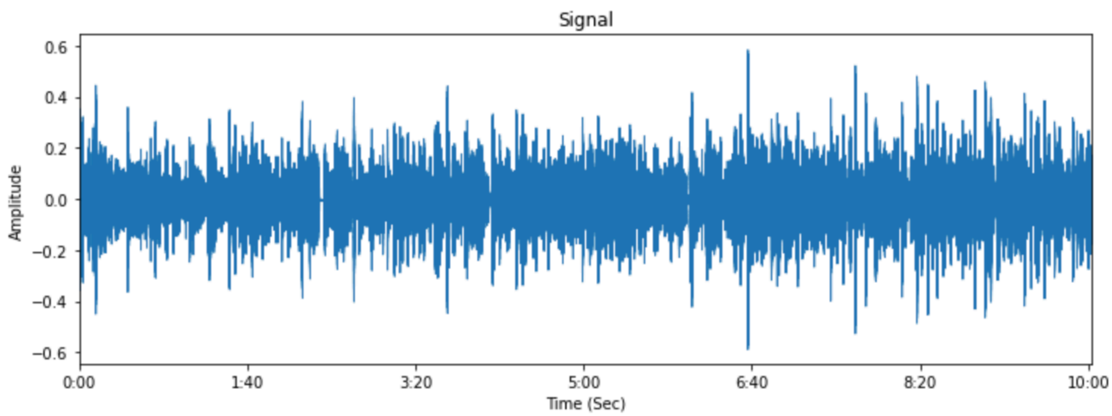
HOMEWORK 5 – EE599

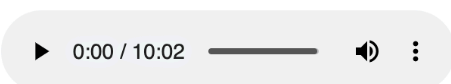
Name: Srujana Subramanya **USC ID: 4419643987**

1. Visualization of the given audio files

The figure below gives the spectrogram plot of the audio file. This is an example of the input given to us:

```
Audio Signal Vector:
[-4.4250610e-05 -5.1911356e-05  7.0553666e-05 ... -6.7294395e-04
 -5.5853359e-04  0.0000000e+00]
```





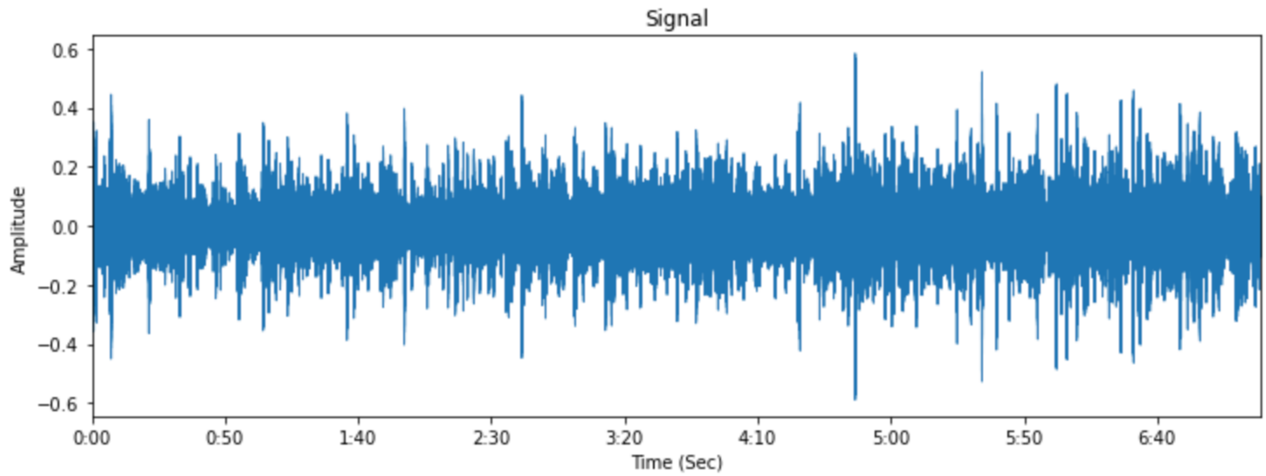
As a preprocessing step, the following snippet of code was written to remove silence from audio files.

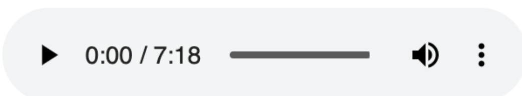
```python
#Preprocessing, removing silence off audio files and saving the silence removed files in: ./train/silence_removed/
from glob import glob
import os
import soundfile as sf

flag = False
if flag:
    path = glob('./train/*/*.wav')
    for file in path:
        head_tail = os.path.split(file)
        file_name = os.path.splitext(head_tail[1])[0]

        #load file
        x, sr = librosa.load(file)

        #remove silence
        y = librosa.effects.split(x,top_db=30)
        l = []
        for i in y:
            l.append(x[i[0]:i[1]] )
        x = np.concatenate(l,axis=0)

        sf.write('./train/silence_removed/' + file_name + '.wav', x, sr)
```

```
/Users/ssrujanaa/opt/anaconda3/lib/python3.8/site-packages/librosa/core/audio.py:162: UserWarning: PySoundFile faile
d. Trying audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
/Users/ssrujanaa/opt/anaconda3/lib/python3.8/site-packages/librosa/core/audio.py:162: UserWarning: PySoundFile faile
d. Trying audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
```

Once silence is removed, the spectrogram of the audio file is as shown below.

```
Audio Signal Vector:
[-4.4250610e-05 -5.1911356e-05  7.0553666e-05 ...  8.3047798e-04
  7.9184567e-04  1.0191973e-03]
```





2. <u>Network architecture.</u>

The overall flow of the network architecture is as shown below:

Given GRU and LSTM, I chose to work with GRU because it reduces overfitting as opposed to LSTM network.

3. <u>Training Approach.</u>

After preprocessing, the features extracted from each audio file (using librosa library, to obtained 64 mfcc features) are passed through the GRU network with 4 hidden units and number of layers 1. The GRU processes sequential data of the audio and further passes it to subsequently connected fully connected layer for classification over extracted features. The linear layers finally reduce the dimensions of the LSTM Network into 3 number of classes as output for our model.

I used weighted Cross Entropy Loss to handle the unbalanced dataset and Adam Optimizer to train my model.