```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  },
  "accelerator": "GPU",
  "gpuClass": "standard"
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "# **Import the Libraries:**"
   ],
   "metadata": {
    "id": "6a1Yf5f2Mpbb"
```

```
  }
 },
 {
  "cell_type": "code",
  "source": [
   "from tensorflow.keras.models import Sequential\n",
   "from tensorflow.keras.layers import Dense\n",
   "from tensorflow.keras.layers import Convolution2D\n",
   "from tensorflow.keras.layers import MaxPooling2D\n",
   "from tensorflow.keras.layers import Flatten"
  ],
  "metadata": {
   "id": "MHMDOqeQVlO0"
  },
  "execution_count": 6,
  "outputs": []
 },
 {
  "cell_type": "markdown",
  "source": [
   "# **Adding CNN Layers:**\n"
  ],
  "metadata": {
   "id": "pgyU84ZiNDcC"
  }
```

```
},
{
 "cell_type": "code",
 "source": [
  "model = Sequential()\n"
 ],
 "metadata": {
  "id": "iYSfVoZSWj95"
 },
 "execution_count": 11,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "model.add(Convolution2D(32,(3,3),input_shape = (64,64,3),activation = \"relu\"))"
 ],
 "metadata": {
  "id": "jJcs8KljWp4a"
 },
 "execution_count": 12,
 "outputs": []
},
{
 "cell_type": "code",
```

```
  "source": [
    "model.add(MaxPooling2D(pool_size=(2,2)))"
  ],
  "metadata": {
    "id": "sEOTDL-mWug2"
  },
  "execution_count": 13,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Convolution2D(32,(3,3),activation='relu'))"
  ],
  "metadata": {
    "id": "0yox9Q1SWyh6"
  },
  "execution_count": 14,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(MaxPooling2D(pool_size=(2,2)))"
  ],
```

```
  "metadata": {
    "id": "Z6g9fQNCW2ba"
  },
  "execution_count": 15,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Flatten()) # ANN Input..."
  ],
  "metadata": {
    "id": "u3GB7KClW6fD"
  },
  "execution_count": 16,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "# **Adding Dense Layers:**"
  ],
  "metadata": {
    "id": "mbCbTICYNOmo"
  }
```

```
},
{
 "cell_type": "code",
 "source": [
  "model.add(Dense(units = 128,kernel_initializer = \"random_uniform\",activation = \"relu\"))"
 ],
 "metadata": {
  "id": "F33fyAoTW9pF"
 },
 "execution_count": 17,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "model.add(Dense(units = 128,kernel_initializer = \"random_uniform\",activation = \"relu\"))"
 ],
 "metadata": {
  "id": "YHGGKI_fXAwQ"
 },
 "execution_count": 18,
 "outputs": []
},
{
 "cell_type": "code",
```

```
  "source": [
    "model.add(Dense(units = 128,kernel_initializer = \"random_uniform\",activation = \"relu\"))\n"
  ],
  "metadata": {
    "id": "xLPiFSRAXFdq"
  },
  "execution_count": 19,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Dense(units = 128,kernel_initializer = \"random_uniform\",activation = \"relu\"))"
  ],
  "metadata": {
    "id": "c8zTQieEXJTb"
  },
  "execution_count": 20,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Dense(units = 128,kernel_initializer = \"random_uniform\",activation = \"relu\"))"
  ],
```

```
  "metadata": {
    "id": "jkKmzrfgXMVL"
  },
  "execution_count": 21,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "# **Adding Output Layer:**"
  ],
  "metadata": {
    "id": "xXmU_i5bNZS4"
  }
},
{
  "cell_type": "code",
  "source": [
    "model.add(Dense(units = 6,kernel_initializer = \"random_uniform\",activation = \"softmax\"))"
  ],
  "metadata": {
    "id": "rYf7dQfJXT5U"
  },
  "execution_count": 22,
  "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "model.summary()"
 ],
 "metadata": {
  "id": "WxMoD5HOXary",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "a90c4e85-54b3-424a-96a4-b86f2ee3b060"
 },
 "execution_count": 23,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "Model: \"sequential\"\n",
    "_________________________________________________________________\n",
    " Layer (type)                Output Shape              Param #   \n",
    "=================================================================\n",
    " conv2d (Conv2D)             (None, 62, 62, 32)        896       \n",
    "                                                                 \n",
```

" max_pooling2d (MaxPooling2D (None, 31, 31, 32) 0 \n",

") \n",

" \n",

" conv2d_1 (Conv2D) (None, 29, 29, 32) 9248 \n",

" \n",

" max_pooling2d_1 (MaxPooling (None, 14, 14, 32) 0 \n",

" 2D) \n",

" \n",

" flatten (Flatten) (None, 6272) 0 \n",

" \n",

" dense (Dense) (None, 128) 802944 \n",

" \n",

" dense_1 (Dense) (None, 128) 16512 \n",

" \n",

" dense_2 (Dense) (None, 128) 16512 \n",

" \n",

" dense_3 (Dense) (None, 128) 16512 \n",

" \n",

" dense_4 (Dense) (None, 128) 16512 \n",

" \n",

" dense_5 (Dense) (None, 6) 774 \n",

" \n",

"===\n",

"Total params: 879,910\n",

"Trainable params: 879,910\n",

```
        "Non-trainable params: 0\n",
        "_________________________________________________________________\n"
      ]
     }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])"
   ],
   "metadata": {
    "id": "Hc5KpmodXfNs"
   },
   "execution_count": 24,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "# **Train the model:**\n"
   ],
   "metadata": {
    "id": "mU4gFNq3N4bv"
   }
```

```
},
{
 "cell_type": "code",
 "source": [
  "model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=9, validation_data=x_test,validation_steps = len(x_test))"
 ],
 "metadata": {
  "id": "q5vgj71KXnvH",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "f00a54f7-5367-4a63-cd00-0bdb6924e202"
 },
 "execution_count": 25,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stderr",
   "text": [
    "/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.\n",
    "  \"\"\"Entry point for launching an IPython kernel.\n"
   ]
  },
  {
```

```
"output_type": "stream",

"name": "stdout",

"text": [

 "Epoch 1/9\n",

 "480/480 [==============================] - 41s 66ms/step - loss: 1.3631 - accuracy: 0.5007 - val_loss: 1.6149 - val_accuracy: 0.4544\n",

 "Epoch 2/9\n",

 "480/480 [==============================] - 31s 65ms/step - loss: 0.7976 - accuracy: 0.6908 - val_loss: 0.9267 - val_accuracy: 0.6988\n",

 "Epoch 3/9\n",

 "480/480 [==============================] - 34s 71ms/step - loss: 0.3399 - accuracy: 0.8819 - val_loss: 0.6958 - val_accuracy: 0.7965\n",

 "Epoch 4/9\n",

 "480/480 [==============================] - 30s 63ms/step - loss: 0.2286 - accuracy: 0.9223 - val_loss: 0.5724 - val_accuracy: 0.8095\n",

 "Epoch 5/9\n",

 "480/480 [==============================] - 30s 63ms/step - loss: 0.1798 - accuracy: 0.9439 - val_loss: 0.4829 - val_accuracy: 0.8488\n",

 "Epoch 6/9\n",

 "480/480 [==============================] - 30s 63ms/step - loss: 0.1416 - accuracy: 0.9555 - val_loss: 0.5124 - val_accuracy: 0.8549\n",

 "Epoch 7/9\n",

 "480/480 [==============================] - 30s 62ms/step - loss: 0.1068 - accuracy: 0.9662 - val_loss: 0.5708 - val_accuracy: 0.8585\n",

 "Epoch 8/9\n",

 "480/480 [==============================] - 30s 63ms/step - loss: 0.0917 - accuracy: 0.9710 - val_loss: 0.4615 - val_accuracy: 0.8714\n",

 "Epoch 9/9\n",
```

```
"480/480 [==============================]  - 30s 62ms/step - loss: 0.0796 - accuracy: 0.9750 - val_loss: 0.7387 - val_accuracy: 0.8535\n"
      ]
     },
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "<keras.callbacks.History at 0x7f85e00f6410>"
       ]
      },
      "metadata": {},
      "execution_count": 25
     }
    ]
   },
   {
    "cell_type": "markdown",
    "source": [
     "# **Save the model:**"
    ],
    "metadata": {
     "id": "te53G3DfNs_P"
    }
   },
   {
```

```
  "cell_type": "code",
  "source": [
    "#Saving Model.\n",
    "model.save('ECG.h5')"
  ],
  "metadata": {
    "id": "Oeq8GP_5Xslp"
  },
  "execution_count": 26,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "# **Testing the model:**"
  ],
  "metadata": {
    "id": "kc-ZXIETU4E3"
  }
},
{
  "cell_type": "code",
  "source": [
    "from tensorflow.keras.models import load_model\n",
    "from tensorflow.keras.preprocessing import image"
```

```
 ],
 "metadata": {
  "id": "V7yaZSAGYo5A"
 },
 "execution_count": 27,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "model=load_model('ECG.h5')"
 ],
 "metadata": {
  "id": "Kux1_xG8Ys2b"
 },
 "execution_count": 28,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "img=image.load_img(\"/content/fig_44.png\",target_size=(64,64))"
 ],
 "metadata": {
  "id": "ZJ2Hk5yRYwMi"
```

```
  },
  "execution_count": 30,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "x=image.img_to_array(img)"
  ],
  "metadata": {
   "id": "DGprfkx8Y_0L"
  },
  "execution_count": 31,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "img"
  ],
  "metadata": {
   "id": "Uf2juK28Jnk9",
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 81
```

```
    },
    "outputId": "067bf602-6ab6-49f2-8f30-c42519daa2aa"
   },
   "execution_count": 32,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "<PIL.Image.Image image mode=RGB size=64x64 at 0x7F858A21FC10>"
      ],
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAEAAAABACAIAAAAlC+aJAAADEElEQVR4nO2av2v6QBjGE43g0E0tDp0c
OnSoi0vXunTqn9HBpVN3FeogUitowSJoK3QoODmJOHSTlqKI6OAkaIe2FBFsNYn3HcI3BFPzq30SCnmm944
3z+Xz3sUkZ2hCCPWX5bD6BH4qG8Bq2QBWywawWjaA1bIBNCgWi+HMzQDY3d3FmcMBCCEfHx84fzgAx3
HBYBDnDwdwuVxerxfnDwcYDAY+nw/nDwcYDoeBQADnDwfo9/tQfzjA3t7eer3G+cMB1us1z/M4fzjAdDql
aRrnDwfY3993Op04fzhAKBSCzgBtzr4QIQSEgZ2BxWIhFAh3HWMByuWyUHjcPAOXUKvVCoVCQvwnl1Cz2
RRj3HUMBPB4PDhzUUCAr68vnLkoIMBkMhHjz89P0ChAgKOjIzGWwvyugACHh4di/P7+DhoFCAB9ERMFB
GBZVozn8zloFCCAy+US47e3N9AougFms5mWNGn5KYoajUZ6B9IoTQCJRIL6/0Dmdru1PJlJy48VUVOj0RCC
6+vrm5sbQsjl5aXqUc/Pz9JmpVJRPcSYVGYglUodHx8L8fn5+cHBAUVROzs7qnV5enqSNjmOM1phNW0jy+fz
Qr2/VafTUS7M3d2dtFkoFPSWVqOYbWBnZ2cK2IPBIBgM3t7e+v3+09NTecLLy4u0CbwkDKNns1lCSLVanc/
n7XY7Ho+nUqlisSj0v76+SpPr9bqCFcuyj4+Pxk7DOICCksnkRk8ikVDIT6fTmUG2FiQG9nFxcVGz7Zf3tVqJay
Ck5MTg4MZ49ar+/v7b/tjsRjP84QQnuc5jsvlcg8PD7vcnTQKo1WryzvF4LG2WSiUhkK9ABZkE0Gw25Z0K6/7q
6kqjs0kA/X5fVz7Lsul0utfrqWaaBLBcLg0cNZvNWJYVm+VyWZ5j0taiXNFoNBqNqNBqNqqY1Go1ut8swDMWwkUh
EnrD1ToyW3+/XkhYOh8PhMEVR2/4lMelTA/nwDKOvdg7H96dqEoD8Ribds/iJ7O11baJpemMVbbxzGne26
lfot2R/L2S1bACrZQNYrX/JGj9yw9MuCQAAAABJRU5ErkJggg==\n"
     },
     "metadata": {},
     "execution_count": 32
    }
   ]
```

```
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ],
  "metadata": {
    "id": "1ABhw5HaZEUe"
  },
  "execution_count": 33,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "x=np.expand_dims(x,axis=0)"
  ],
  "metadata": {
    "id": "GbXNMF7hZPg4"
  },
  "execution_count": 34,
  "outputs": []
},
{
  "cell_type": "code",
```

```
"source": [
  "pred = model.predict(x)\n",
  "y_pred=np.argmax(pred)\n",
  "y_pred"
],
"metadata": {
  "id": "YBX_bDTaaBf_",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "20702708-8377-431f-fb5c-8c8e6560e7c4"
},
"execution_count": 35,
"outputs": [
 {
   "output_type": "stream",
   "name": "stdout",
   "text": [
     "1/1 [==============================]   - 0s 151ms/step\n"
   ]
 },
 {
   "output_type": "execute_result",
   "data": {
     "text/plain": [
```

```
        "4"
       ]
      },
      "metadata": {},
      "execution_count": 35
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "index=['left Bundle Branch block',\n",
    "       'Normal',\n",
    "       'Premature Atrial Contraction',\n",
    "       'Premature Ventricular Contraction',\n",
    "       'Right Bundle Branch Block',\n",
    "       'Ventricular Fibrillation']\n"
  ],
  "metadata": {
    "id": "FFMgfCEzZe5C"
  },
  "execution_count": 36,
  "outputs": []
},
{
```

```
"cell_type": "code",
"source": [
  "\n",
  "result = str(index[y_pred])\n",
  "result"
],
"metadata": {
  "id": "hElDLfK7ZfbV",
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 36
  },
  "outputId": "0ea0955d-9803-4c9b-fe97-558003733f03"
},
"execution_count": 37,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "'Right Bundle Branch Block'"
      ],
      "application/vnd.google.colaboratory.intrinsic+json": {
        "type": "string"
      }
```

```
        },
        "metadata": {},
        "execution_count": 37
       }
     ]
    }
  ]
}
```