

CNN

November 20, 2022

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
[ ]: drive.mount("/content/drive", force_remount=True)
```

Mounted at /content/drive

```
[ ]: !unzip "/content/drive/MyDrive/Classification of Arrhythmia by Using Deep_
↳ Learning with 2-D ECG Spectral Image Representation.zip"
```

Streaming output truncated to the last 5000 lines.

```
inflating: data/train/Premature Ventricular Contractions/fig_1532.png
inflating: data/train/Premature Ventricular Contractions/fig_1533.png
inflating: data/train/Premature Ventricular Contractions/fig_1534.png
inflating: data/train/Premature Ventricular Contractions/fig_1535.png
inflating: data/train/Premature Ventricular Contractions/fig_1536.png
inflating: data/train/Premature Ventricular Contractions/fig_1537.png
inflating: data/train/Premature Ventricular Contractions/fig_1538.png
inflating: data/train/Premature Ventricular Contractions/fig_1539.png
inflating: data/train/Premature Ventricular Contractions/fig_154.png
inflating: data/train/Premature Ventricular Contractions/fig_1540.png
inflating: data/train/Premature Ventricular Contractions/fig_1541.png
inflating: data/train/Premature Ventricular Contractions/fig_1542.png
inflating: data/train/Premature Ventricular Contractions/fig_1543.png
inflating: data/train/Premature Ventricular Contractions/fig_1544.png
inflating: data/train/Premature Ventricular Contractions/fig_1545.png
inflating: data/train/Premature Ventricular Contractions/fig_1546.png
inflating: data/train/Premature Ventricular Contractions/fig_1547.png
inflating: data/train/Premature Ventricular Contractions/fig_1548.png
inflating: data/train/Premature Ventricular Contractions/fig_1549.png
inflating: data/train/Premature Ventricular Contractions/fig_155.png
inflating: data/train/Premature Ventricular Contractions/fig_1550.png
inflating: data/train/Premature Ventricular Contractions/fig_1551.png
inflating: data/train/Premature Ventricular Contractions/fig_1552.png
inflating: data/train/Premature Ventricular Contractions/fig_1553.png
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
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
```

Import the Libraries

```
[ ]: from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense
      from tensorflow.keras.layers import Convolution2D
      from tensorflow.keras.layers import MaxPooling2D
      from tensorflow.keras.layers import Flatten
```

Import The Image data Generator

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

Configure Image Data Generator Class

```
[ ]: train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.
    ↳ 2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

Apply Image Data Generator Functionality To Trainset and Testset

```
[ ]: x_train = train_datagen.flow_from_directory("/content/data/train", target_size =
    ↳ (64,64), batch_size = 32, class_mode = "categorical")
x_test = test_datagen.flow_from_directory("/content/data/test", 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.class_indices
```

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

Adding CNN Layers

```
[ ]: model = Sequential()
```

```
[ ]: model.add(Convolution2D(32,(3,3),input_shape = (64,64,3),activation = "relu"))
```

```
[ ]: model.add(MaxPooling2D(pool_size = (2,2)))
```

```
[ ]: model.add(Convolution2D(32,(3,3),activation='relu'))
```

```
[ ]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
[ ]: model.add(Flatten()) # ANN Input...
```

Adding Dense Layers

```
[ ]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =
    ↳ "relu"))
```

```
[ ]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =
    ↳ "relu"))
```

```
[ ]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =
    ↳ "relu"))
```

```
[ ]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =  
↪"relu"))
```

```
[ ]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation =  
↪"relu"))
```

Adding Output Layer

```
[ ]: model.add(Dense(units = 6,kernel_initializer = "random_uniform",activation =  
↪"softmax"))
```

```
[ ]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 128)	802944
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 128)	16512
dense_5 (Dense)	(None, 6)	774
=====		
Total params: 879,910		
Trainable params: 879,910		
Non-trainable params: 0		

```
[ ]: model.
      ↪ compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Train the model

```
[ ]: model.fit_generator(generator=x_train, steps_per_epoch = len(x_train), epochs=9,
      ↪ validation_data=x_test, validation_steps = len(x_test))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.

"""Entry point for launching an IPython kernel.

```
Epoch 1/9
480/480 [=====] - 98s 202ms/step - loss: 1.3996 -
accuracy: 0.4905 - val_loss: 1.7224 - val_accuracy: 0.4085
Epoch 2/9
480/480 [=====] - 95s 198ms/step - loss: 0.6987 -
accuracy: 0.7241 - val_loss: 0.6951 - val_accuracy: 0.8007
Epoch 3/9
480/480 [=====] - 95s 197ms/step - loss: 0.2940 -
accuracy: 0.9082 - val_loss: 0.5598 - val_accuracy: 0.8586
Epoch 4/9
480/480 [=====] - 95s 198ms/step - loss: 0.2009 -
accuracy: 0.9382 - val_loss: 0.5038 - val_accuracy: 0.8548
Epoch 5/9
480/480 [=====] - 95s 198ms/step - loss: 0.1519 -
accuracy: 0.9535 - val_loss: 0.5497 - val_accuracy: 0.8435
Epoch 6/9
480/480 [=====] - 95s 198ms/step - loss: 0.1341 -
accuracy: 0.9572 - val_loss: 0.5558 - val_accuracy: 0.8551
Epoch 7/9
480/480 [=====] - 95s 198ms/step - loss: 0.1195 -
accuracy: 0.9630 - val_loss: 0.8218 - val_accuracy: 0.8372
Epoch 8/9
480/480 [=====] - 95s 198ms/step - loss: 0.1029 -
accuracy: 0.9654 - val_loss: 0.4420 - val_accuracy: 0.8696
Epoch 9/9
480/480 [=====] - 96s 199ms/step - loss: 0.0887 -
accuracy: 0.9714 - val_loss: 0.4842 - val_accuracy: 0.8728
```

```
[ ]: <keras.callbacks.History at 0x7f03f8d5bb50>
```

Save the model

```
[ ]: #Saving Model.
      model.save('ECG.h5')
```

Testing the model

```
[ ]: from tensorflow.keras.models import load_model
     from tensorflow.keras.preprocessing import image
```

```
[ ]: model=load_model('ECG.h5')
```

```
[ ]: img=image.load_img("/content/data/test/Left Bundle Branch Block/fig_5906.
     ↳png",target_size=(64,64))
```

```
[ ]: x=image.img_to_array(img)
```

```
[ ]: img
```

```
[ ]:
```



```
[ ]: import numpy as np
```

```
[ ]: x=np.expand_dims(x,axis=0)
```

```
[ ]: pred = model.predict(x)
     y_pred=np.argmax(pred)
     y_pred
```

1/1 [=====] - 0s 140ms/step

```
[ ]: 0
```

```
[ ]: index=['left Bundle Branch block',
           'Normal',
           'Premature Atrial Contraction',
           'Premature Ventricular Contraction',
           'Right Bundle Branch Block',
           'Ventricular Fibrillation']
```

```
[ ]: result = str(index[y_pred])
     result
```

```
[ ]: 'left Bundle Branch block'
```