Project Development Phase

Delivery of Sprint 2

Date	11 November 2022
Team ID	PNT2022TMID16980
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Task 1:

Model Building:

Adding CNN Layers:

Code:

```
#ADDING CNN LAYERS

model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))#con
volution layer
model.add(MaxPooling2D(pool_size=(2,2)))#MaxPooling2D for downsampling
the input

model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())#flatten the dimension of the image
```

Adding Dense Layers:

Code:

```
#ADDING DENSE LAYERS

model.add(Dense(32))#deeply connected neural network layers.
model.add(Dense(6,activation='softmax'))
```

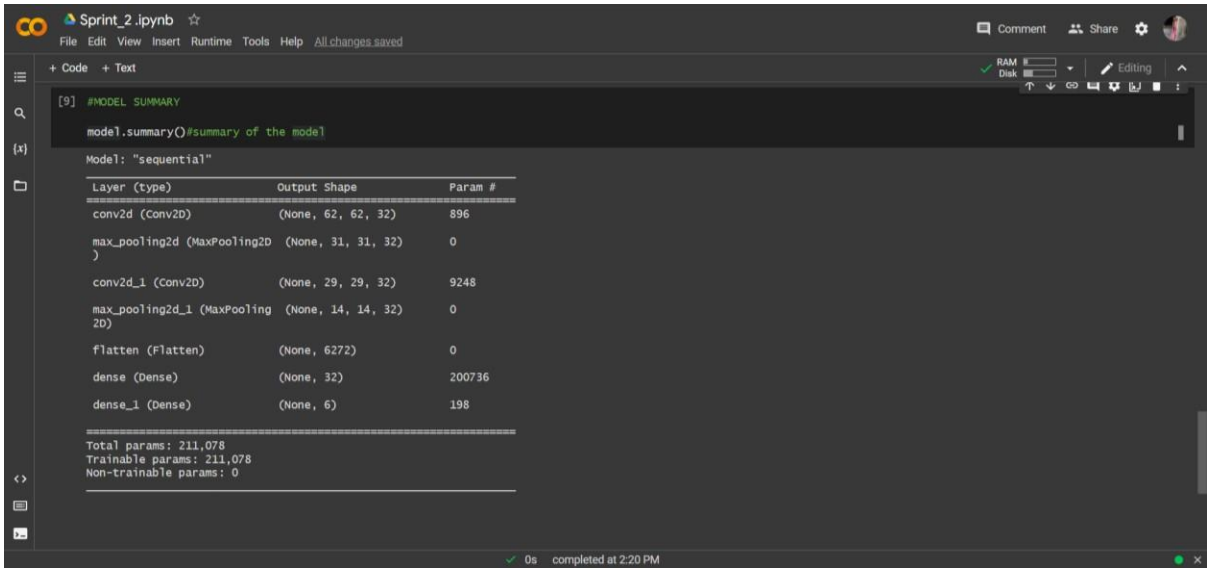
Model Summary:

Code:

```
#MODEL SUMMARY

model.summary()#summary of the model
```

Output:



Configure the Learning Process:

Code:

```
#CONFIGURE THE LEARNING PROCESS

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

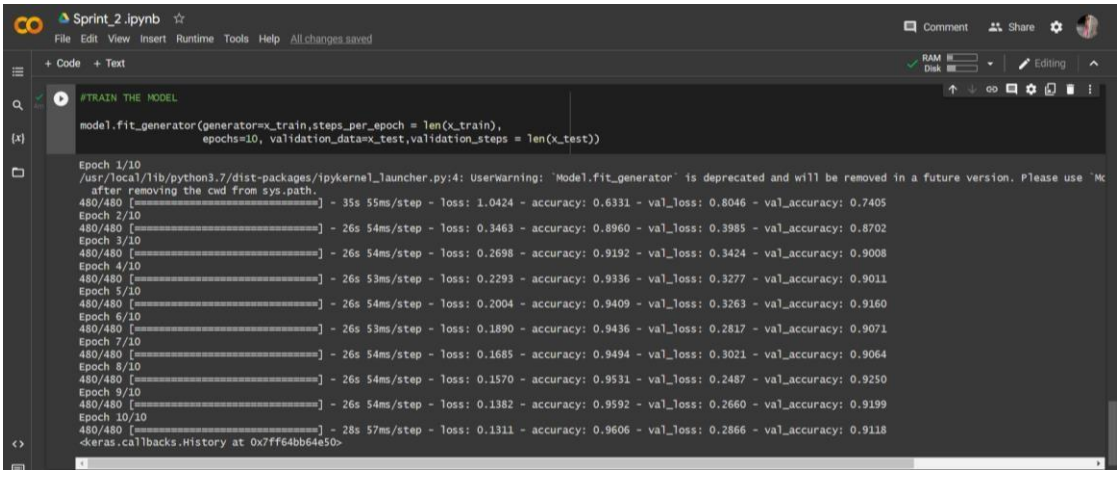
Train the Model:

Code:

```
#TRAIN THE MODEL

model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
      epochs=10, validation_data=x_test,validation_steps = len(x_test))
```

Output:



Save the Model:

Code:

```
#SAVE THE MODEL

model.save('ECG.h5')
```

Test the Model:

Code:

```
#TEST THE MODEL

#LOAD NECESSARY LIBRARIES
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

model = load_model("ECG.h5") #loading the model for testing

#IMAGE AS INPUT
img = image.load_img(r"/content/data/test/Premature Atrial Contraction/
fig_100.png",target_size=(64,64))#loading of the image
x = image.img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred = model.predict(x)#predicting the classes
pred

#PREDICTING THE OUTPUT
index=['Left Bundle Branch Block','Normal','Premature Atrial Contractio
n',
       'Premature Ventricular Contractions','Right Bundle Branch Block'
,'Ventricular Fabrillation']
result=str(index[np.argmax(pred)])
result
```

Output: