Project Overview:

Heart Sound Segmentation, as given in Kaggle Competition: https://www.kaggle.com/kinguistics/heartbeat-sounds, is my focus for the Capstone Project

The competition has two parts. I have attempted to address the 1st Challenge, which is :

Locate Heart Sound 1 and 2 (distinctive Lub..Dub…. Lub..Dub rythm) with in given audio file recorded through Digiscope of Phone app.

A set of labeled audio file is provided in the Kaggle competition.

It is a Multi class classification problem:

We have to identify 3 calsses:

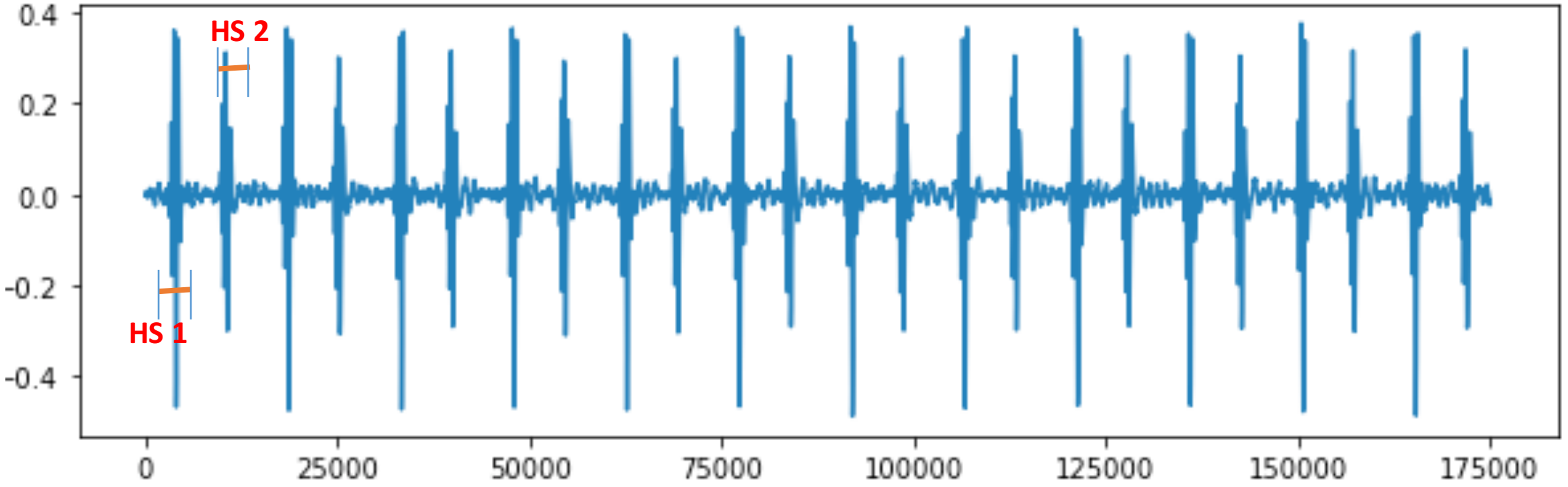
1. Heart Sound 1 ,
2. Heart Sound 2
3. All other noises picked up in the audio

Relevance in real life of this Project:

- According to the World Health Organization, cardiovascular diseases (CVDs) are the number one cause of death globally: more people die annually from CVDs than from any other cause. An estimated 17.1 million people died from CVDs in 2004, representing 29% of all global deaths
- Hence any method that helps in pre screening for CVDs is very important
- Recording heart sound with smart phones and digiscope, can be very convenient and inexpensive pre screening mechanism
- A challenge may be general public will not have proper training to place their phone/digiscope to pick up heart sounds
- In this process if machine learning model can assist to locate heart sounds apart from ambient noise and other noise generated
- inside body from lungs /stomach
- This can be stepping stone to build more sophisticated pre screening mechanism to detect heart conditions very early and get
- proper medical intervention
- https://www.kaggle.com/kinguistics/heartbeat-sounds http://www.peterjbentley.com/heartchallenge/

A Sample input audio signal

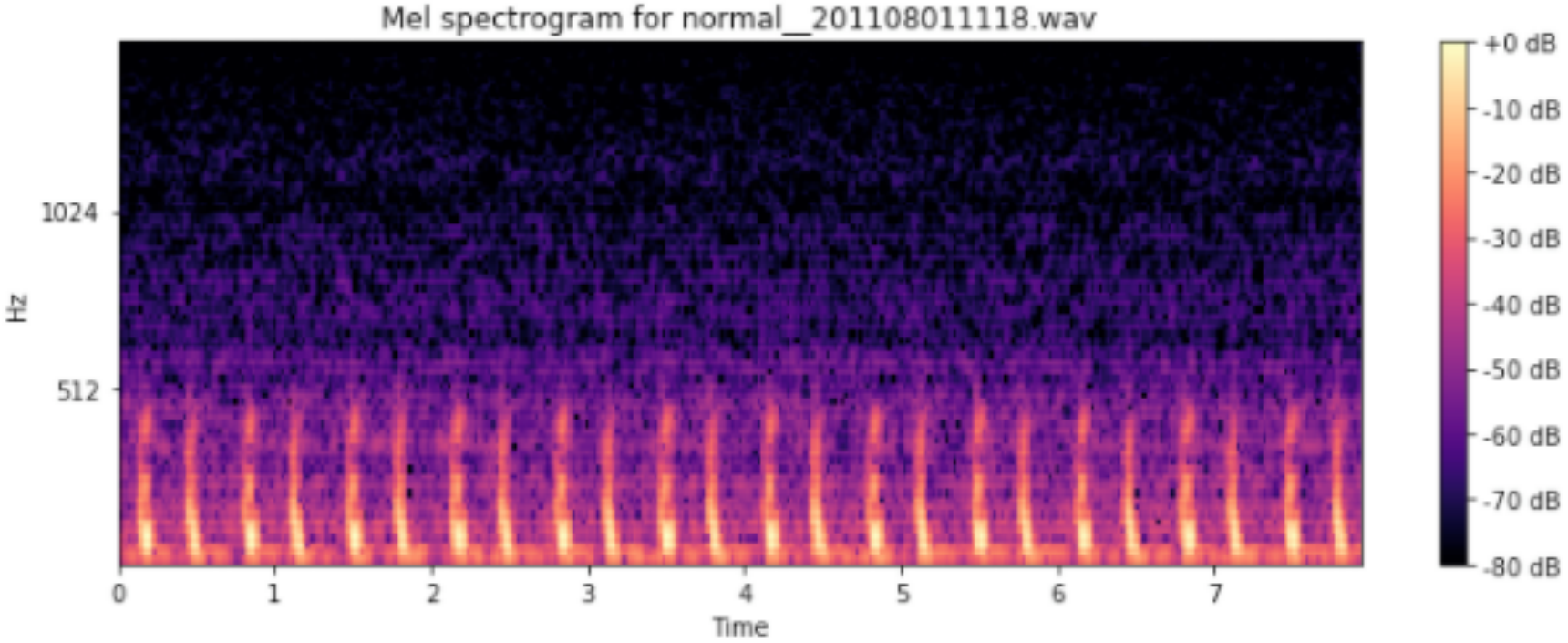
Normal Heart Sounds Audio signal:
Two distinct sounds are tagged as HS1 and HS2 on the graphs



Python Librosa library has been used for Audio signal processing:
Code reference: Classify_Heart_Sound.pynb:
x, fs = librosa.load('./normal__201108011118.wav')

A Sample input audio signal: Spectogram

Using Python Librosa library the sample audio file is converted into a Spectogram image



Code reference: Librosa:

```
S = librosa.feature.melspectrogram(y=x, sr=44100, n_mels=128,fmax=2000)
plt.figure(figsize=(10, 4))
librosa.display.specshow(librosa.power_to_db(S,ref=np.max),y_axis='mel', fmax=2000,x_axis='time')
```

The Strategy: Convolution Neural Net

Since Heart sound follows a regular pattern and looking at the spectogram, it is evident the problem is very similar to Image classification where patterns are identified through feature extractions.

The strategy is utilize image classification techniques following Conv.

I have found to start with, it will be a good idea to follow the MNIST example of hand written digit classification using Tensorflow and Convolution Neural Net and adapt the code towards audio signal classification (HS1, HS2 and noise)

Since it is a multiclass classification problem, we will use Confusion Matrix, F1 Scores to measure the performance of the model

Overall plan

Data Processing

- Convert Audio Signal to number arrays
- Extract Class specific ndarrays as per the labels provided

Train Test Split

- Make sure each class of data is well represented

Build Convolution NN

- Apply Conv. layer and pooling layer to Extract feature
- Fully connected layer to predict the class
- Train the model/ test the model

Metrics

- Confusion Matrix
- F1 Score

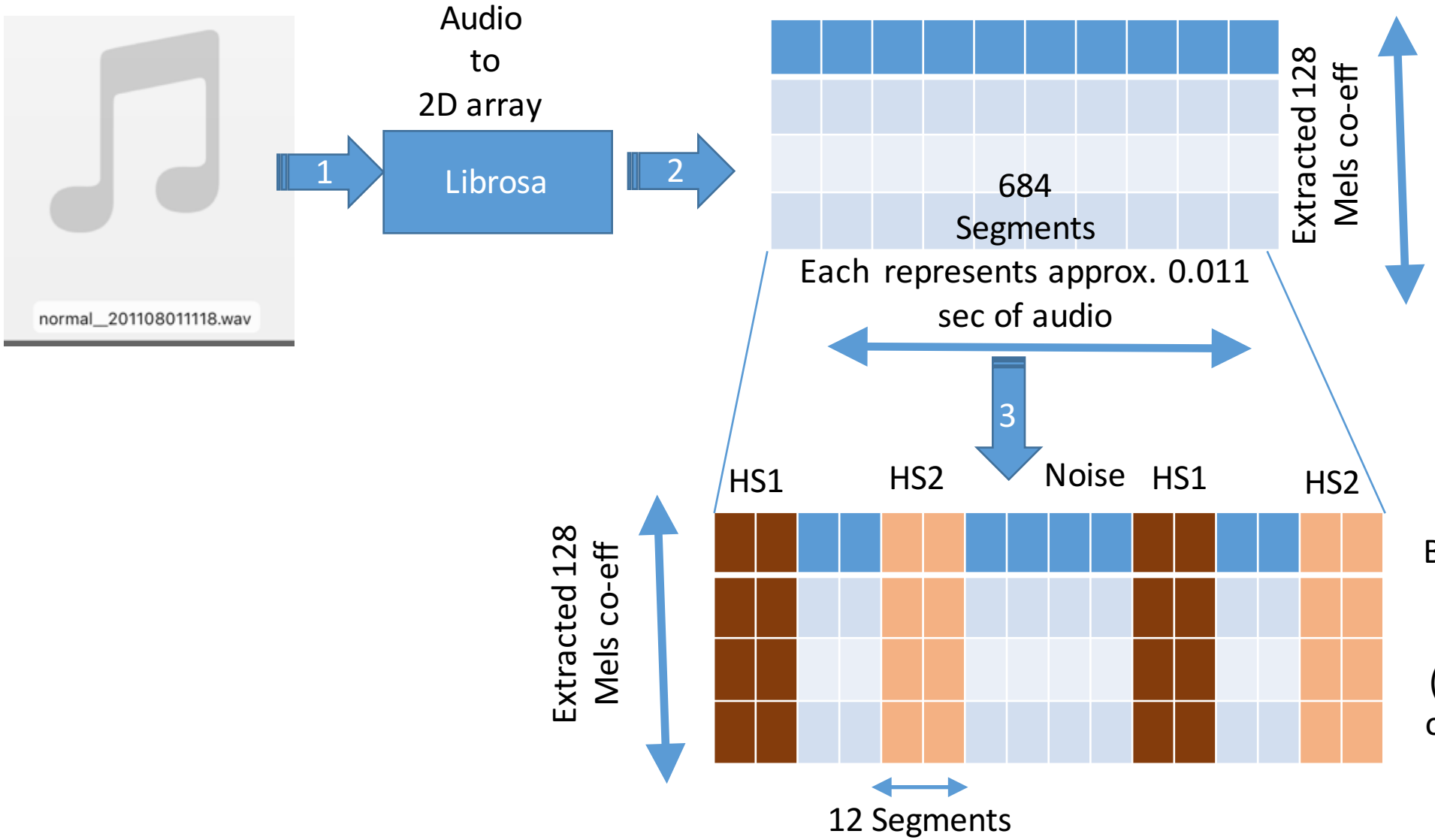
Exploring the Input Data Space

1. Audio signal info was converted into Array (mels_Co_eff): using Librosa python package, in pre processing step through main, settings and extract_hs_features.py 3 python files.
2. Each sample audio contains 128 Mel Co effs or features per 0.011 second of audio. The explanation of 0.011 sec sample segment is given below
3. From the audio property we know the audio is recorded at 44100 samples per second: aka **Sampling Rate**
4. In librosa one segment or slice of data under analysis by default contians 512 samples: aka default Hop_length
   - So each librosa segment contains (512/44100)=0.011 seconds of audio
   - Each segment (0.011 s duration of audio) is represented by 128 co effs
   - An audio file such as "normal__201108011118.wav" which is approx. 8 seconds long has (44100*8/512)= approx. 684 segements

So for an example the audio signal info recorded in 'normal__201108011118.wav' can be represented by an array of size 128*684

For each audio file we will extract 128 mels co eff for per 0.011 secs of audio sample. Based on the audio file duration we will get respective array 128*

Data Extraction

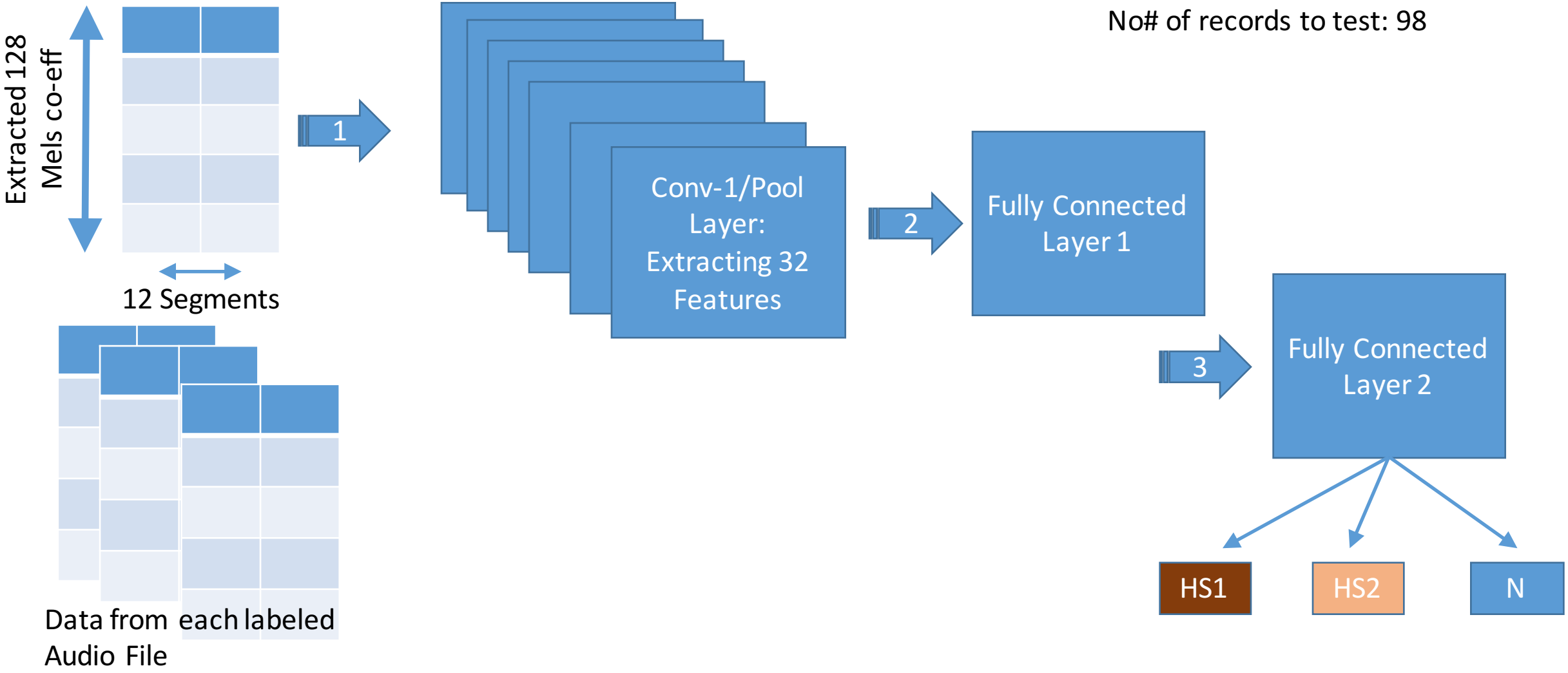


Based on the given labels a slice of 12 segments are extracted per label (HS1/HS2 & Noise based on their occurrence from 684 segments

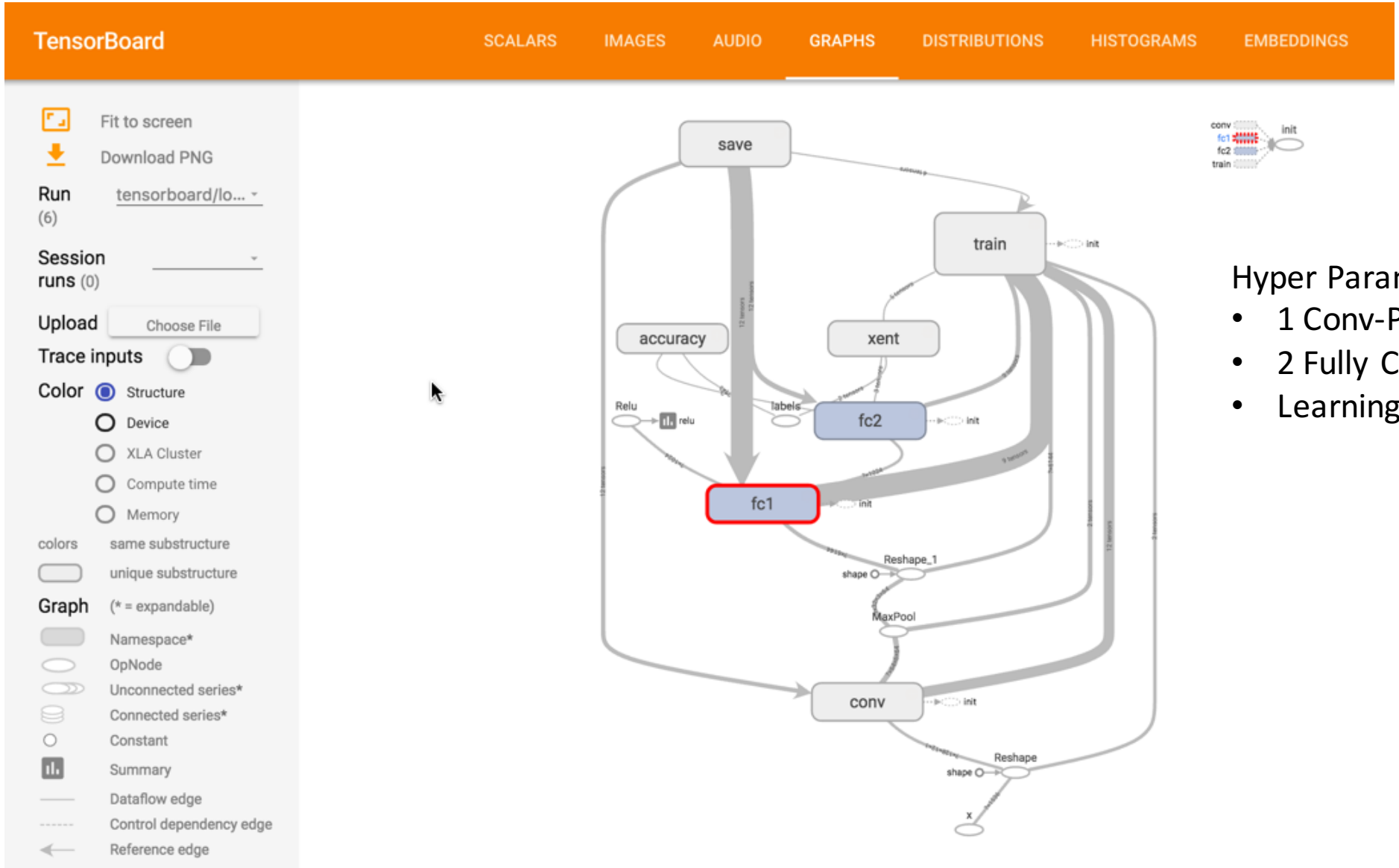
Model: Convolution Neural Net

Label:
HS1, HS2, Noise

No# of records to train: 488
No# of records to validate: 56
No# of records to test: 98



Conv. Net Architecture: Tensor Board

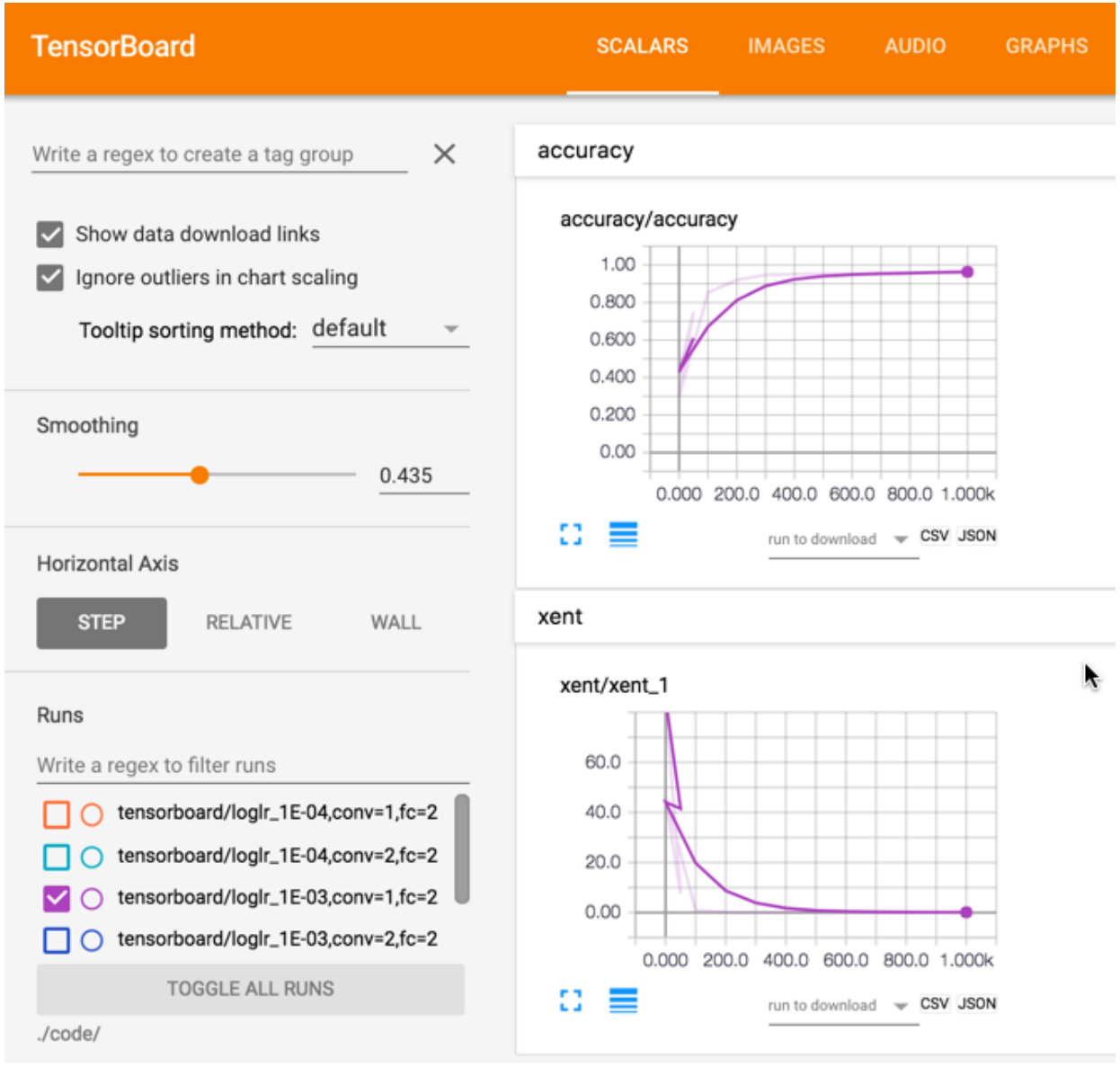


Hyper Parameters:

- 1 Conv-Pool Layer
- 2 Fully Connected Layers
- Learning Rate=1E-03

Training Step: Tensorboard

Accuracy Improved and Cross Entropy Dropped in Training Process

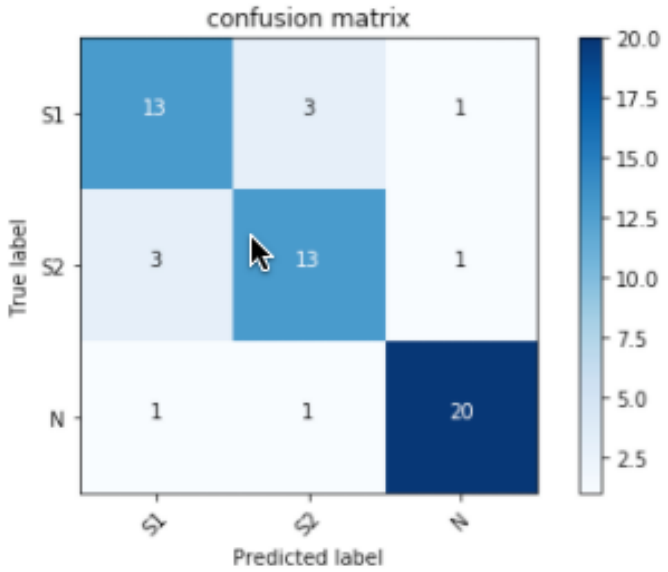


Metrics

- On Validation data :
  - Precision, Recall and F1 Score for each class

Class	Precision	Recall	F1-Score	Support
HS1	0.76	0.76	0.90	17
HS2	0.76	0.76	0.90	17
Noise	0.76	0.76	0.90	22

  - Confusion matrix:

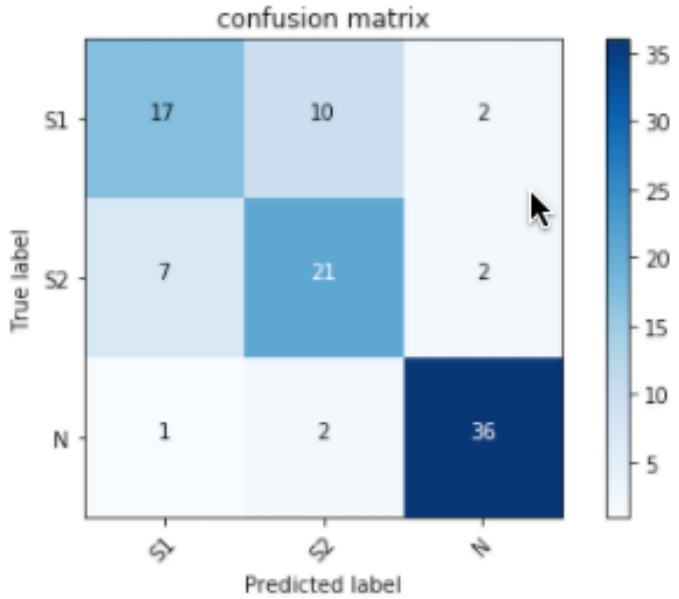


Metrics

- On Test data :
  - Precision, Recall and F1 Score for each class

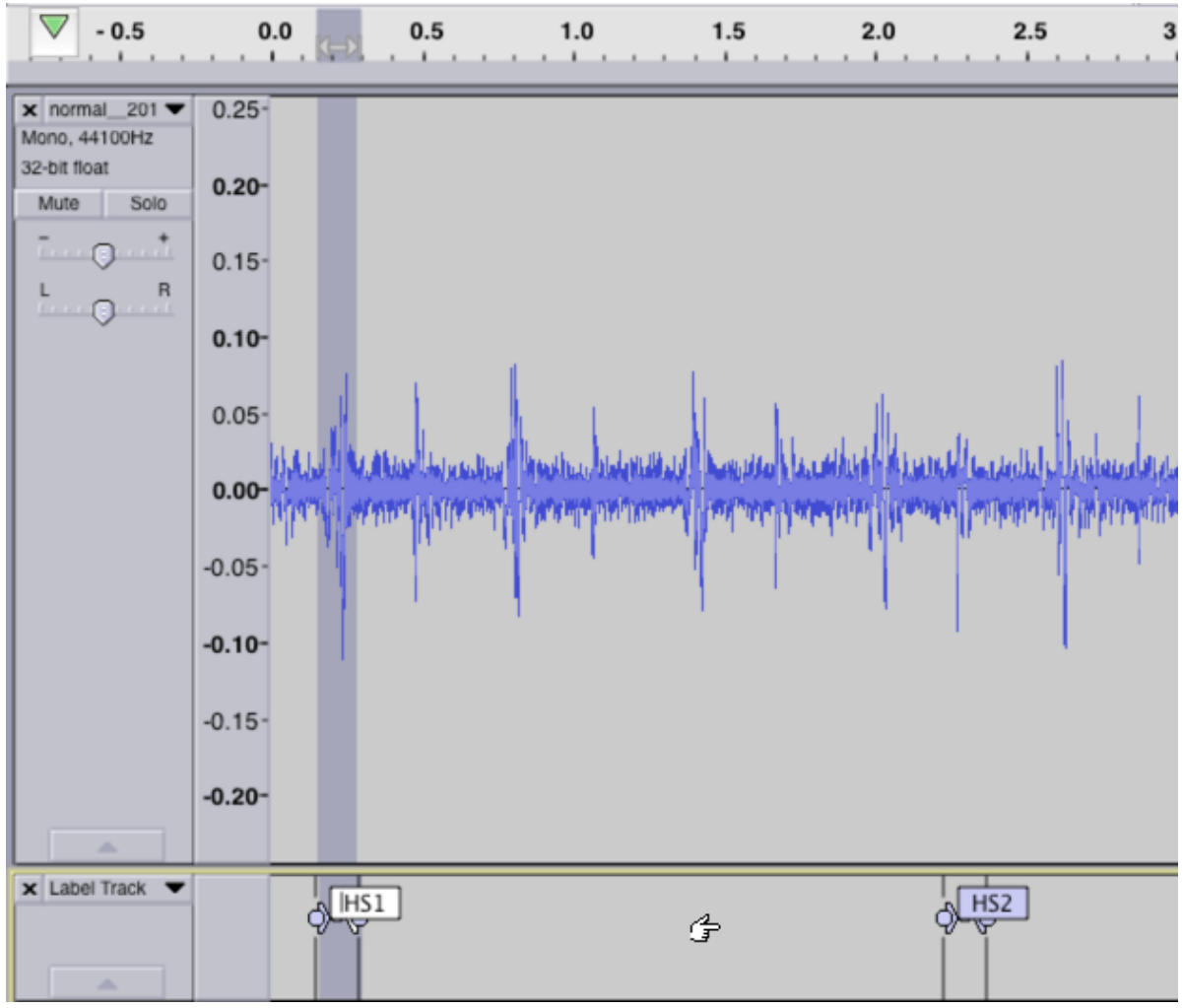
Class	Precision	Recall	F1-Score	Support
HS1	0.68	0.63	0.90	29
HS2	0.58	0.7	0.92	30
Noise	0.62	0.66	0.91	39

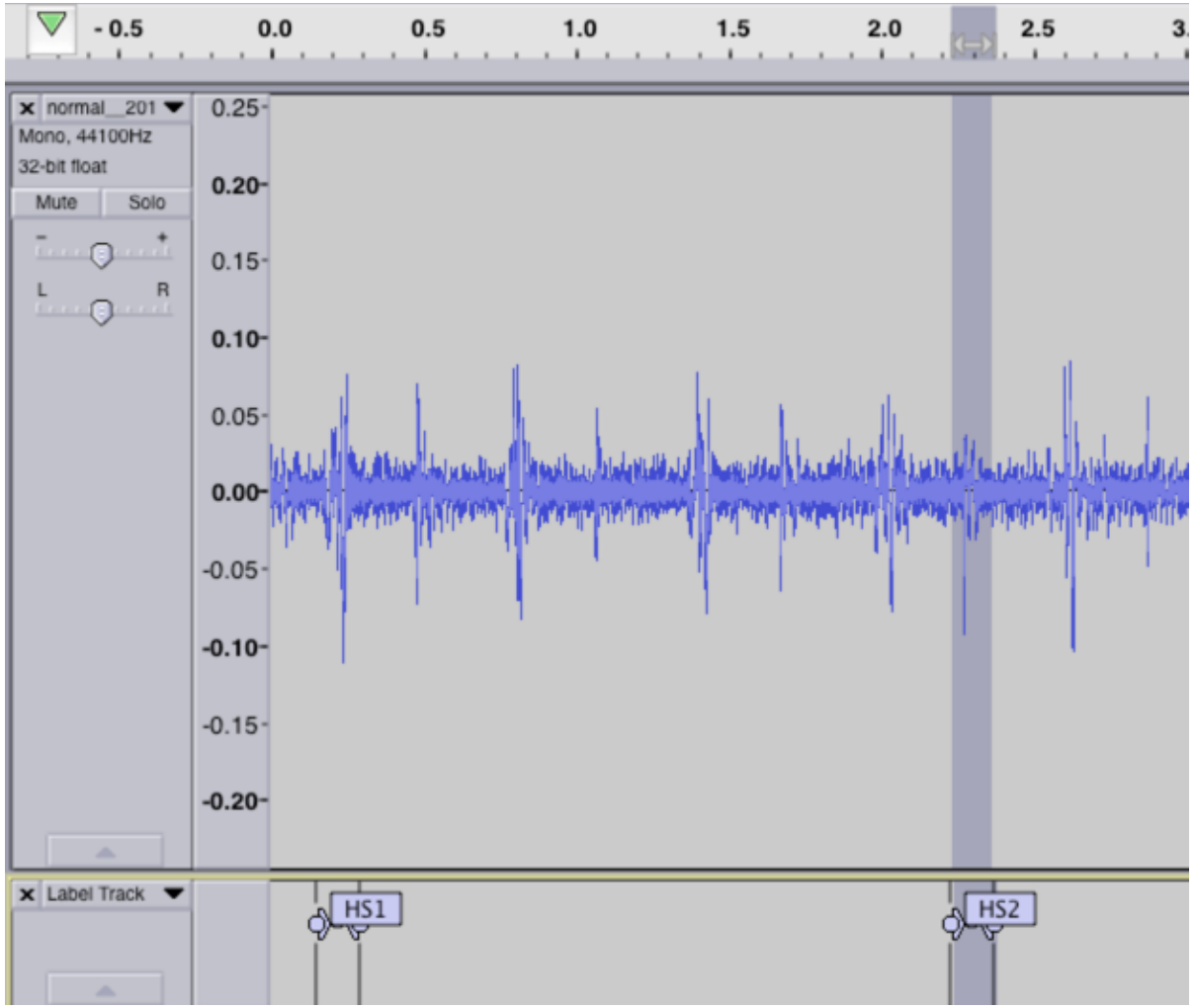
  - Confusion matrix:



Heart Sound Location- Identification

Below are two random examples of correctly classified Heart Sounds: file Ref: normal_201102081321.wav





Conclusion:

- For this Multi Class classification problem:
  - The model correctly identifies difference between Heart Sound and noise
  - But with in two Heart sounds, recall and precision are not so good what I expected
    - The F1 score on Validation data set is good
    - But it falls off for test data
    - It indicates more data is needed to Train the model with out overfitting to train data
  - With Limited number of labeled data available in Kaggle source (only 21 labeled audio file) , and
  - Given that Heart sounds are tricky, they change based on persons mood, stress, phyiscal condition, age, sex
    - I will consider the model will be a good first step
    - It proves Conv. Net (following the MNIST path) has potential in audio classification
    - The model can be improved with more deep layers & with more labeled data

Files in use

- Files used with description:

1. README.md: Contains the details of Program flow & file dependency
2. Settings.py, Extract_HS_Features.py and main.py extracts the mels co effs from wav files and pickles the result for data processing
3. labeled_mel_dict.p: contains the Mels features (2D array)extracted from audio signal
4. set_a_timing_csv.p: contains the sound location details , to be treated as labels
5. Classify_Heart_Sound.ipynb: contains the whole flow of the Model with step by step processing details

References

- MNIST example using Conv.net: https://www.tensorflow.org/get_started/mnist/pros
- convolutional neural network to distinguish the sound of foosball goals from other noises using TensorFlow:https://humblesoftwaredev.wordpress.com/2016/05/02/an-audio-dataset-and-ipython-notebook-for-training-a-convolutional-neural-network-to-distinguish-the-sound-of-foosball-goals-from-other-noises-using-tensorflow/
- Tf Dev summit: https://github.com/dandelionmane/tf-dev-summit-tensorboard-tutorial/blob/master/mnist.py