Objective 1: To find the standard 12 lead ECG rhythm chart

Different ECG machines manufacturer use different standards. But most common standard used:-

- Length of ECG: 10 seconds
- ECG speed: 25 mm/sec
- ECG amplitude scale: 10 mm/mV
- Length of each lead: 2.5 seconds
- Total leads: 12
- Rows * Columns: 3 * 4

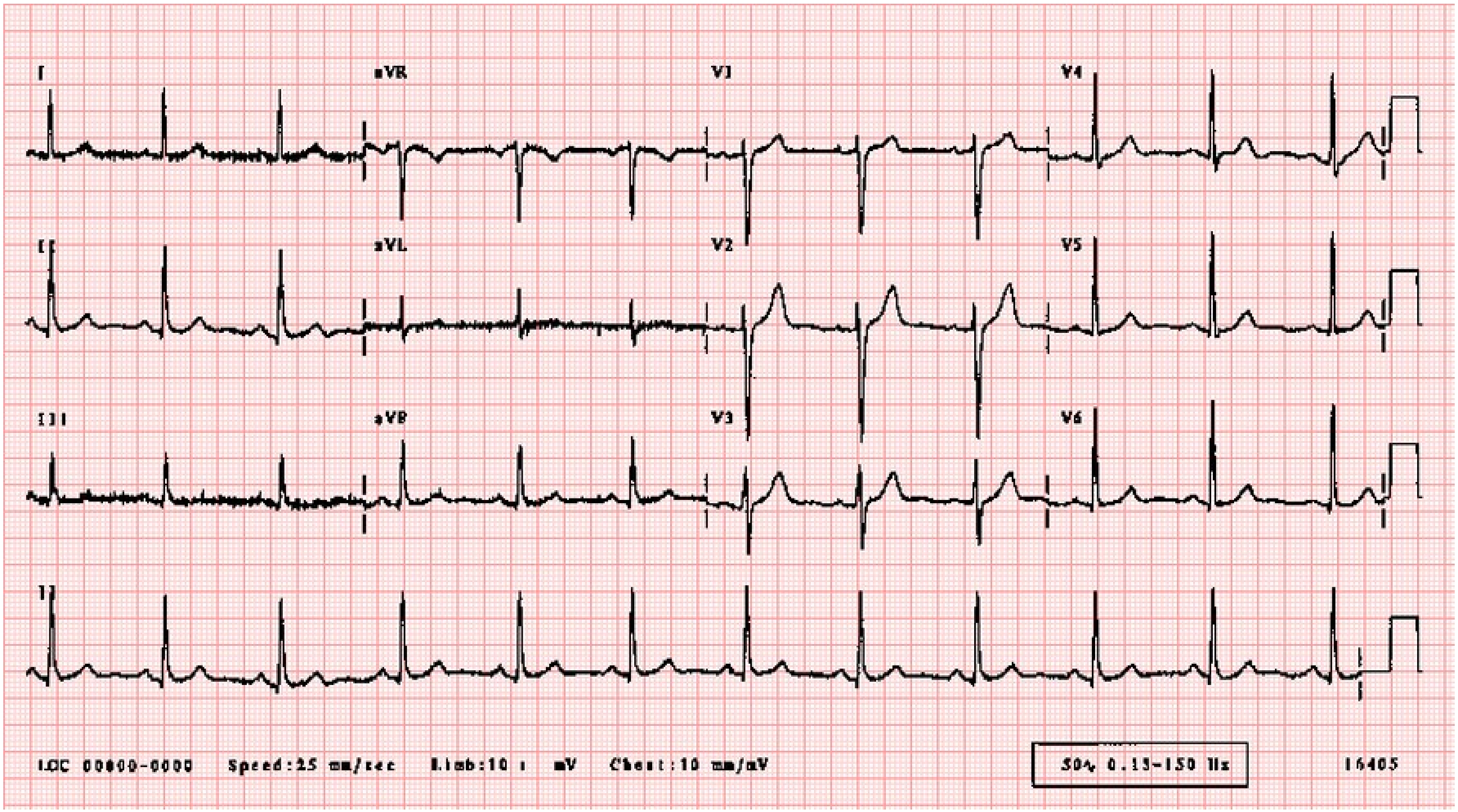
  Hence, there must be 50 large boxes in horizontal direction.
  Each large box is of size 5 mm * 5 mm.

  Each large box has 25 small boxes, each of size 1 mm * 1 mm.

  Doing the maths, 1 small box => 1 mm => 0.04 seconds
  & 1 large box => 5 mm => 0.20 seconds

- The chart also has an impulse in the beginning of height 1 mV or 10 mm used to check and calibrate the machine chart.
- The chart also has other useful informations.

Sample Original Chart

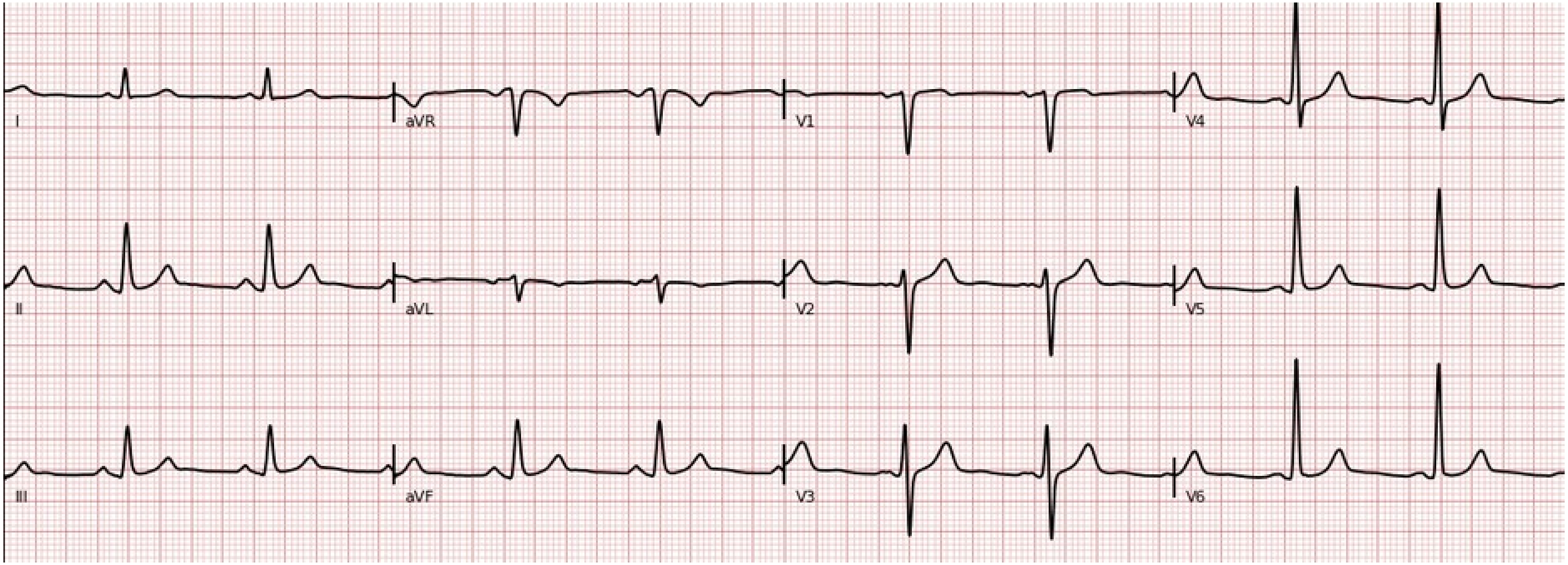


Objective 2: To create ECG charts from PTB dataset

- Imported the PTB databse[1].
- Done Pre-processing work:
  - Used 25 point moving average filter
  - Used IIR notch filter with notch at 50 Hz
- Searched & found ecg-plot python library online[2], but it is not giving the result in desired format.
- Forked the file and edited according to our found standards as per Objective 1.

[1] https://archive.physionet.org/physiobank/database/ptbdb/
[2] https://pypi.org/project/ecg-plot/

Sample ECG Charts Created



Saved the created charts in 2 folders according to their labels: HC & MI

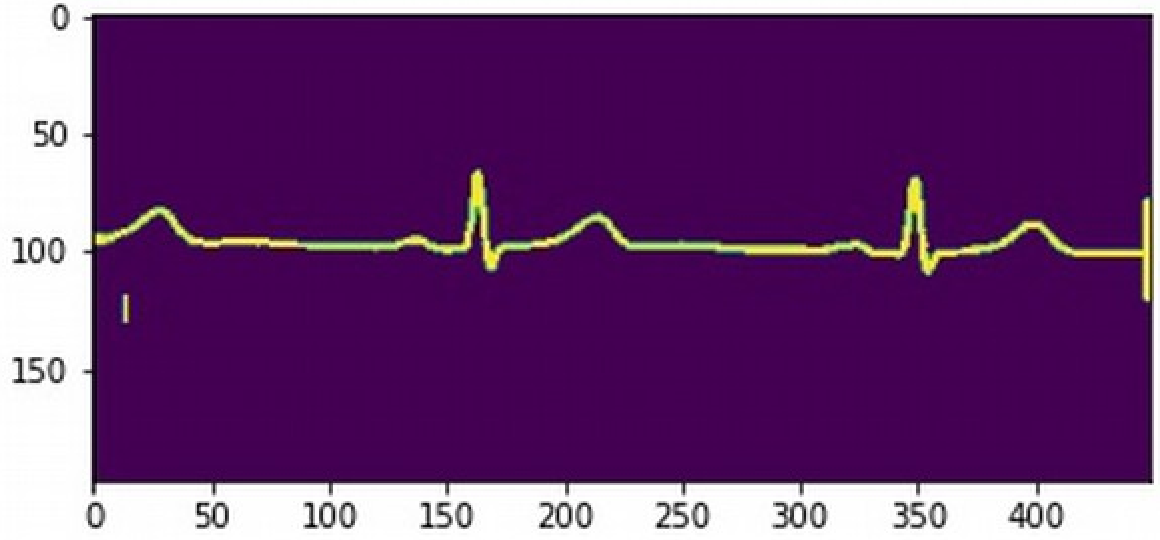
Digital Image Processing

- In this step, our objective was to read the images from the 2 folders and retrieve the multi lead ECG data.
- Challenges:-
  - Filter out the graph axis, lines and the text informations, and keep only the ECG lines.

    <u>Solution:-</u> Converted the RGB images to gray-scale. Using a threshold, removed the unwanted information, converted it to binary image. Cropped out the boundaries and text data.
  - Separate the 12 different ECG charts from one.

    <u>Solution:-</u> Using masking and cropping, divided the chart into 12 parts, stored them separately.

Sample Recovered ECG of lead V1



Note: This is a binary image.
It also has a text I and some boundary defect at the end.

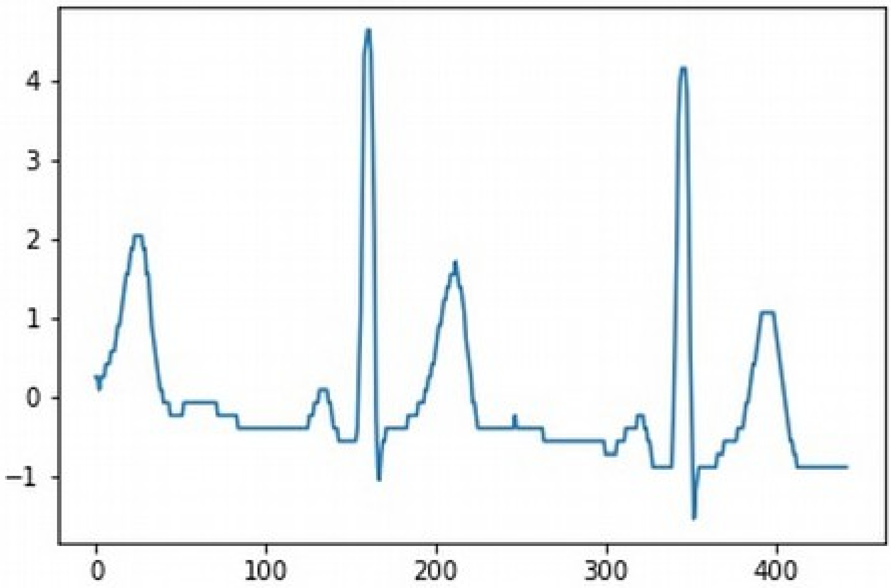
Convert Graph to Plot

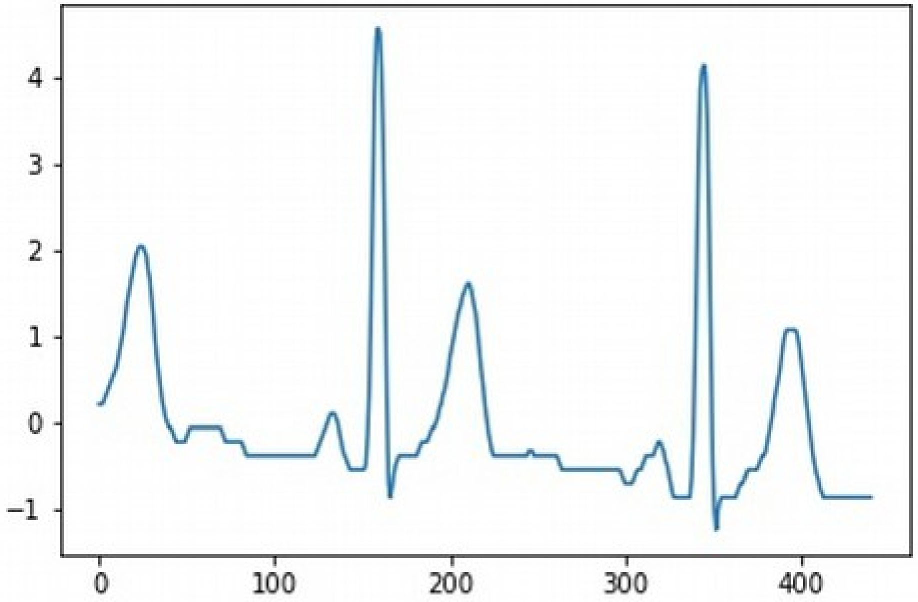
- We have to retrieve back the data from the graph.
- Challenges:-
  - Filter out the boundary defects and the lead information(text), and keep only the ECG line.

    Solution:- Cropped out the boundaries and text data and started vertical scanning to detect the ECG line. If line is absent, keep the previous sample value as the current sample.
  - The plot we get was not smooth.

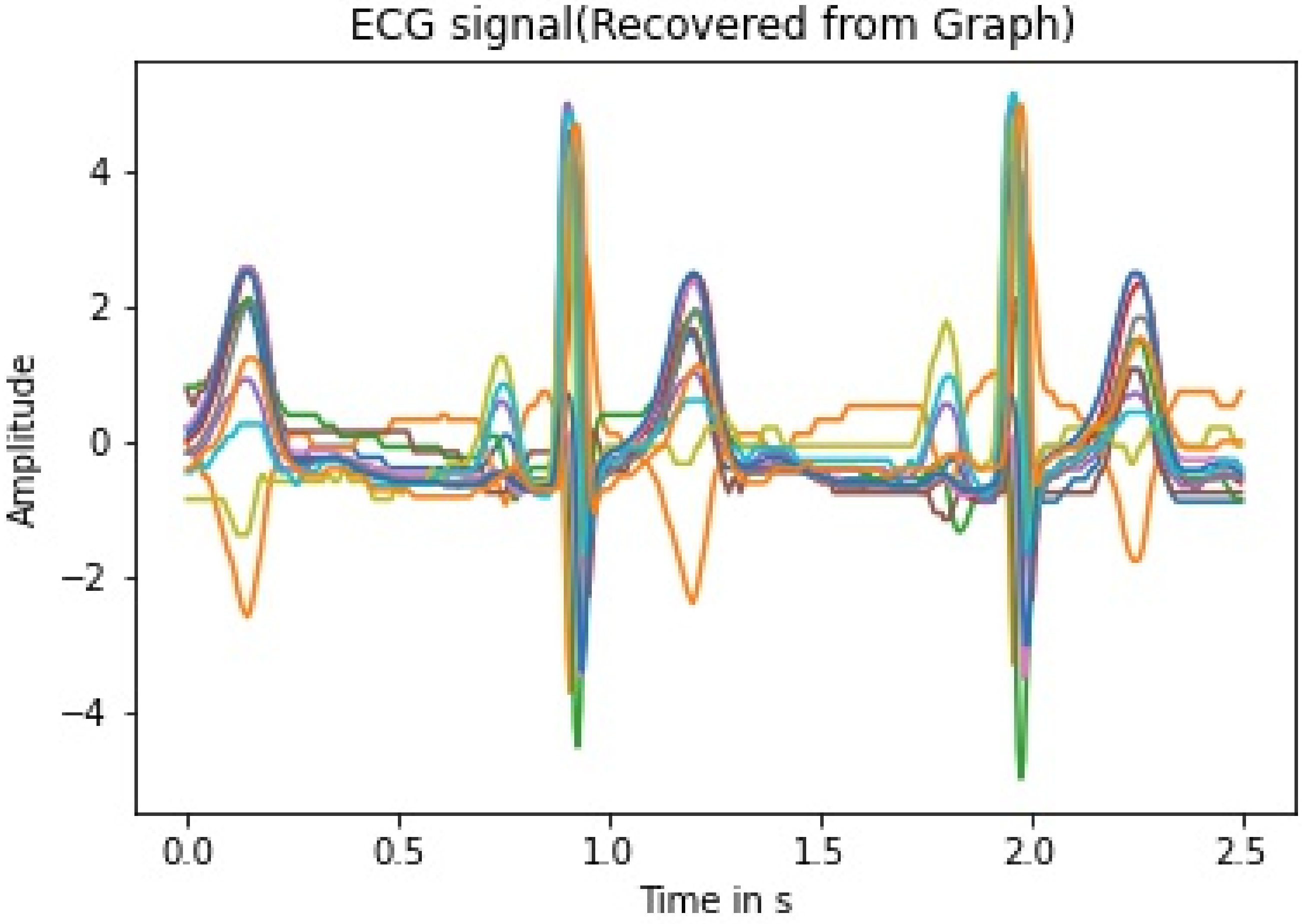
    Solution:- Used moving average filter.

Raw ECG Plot & Filtered ECG Plot





Filtered ECG Plot

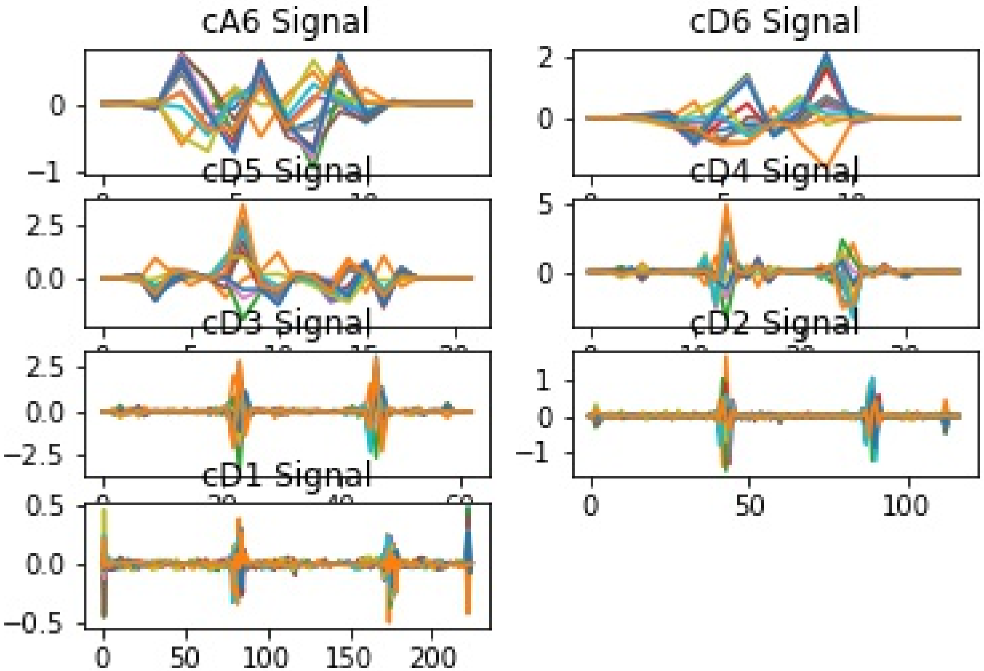


Objective 3: Apply SOTA paper on the retrieved data[3]

- Feed the data to the Program created before.
- Challenges:-
  - The paper used 4 beats to classify MI & HC ECG. But we have data of 2.5 seconds ECG, we have used *frame specific approach* instead of *beat specific approach*.
  - The recovered plot has very few sample points, so doing 6 level decomposition into sub-bands cause the problem of boundary effect.

[3] Multiscale Energy and Eigenspace Approach to Detection and Localization of Myocardial Infarction
https://ieeexplore.ieee.org/document/7047810

Output after Sub-band Decomposition



Results

- After performing 5 fold Cross validation:-

KNN mean accuracy	0.587755
Linear SVM mean accuracy	0.761224
RBF SVM mean accuracy	0.781632

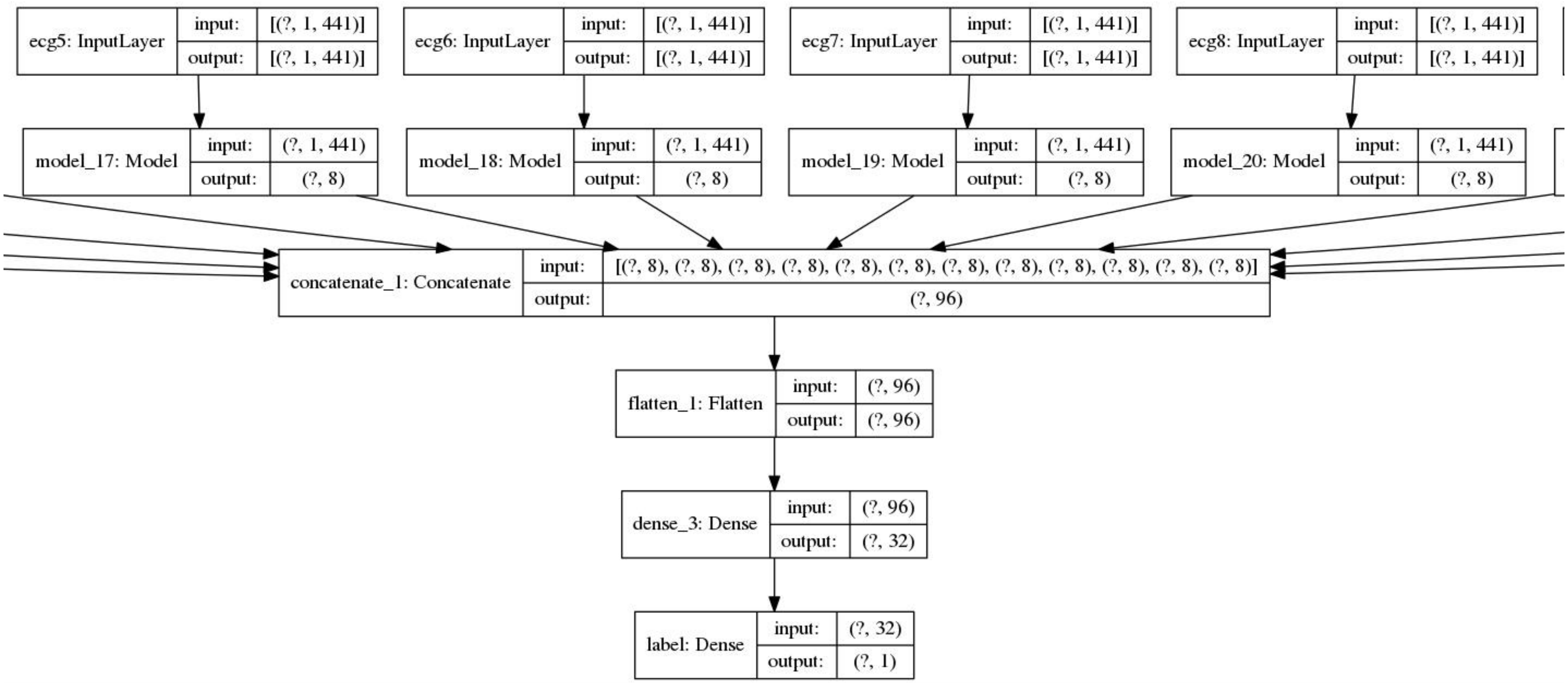
*The value can be improved further by hyper-parameter tuning.

Objective 4: Use of DNN for classification

- Two categories:-
  - Use 2D CNN directly on 12-ECG graphs.
  - Use 1D CNN on retrieved 12-ECG plots.

  <u>Challenges:</u> Tried using few architectures but facing the problem of "Constant Accuracy & Loss" during training.

DNN Architecture



*Work is under Progress

Future Work

- Use of Data Augmentation.
- Use of Skewing, zooming, shrinking images and adapt to these changes.
- Tuning Hyper parameters for better performance.
- Completing the classification using DNN work.