categorical', batch\_size=100)

Found 15341 images belonging to 6 classes.

#Augmentation On Testing Variable ftest = test\_datagen.flow\_from\_directory('/content/data/test', target\_size=(64,64),

class\_mode='categorical', batch\_size=100)

Found 6825 images belonging to 6 classes.

# Model Building :

## Adding Layers :

#Import req. Lib. from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense

# Build a CNN Block: model = Sequential() #intializing sequential model model.add(Convolution2D(32,(3,3),activation='relu', input\_shape=(64,64,3))) #convolution laye model.add(MaxPooling2D(pool\_size=(2, 2))) #Maxpooling layer model.add(Flatten()) #Flatten layer model.add(Dense(400,activation='relu')) #Hidden Layer 1 model.add(Dense(200,activation='relu')) #Hidden Layer 2 model.add(Dense(6,activation='softmax')) #Output Layer

## Compiling :

# Compiling The Model... model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

## Fit / Train The Model :

#Train Model:

model.fit\_generator(ftrain,

steps\_per\_epoch=len(ftrain), epochs=10, validation\_data=ftest, validation\_steps=len(ftest))

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:6: UserWarning: `Model.fit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epoch 1/10 |  | | | | | |
| 154/154 [==============================] | - 38s | 183ms/step | - loss: | 1.3586 | - accuracy: | 0. |
| Epoch 2/10 |  |  |  |  |  |  |
| 154/154 [==============================]  Epoch 3/10 | - 28s | 182ms/step | - loss: | 0.5405 | - accuracy: | 0. |
| 154/154 [==============================]  Epoch 4/10 | - 29s | 188ms/step | - loss: | 0.3288 | - accuracy: | 0. |
| 154/154 [==============================] | - 28s | 179ms/step | - loss: | 0.2590 | - accuracy: | 0. |
| Epoch 5/10 |  |  |  |  |  |  |
| 154/154 [==============================] | - 27s | 178ms/step | - loss: | 0.2221 | - accuracy: | 0. |
| Epoch 6/10 |  |  |  |  |  |  |
| 154/154 [==============================]  Epoch 7/10 | - 28s | 180ms/step | - loss: | 0.1891 | - accuracy: | 0. |
| 154/154 [==============================] | - 27s | 177ms/step | - loss: | 0.1738 | - accuracy: | 0. |
| Epoch 8/10 |  |  |  |  |  |  |
| 154/154 [==============================] | - 28s | 179ms/step | - loss: | 0.1544 | - accuracy: | 0. |

Epoch 9/10

154/154 [==============================] - 28s 180ms/step - loss: 0.1382 - accuracy: 0.

Epoch 10/10

154/154 [==============================] - 29s 186ms/step - loss: 0.1234 - accuracy: 0.

<keras.callbacks.History at 0x7f21e18a1c50>

## Saving The Model :

#Save Model model.save('CAUDL.h5')

# Testing The Model :

#Import req. Lib. from tensorflow.keras.preprocessing import image import numpy as np

#Testing No 1 :- img = image.load\_img('/content/data/test/Left Bundle Branch Block/fig\_5910.png',target\_size=( f = image.img\_to\_array(img) #Convertinng image to array f

= np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index

op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 15ms/step

'Left Bundle Branch Block'

#Testing No 2 :- img = image.load\_img('/content/data/test/Normal/fig\_2203.png',target\_size=(64,64)) #Reading i f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 15ms/step

'Normal'

#Testing No 3 :- img = image.load\_img('/content/data/test/Premature Atrial Contraction/fig\_1383.png',target\_si f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 17ms/step

'Premature Atrial Contraction'

#Testing No 4 :- img = image.load\_img('/content/data/test/Premature Ventricular Contractions/VEBfig\_1.png',tar f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 19ms/step

'Normal'

## ...Testing No 4 showing a wrng result ❗

#Testing No 5 :- img = image.load\_img('/content/data/test/Right Bundle Branch Block/fig\_100.png',target\_size=( f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions

pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

Block'

1/1 [==============================] - 0s 19ms/step

'Right Bundle Branch

#Testing No 6 :- img = image.load\_img('/content/data/test/Ventricular Fibrillation/VFEfig\_122.png',target\_size f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 15ms/step

'Ventricular Fibrillation' Model Tuning:

from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau early\_stop = EarlyStopping(monitor='val\_accuracy',

patience=5)

lr = ReduceLROnPlateau(monitor='val\_accuaracy',

factor=0.5, min\_lr=0.00001) callback = [early\_stop,lr]

#Train model model.fit\_generator(ftrain, steps\_per\_epoch=len(ftrain), epochs=100, callbacks=callback, validation\_data=ftest, validation\_steps=len(ftest))

Epoch 1/100

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:7: UserWarning: `Model.fit import sys

154/154 [==============================] - ETA: 0s - loss: 0.1177 - accuracy: 0.9613WAR

154/154 [==============================] - 28s 181ms/step - loss: 0.1177 - accuracy: 0.

Epoch 2/100

154/154 [==============================] - ETA: 0s - loss: 0.1034 - accuracy: 0.9677WAR

154/154 [==============================] - 29s 186ms/step - loss: 0.1034 - accuracy: 0.

Epoch 3/100

154/154 [==============================] - ETA: 0s - loss: 0.0962 - accuracy: 0.9709WAR

154/154 [==============================] - 27s 177ms/step - loss: 0.0962 - accuracy: 0.

Epoch 4/100

154/154 [==============================] - ETA: 0s - loss: 0.0890 - accuracy: 0.9728WAR

154/154 [==============================] - 27s 175ms/step - loss: 0.0890 - accuracy: 0.

Epoch 5/100

154/154 [==============================] - ETA: 0s - loss: 0.0812 - accuracy: 0.9743WAR

154/154 [==============================] - 27s 177ms/step - loss: 0.0812 - accuracy: 0.

Epoch 6/100

154/154 [==============================] - ETA: 0s - loss: 0.0808 - accuracy: 0.9738WAR

154/154 [==============================] - 27s 176ms/step - loss: 0.0808 - accuracy: 0.

Epoch 7/100

154/154 [==============================] - ETA: 0s - loss: 0.0711 - accuracy: 0.9776WAR

154/154 [==============================] - 27s 176ms/step - loss: 0.0711 - accuracy: 0.

Epoch 8/100

154/154 [==============================] - ETA: 0s - loss: 0.0631 - accuracy: 0.9789WAR

154/154 [==============================] - 27s 176ms/step - loss: 0.0631 - accuracy: 0.

Epoch 9/100

154/154 [==============================] - ETA: 0s - loss: 0.0647 - accuracy: 0.9802WAR

154/154 [==============================] - 28s 180ms/step - loss: 0.0647 - accuracy: 0.

Epoch 10/100

154/154 [==============================] - ETA: 0s - loss: 0.0530 - accuracy: 0.9828WAR

154/154 [==============================] - 28s 179ms/step - loss: 0.0530 - accuracy: 0.

Epoch 11/100

154/154 [==============================] - ETA: 0s - loss: 0.0465 - accuracy: 0.9849WAR

154/154 [==============================] - 27s 178ms/step - loss: 0.0465 - accuracy: 0.

<keras.callbacks.History at 0x7f21c0691510>

#Testing No 1 :- img = image.load\_img('/content/data/test/Left Bundle Branch Block/fig\_5898.png',target\_size=( f = image.img\_to\_array(img) #Convertinng image to array f

= np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 20ms/step

'Left Bundle Branch Block'

#Testing No 2 :- img = image.load\_img('/content/data/test/Normal/fig\_2113.png',target\_size=(64,64)) #Reading i f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding

dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 14ms/step

'Normal'

#Testing No 3 :- img = image.load\_img('/content/data/test/Premature Atrial Contraction/fig\_100.png',target\_siz f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 15ms/step

'Premature Atrial Contraction'

#Testing No 4 :-

img = image.load\_img('/content/data/test/Premature Ventricular Contractions/fig\_6090.png',tar f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index

op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 18ms/step

'Premature Ventricular Contractions'

## ...Testing No 4 now shows the correct result ✅

#Testing No 5 :- img = image.load\_img('/content/data/test/Right Bundle Branch Block/fig\_100.png',target\_size=( f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

Block'

1/1 [==============================] - 0s 42ms/step

'Right Bundle Branch

#Testing No 6 :- img = image.load\_img('/content/data/test/Ventricular Fibrillation/VFEfig\_198.png',target\_size f = image.img\_to\_array(img) #Convertinng image to array f = np.expand\_dims(f,axis=0) #Expanding dimensions pred = np.argmax(model.predict(f)) #predicting higher propability index op = ['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricul op[pred] #List indexing with output

1/1 [==============================] - 0s 60ms/step

'Ventricular Fibrillation'

## Saving The Model :

#Save Model model.save('CAUDL.h5') [Colab paid](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) [products](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) - [Cancel contracts here 0](https://colab.research.google.com/cancel-subscription)s

  

completed at 10:41 PM