

Team id	PNT2022TMID00865
Project name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Accuracy Screenshot

The screenshot shows a Jupyter Notebook titled 'Arrhythmia.ipynb'. The code cell contains the following Python code:

```
[ ] model.fit(x_train,epochs=1,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

The output shows the training progress for 1 epoch:

```
120/120 [-----] - 182s 2s/step - loss: 0.1781 - accuracy: 0.9447 - val_loss: 0.5165 - val_accuracy: 0.8362
<keras.callbacks.History at 0x7efd403a8150>
```

The next code cell contains:

```
[ ] history=model.fit(x_train,epochs=1,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

The output shows the training progress for 1 epoch:

```
120/120 [-----] - 138s 1s/step - loss: 0.0979 - accuracy: 0.9687 - val_loss: 0.5897 - val_accuracy: 0.8620
```

The final code cell contains:

```
[ ] model.fit(x_train,epochs=5,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

The output shows the training progress for 5 epochs:

```
Epoch 1/5
120/120 [-----] - 132s 1s/step - loss: 0.0981 - accuracy: 0.9686 - val_loss: 0.5342 - val_accuracy: 0.8716
Epoch 2/5
120/120 [-----] - 127s 1s/step - loss: 0.0900 - accuracy: 0.9727 - val_loss: 0.6108 - val_accuracy: 0.8668
Epoch 3/5
120/120 [-----] - 126s 1s/step - loss: 0.0926 - accuracy: 0.9725 - val_loss: 0.6126 - val_accuracy: 0.8637
Epoch 4/5
120/120 [-----] - 127s 1s/step - loss: 0.0836 - accuracy: 0.9737 - val_loss: 0.5963 - val_accuracy: 0.8623
Epoch 5/5
120/120 [-----] - 129s 1s/step - loss: 0.0743 - accuracy: 0.9756 - val_loss: 0.4265 - val_accuracy: 0.8923
<keras.callbacks.History at 0x7f866830d350>
```

Summary screenshot

The screenshot shows a Jupyter Notebook titled 'Arrhythmia.ipynb'. The code cell contains the following Python code:

```
[ ] model.add(Dense(100,activation='relu'))
```

The next code cell contains:

```
[ ] model.add(Flatten())
```

The final code cell contains:

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

The output shows the model summary:

```
Model: "sequential"
Layer (type) Output Shape Param #
-----
conv2d (Conv2D) (None, 62, 62, 32) 896
max_pooling2d (MaxPooling2D) (None, 31, 31, 32) 0
flatten (Flatten) (None, 30752) 0
Total params: 896
Trainable params: 896
Non-trainable params: 0
```