Code Documentation & Installation

Client Name : Salaheddin Alakkari

Project Name : Deep Learning Model to Detect Heart Arrhythmia in ECG Data

Group 32

Pascal Raos, 3rd Year ICS

Maryann Foley, 3rd Year ICS

Anton Tiscoschi, 3rd Year ICS

Ailbhe Merriman, 3rd Year CSB

Kevin O'Donnell, 2nd Year CSB

Minjuan Luo, 2nd Year ICS

Zhongyuan Liu, 2nd Year ICS

1. Installation of Dependencies

A precursor to the steps below is that you have a version of Python installed that supports venv virtual environment creation. The project can be found alongside in the ZIP file or on our github repository: https://github.com/MaryannFoley/SWENG-Group-32-2022

1. In a terminal, go to the folder in which you want to keep your venv
2. Run “python3 -m venv EXVENV” (EXVENV is simply the name for the environment, you can name it whatever you wish)
3. Activate your virtual environment by running “source EXVENV /bin/activate” (or if that doesn’t work on your OS, cd into EXVENV/Scripts and run ". activate")
4. Now that the environment is activated, Install dependencies by typing “ pip3 install -r requirements.txt”
5. You now have the necessary python flask and tensorflow dependencies to run the web application and use the deep learning models

2. Running the Application

The next steps are running the application and passing in an input file to be tested by the models.

1. Go to the root directory for the project and run “python3 __init__.py” to start the webserver
2. To access the running application, either click on the ip link in the terminal or manually go there yourself by putting http://127.0.0.1:5000/ into your browser address bar
3. Now you can manually input testing files in the provided box that the AIs will make predictions of (files that you can use are provided with code zip file)
4. After predictions are made you will be brought to the results page, where it will indicate the results of every data input by row.
5. You can return to the home page with the previous screen button at the bottom of the results page.

3. Description of Files

The next steps are running the application and passing in an input file to be tested by the models.

In the root directory you can find the three models we implemented in their respective folders, with simplemodel being the CNN one. The __init__.py file executes the python flask frontend and accesses necessary functions for the models.

The aisrc folder contains the python files that create and train the models, and the main and computePrediction files are called by the frontend to make predictions on data. In aisrc, the data file contains the training and testing data we used to train the models