Name: Nikita Ramesh Rao

Email: nikitar95@gmail.com

Phone: +918095689501

Location: Bengaluru

Problem Statement : Build a keras or tensorflow model to detect high peaks in audio or video files like gunshot scream etc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method and Task Details** | **Solution Screen Shot** | **Time Taken and result** | **Challenges faced and how it has been overcome** | **Notes and learned outcomes from this task** |
| 1. **Fast Fourier Transform**   2)**Building keras model**  3)**Peak detection using onset envelope** | Chart, histogram  Description automatically generated  A snapshot of a graph plotted using scipy fft to visualise the peak.      A snapshot of an audio file to visualize peaks.  A snapshot to see at which points the peaks are detected.  Text  Description automatically generated | **Time taken** : 2 days  **End Results:** Visualize the peaks through graph.  **Time taken** 4 days  **End results** : Training the model is always a better method to detect high peaks or other background noises(human voices)  There would be slight difference in the prediction if the shape of spectrogram is similar.  **Time taken** 1 day | **Challenges faced**:While researching this method to understand the mathematics behind it and learn the difference between fourier transform,fast fourier transform and short time fourier transform  Also which way is better to visualise the peaks  Using matplotlib or librosa  Matplotlib is always better to visualise.  **Challenges faced**:To create dataset in a proper format and the required CSV file to train the model.  To choose the right kind of model architecture.  To choose which method is better for feature extraction to train the model.  Found papers on sound event detection and sound classification  **Challenges faced** : How to get peak points  It could be easily achieved by using the onset envelope and librosa’s peak\_picks method | **Notes:** This method does not give much information on as to at what time there is a high peak and also it does not say about how the human ear would hear.  So in order to overcome this we can use STFT which converts FFT which uses time domain to frequency domain along with MFCC which is mel frequency which automatically calculates STFT and extracts the features.  **Learned outcomes:** STFT gives better information about an audio file than FFT.  **Notes**: With a good dataset and a basic MLP architecture we can get good accuracy for a set of audio files  **Learned outcomes:** How mfcc calculates frequencies using stft,creating mel spectrograms.  This is the best way to visualise peaks and also to get the output in a list form as to at which point the peaks are detected |