Assuming the user has the .hea files and .mat files in one folder and corresponding HEA and MAT files have the same name.

**Effective tree for using the Model**

full\_path/ECG/

├── Data/

│ ├── <Files.hea>

│ └── <Files.mat>

├── Dataset/

│ ├── ecg\_signal

│ ├── features

│ ├── labels

│ ├── train\_signal

│ ├── train\_feat

│ ├── train\_labels

│ ├── test\_signal

│ ├── test\_feat

│ ├── test\_labels

│ └── <...>

└── Model/

├── final\_model.pth

├── model\_epoch\_<n>.pth

└── <...>

**Creating Python Environment**

The python version used is Python 3.11.11. It is advised to make a virtual environment and then install the libraries needed for the code. Once the environment is created, activate it and load the requirement.txt file by running this in the environment terminal.

pip install -r requirements.txt

**Creating Training Dataset**

Open up create\_dataset.ipynb. Write dataset\_path='full\_path/ECG/Data' and memmap\_path='full\_path/ECG/Dataset/' (remember to put / when mentioned).

If a custom abnormality is needed to be classified then write the codes as strings in the list else keep it as []. If it was kept as [] then mention how many top n abnormalities to be classified by disease\_count=n (default is 7).

Now run the entire notebook. After the notebook has ran, go to the end as note the shape as (size,12,5000). Here size will be the effective dataset size. Remember this as it will be needed later. Call it 'dataset size'

**Splitting Dataset**

Open up divide\_dataset.ipynb. Write memmap\_path='full\_path/ECG/Dataset/' and put size= 'dataset size'. Set the disease\_count as chosen before and the dataset\_threshold to a desired ratio (preferable 0.8 or 0.9). The shape will be showed at the end of the notebook.

**Training the Model**

Open up train\_model.ipynb. Write model\_path='full\_path/ECG/Model/' and memmap\_path='full\_path/ECG/Dataset/' (remember to put / when mentioned). Set the disease\_size as chosen before. Put the dataset\_size = 'dataset size' or the splitted size. Put save\_index = n to save the model every n epochs. In the next cell, if the original dataset is being used (it has the name 'ecg\_signal') then put org = True and if the splitted dataset is being used (it has name 'train\_signal') then put org = False.

It is advised to keep the loss\_threshold and loss\_counter\_max to be left as default. Now run the notebook. At the end after training the backup model names will be shown.

**Testing the Model**

Open up test\_model.ipynb. Write model\_path='full\_path/ECG/Model/' and memmap\_path='full\_path/ECG/Dataset/' (remember to put / when mentioned). Set the disease\_size as chosen before. Put the dataset\_size = 'dataset size' or the splitted size. Put the model name that is going to be tested. Backup models will have the name of format 'model\_epoch\_n.pth'. Now run the notebook.

At the end the ROC AUC score will be shown. You can see the meaning of the score as written above that cell.

**Workarounds**

If two seperate datasets are used for training and testing then this is the advised method. First run the create\_dataset.ipynb on the training files. Once that is completed rename the "ecg\_signal" to "train\_signal", "features" to "train\_feat" and "labels" to "train\_labels". Now run create\_dataset.ipynb on the testing files. Once that is completed rename the "ecg\_signal" to "test\_signal", "features" to "test\_feat" and "labels" to "test\_labels". Now run train\_model.ipynb and test\_model.ipynb with org=False.

**Available**

A model trained on 50 epochs is already avaialble, so it can be directly used to test any dataset. Create the test\_dataset using create\_dataset.ipynb and load it into test\_dataset.ipynb with org=True and then run the notebook.