Team ID : PNT2022TMID48713

Date : 05 November 2022

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

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

Sprint - 2

```
#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 [==============================] - 28s 182ms/step - loss: 0.5405 - accuracy: 0.
Epoch 3/10

154/154 [==============================] - 29s 188ms/step - loss: 0.3288 - accuracy: 0.
Epoch 4/10
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 [==============================] - 28s 180ms/step - loss: 0.1891 - accuracy: 0.
Epoch 7/10
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
```

```
1/1 [==============================] - 0s 19ms/step
'Right Bundle Branch
Block'
```

```
#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
```

```
1/1 [==============================] - 0s 42ms/step
'Right Bundle Branch
Block'
```

```
#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 products - Cancel contracts here 0s

completed at 10:41 PM