**CS461: Artificial Intelligence**

**Project Report Description**

**Group Members:**

Tashik 18k-0142

Sajjad Ali 18k-0355

**Abstract**

Recent research on machine learning focuses on audio source identification in complex environments. They rely on extracting features from audio signals and use machine learning techniques to model the sound classes. However, such techniques are often not optimized for a real-time implementation and in multi-source conditions. We propose a new real-time audio single-source classification method based on a dictionary of sound models (that can be extended to a multi-source setting). The sound spectrums are modeled with mixture models and form a dictionary. The classification is based on a comparison with all the elements of the dictionary by computing likelihoods and the best match is used as a result. We found that this technique outperforms classic methods within a temporal horizon of 0.5s per decision (achieved 6% of errors on a database composed of 50 classes). Future works will focus on the multi-sources classification and reduce the computational load.

**Background and Motivation**

Audio classification is the process of listening to and analyzing audio recordings. Also known as sound classification, this process is at the heart of a variety of modern AI technology including virtual assistants, automatic speech recognition, and text to speech applications. Sound/Audio Classification is one the most widely used applications in Audio Deep Learning. It involves learning to classify sounds and to predict the category of that sound. Audio Classification is an interesting problem in Machine Learning and Deep Learning, which mainly target for recognizing and relating sounds from audio. This type of problem can be applied to many scenarios for example classifying music clips to identify the genre of the music, or classifying short utterances by a set of speakers to identify the speaker based on the voice.

Observing the recent advancements in the field of image classification where convolutional neural networks are used to classify images with high accuracy and at scale, it begs the question of the applicability of these techniques in other domains, such as sound classification.

Motivated by Google research which recently released a sound vocabulary and dataset aiming to provide a common, realistic-scale evaluation platform for audio event classification such as human sounds, music genres, environmental sounds (see <https://research.google.com/audioset/> ).

**Problem Statement**

When given an audio sample in a computer readable format (such as a .wav file) of a few seconds duration, we want to be able to determine if it contains one of the target urban sounds with a corresponding Classification Accuracy score. In output, our model will be able to predict different sounds belonging to different classes given in the dataset.

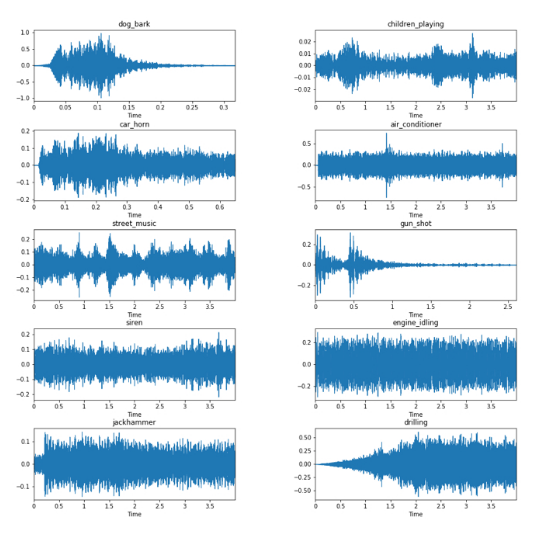
**Research Paper**

We used the following paper as a reference,  
<https://ieeexplore.ieee.org/document/7177954>

**Dataset**

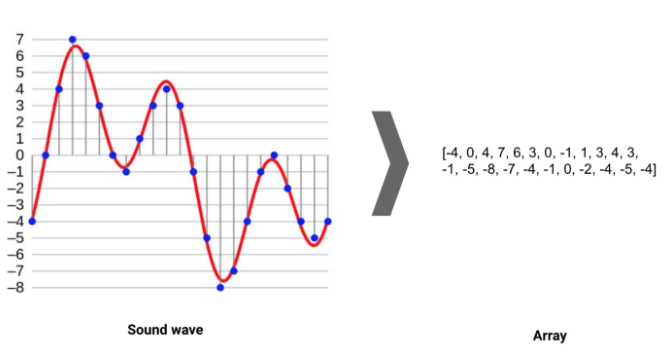
For this we have use a dataset called Urbansoun8K. The dataset contains 8732 sound excerpts (<= 4s) of urban sounds from 10 classes, which are as follow:

1. Air Conditioner
2. Car Horn
3. Children Playing
4. Dog Bark
5. Drilling
6. Engine Idling
7. Gun Shot
8. Jackhammer
9. Siren
10. Street Music



**Audio file overview**

These sound excerpts are digital audio files in .wav format. Sound waves are digitized by sampling them at discrete intervals known as the sampling rate (typically 44.1kHz for CD quality audio meaning samples are taken 44,100 times per second). Each sample is the amplitude of the wave at a particular time interval, where the bit depth determines how detailed the sample will be also known as the dynamic range of the signal (typically 16bit which means a sample can range from 65,536 amplitude values).



**Audio Features**

There are, in general, two types of audio features: the **physical** features and the **perceptual** features. Physical features refer to mathematical measurements computed directly from the sound wave, such as the **energy function**, the **spectrum**, the **cepstral** coefficients, the **fundamental frequency**, and so on.

**Sound**

Sound is a vibration that creates a wave that can travel through air, through liquids or through solids. These vibrations can be changed in two ways either by changing the **frequencies** (how quickly the sound vibration happens known as pitch) or by changing the **amplitude** (how much energy is applied to vibrations known as volume).

**Extract Features**

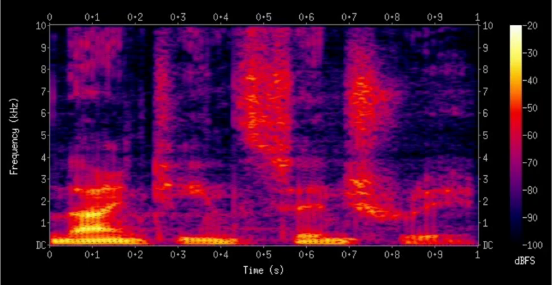
To Extract Features, we have created a visual representation of each of the audio samples which will allow us to identify features for classification, using the same techniques used to classify images with high accuracy.

Spectrograms are a useful technique for visualizing the spectrum of frequencies of a sound and how they vary during a very short period of time. We will be using a similar technique known as Mel-Frequency Cepstral Coefficients (MFCC).



**Spectrogram**

A spectrogram is a visual representation of the spectrum of **frequencies** of a signal which allows us to see sounds, it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data are represented in a 3D plot, they may be called waterfalls.

****

* Higher frequency or higher pitch sounds are at the top.
* Lower frequency or lower pitch sounds are at the bottom.
* Higher amplitude or higher volume frequencies are in brighter color.
* Lower amplitude or lower volume frequencies are in dark color.

**Mel Frequency Cepstral Coefficients (MFCC)**

**MFCC Calculation**

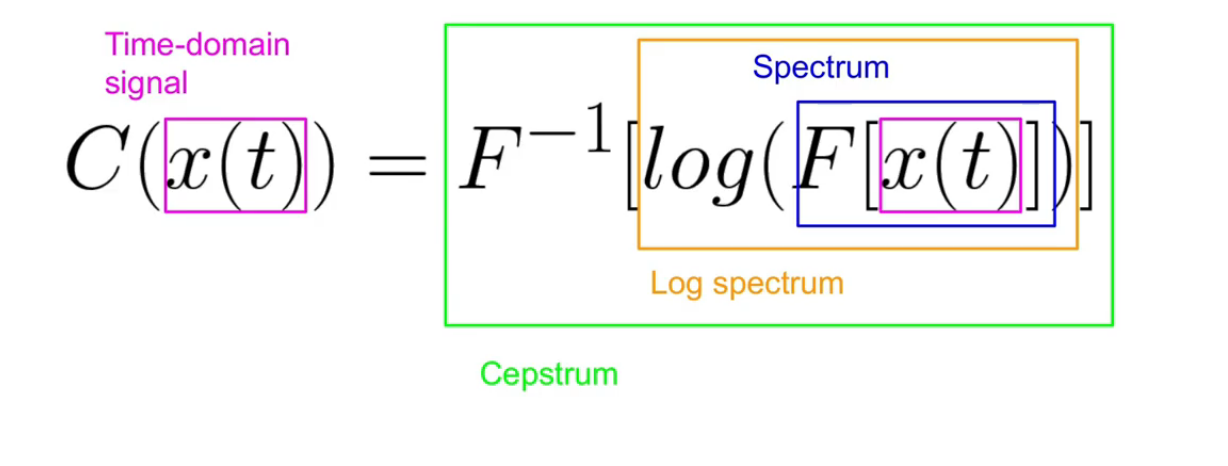
1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel-filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filter bank energies because humans perceive frequency logarithmically.
5. Take the DCT of the log filter bank energies.

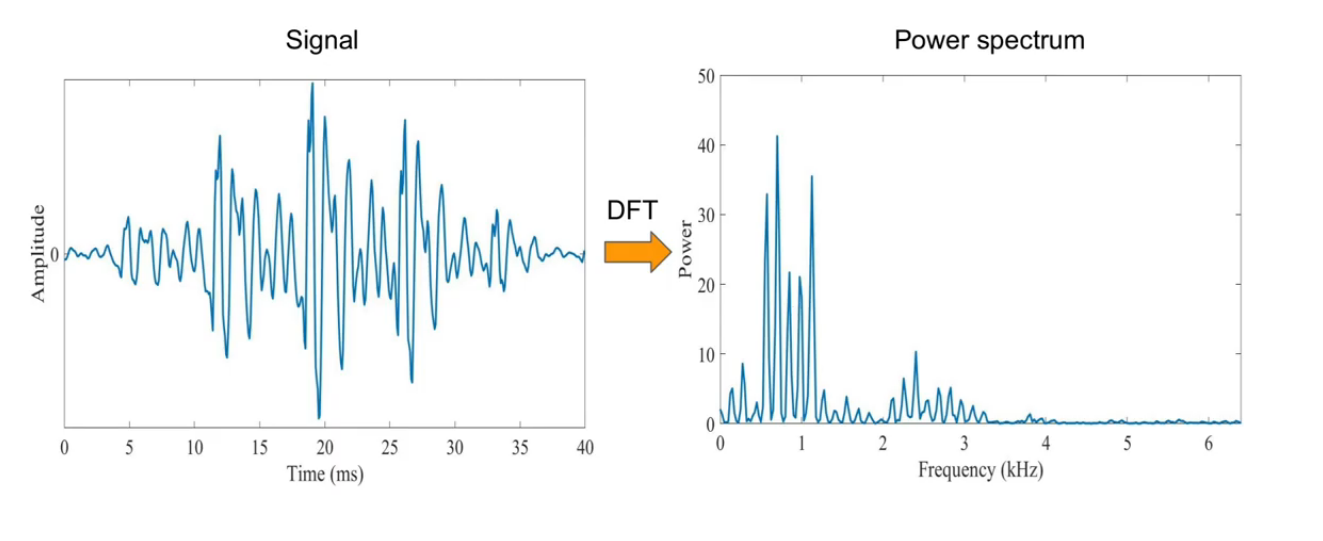
**Ideal Audio Feature**

1. Time-frequency representation.
2. Perceptually-relevant amplitude representation (This is non-linear and logarithmic and can be achieved using the vanilla spectrogram).
3. Perceptually-relevant frequency representation. (This representation can only be achieved using Mel spectrogram).

**Why MFCC?**

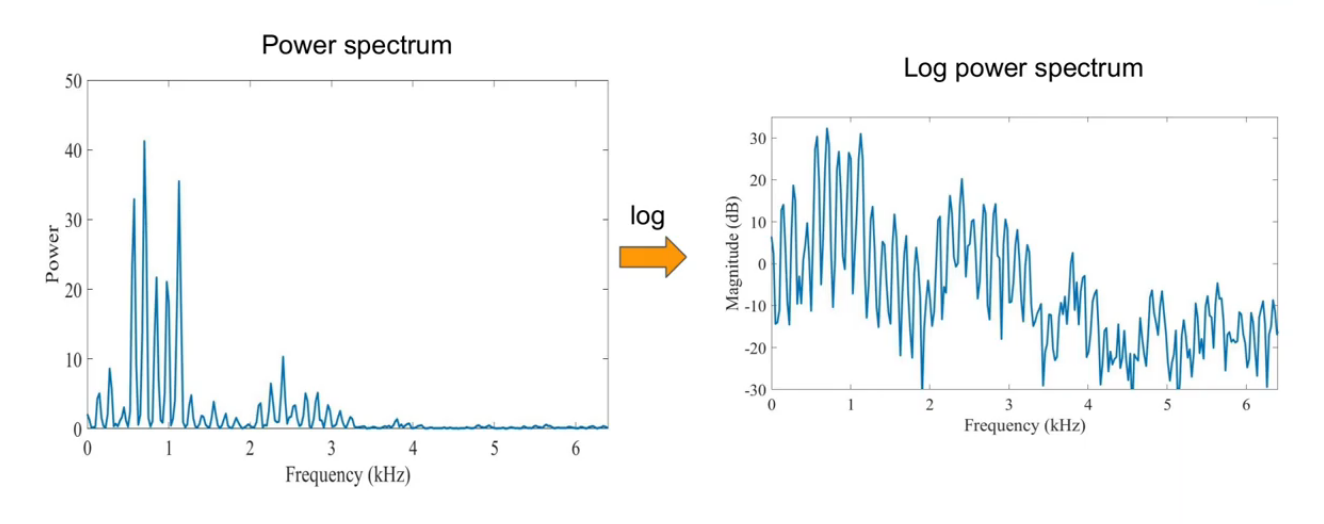
Mel scale is a scale that relates the perceived frequency of a tone to the actual measured frequency. It scales the frequency in order to match more closely what the human ear can hear (humans are better at identifying small changes in speech at lower frequencies because humans can perceive sound logarithmically). Using Mel scale, Mel spectrogram and the MFCC, we can make a deep learning model that can learn like a human. We use a Mel scale because it is a relevant scale for pitch. MFCC is used because it describes the “large” structures of spectrum, ignores fine spectral structures and works well in audio classification, speech and music recognition.

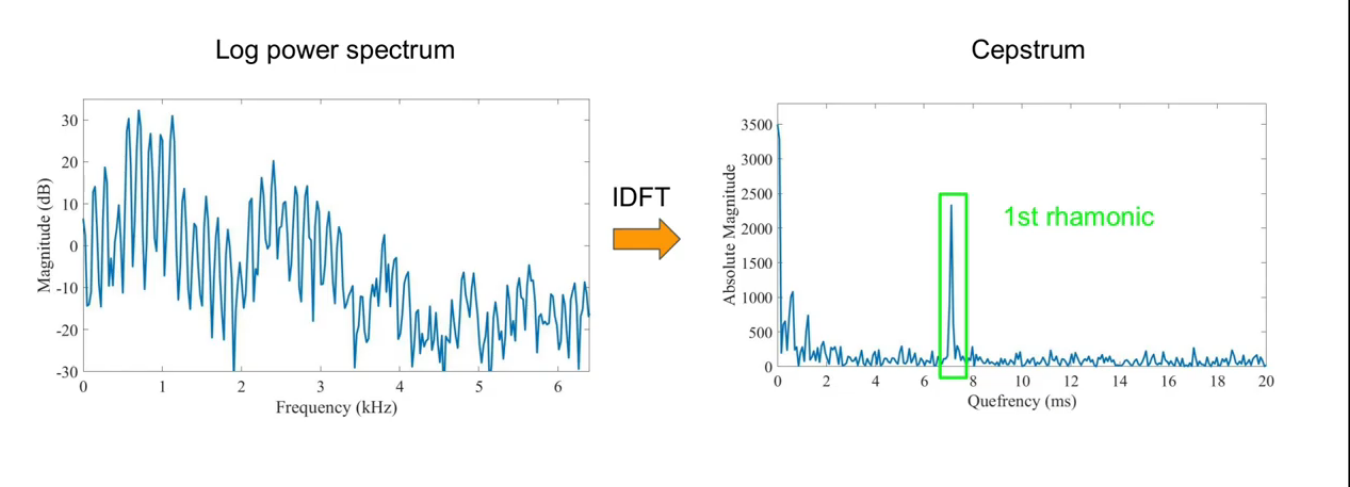


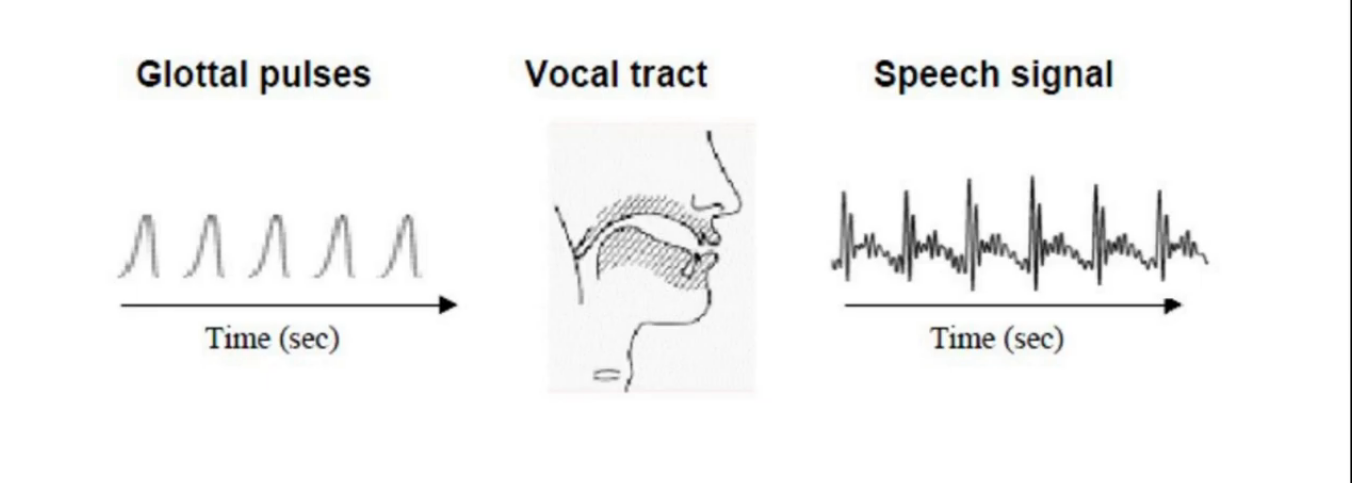


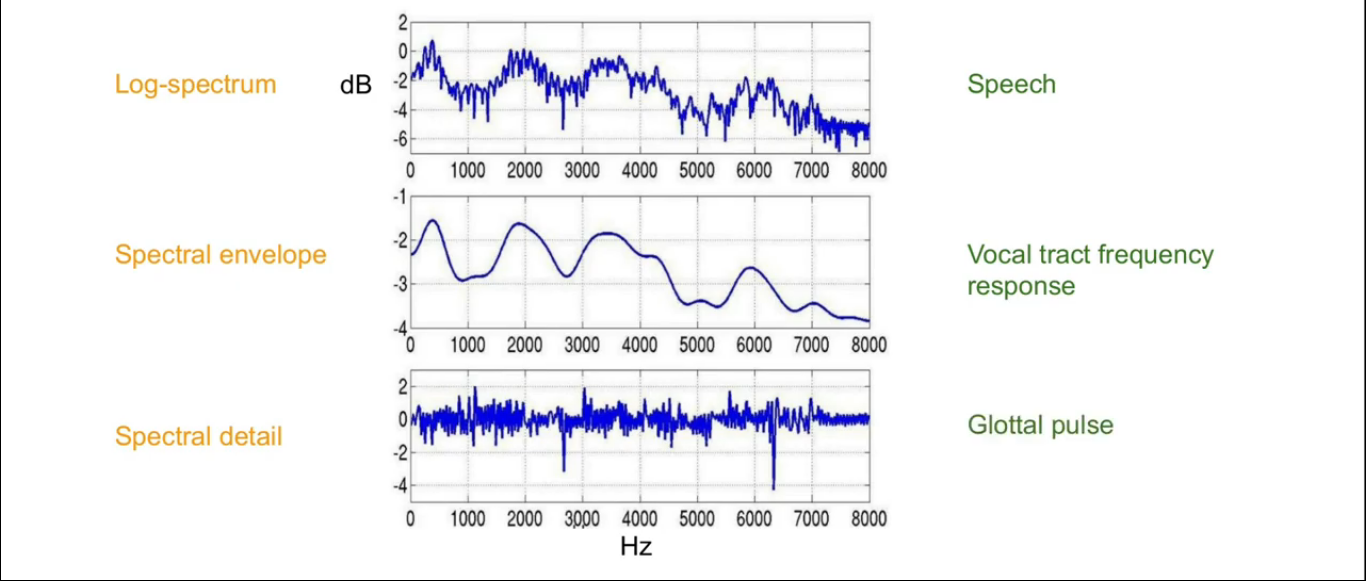
Discrete Fourier Transformation

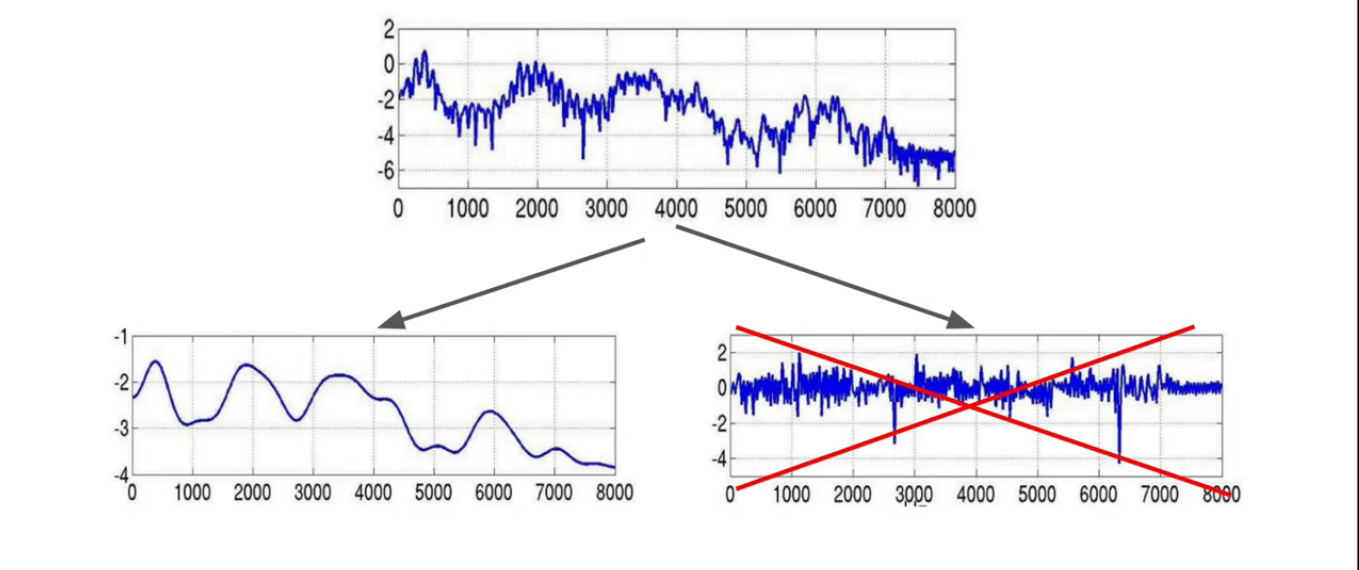
The Discrete Fourier Transform (DFT) is of paramount importance in all areas of digital signal processing. It is used to derive a frequency-domain (spectral) representation of the signal.

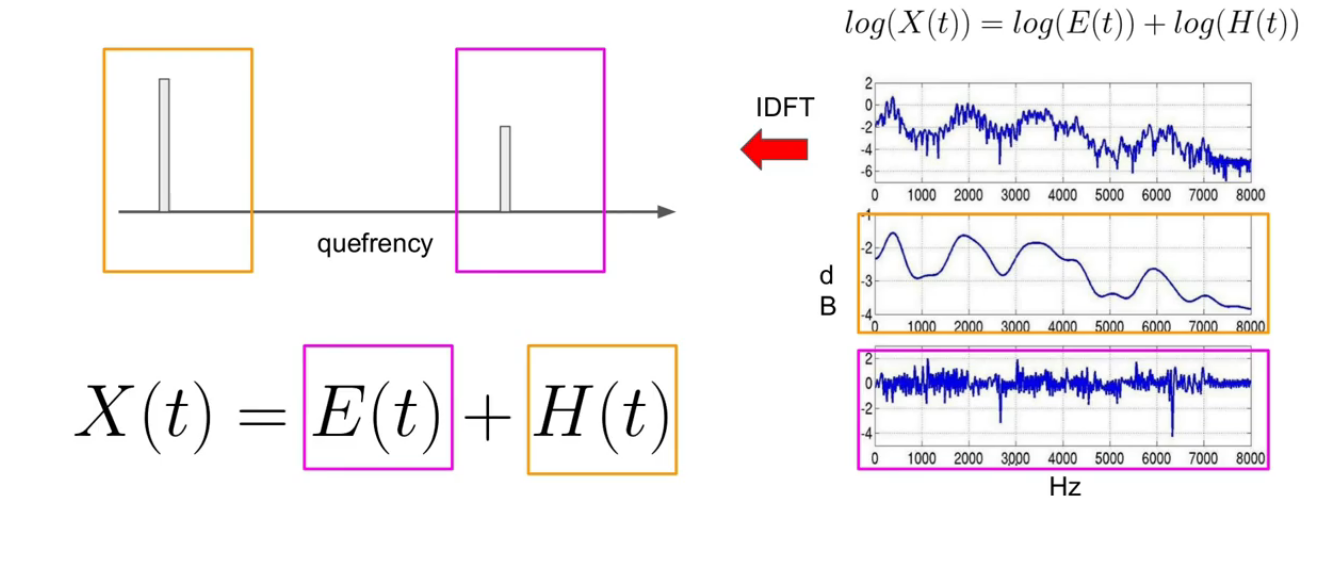




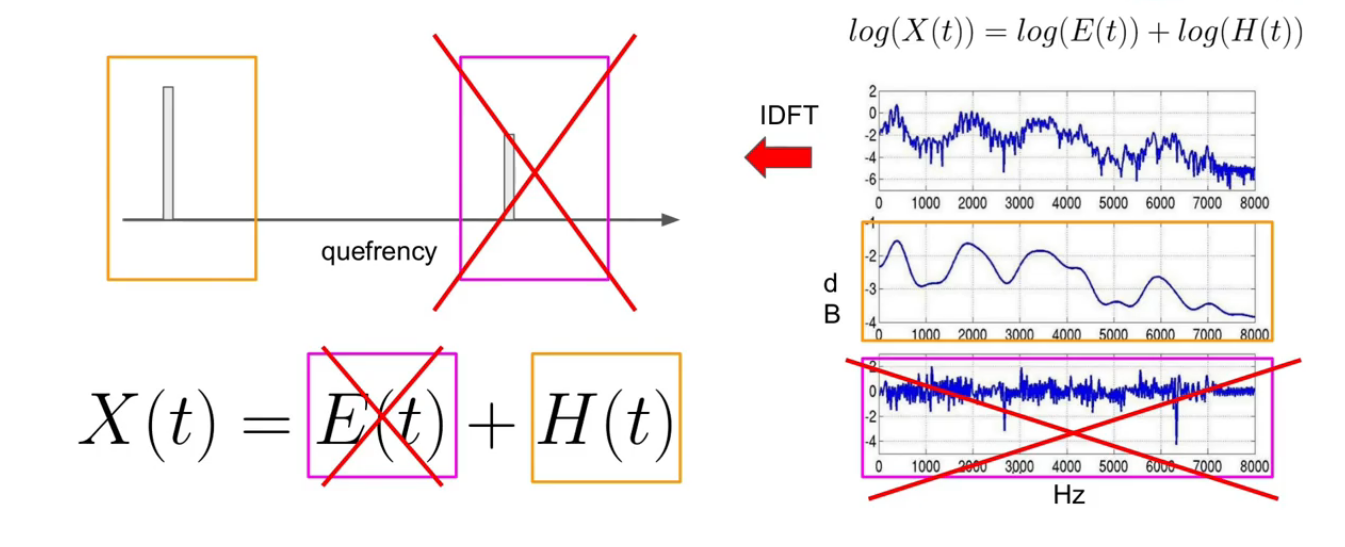






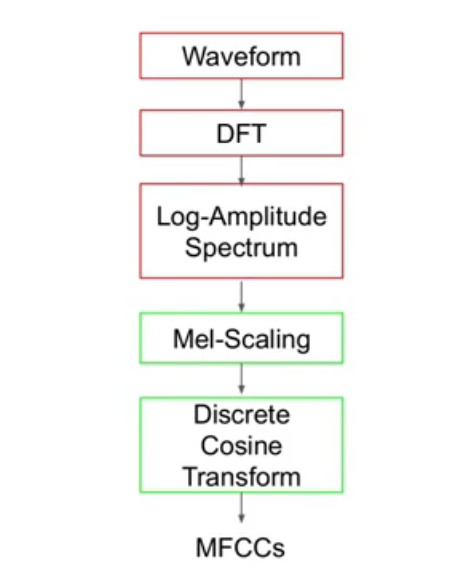


After applying liftering (Filtering) we can eliminate the glottal pulse.



**Mel Scale:**

The Mel scale is a scale of pitches judged by listeners to be equal in distance one from another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz tone, 40 dB above the listener's threshold, with a pitch of 1000 Mels.



The discrete cosine transform is similar to applying the inverse at the logarithmic spectrum.

Higher points in the **vocal tract frequency** represent **formants** (identities of sounds).

**Audio Channel**

We have observed that most of the samples have two audio channels (stereo) and few with one channel (mono).

**Mono** signals are recorded and played back using a single audio channel, while **stereo sounds** are recorded and played back using two audio channels.

**Libraries**

We have used mainly two important python built-in libraries which are as follow:

**Librosa:**

Librosa is python package for music and audio processing and will allow us to load audio in our notebook as NumPy array for analysis and manipulation. Librosa have mainly two purposes the first is to converge signals, it actually makes one signal that is the Mono. The second is to represent an audio signals with respect to normalized pattern between -1 and 1.

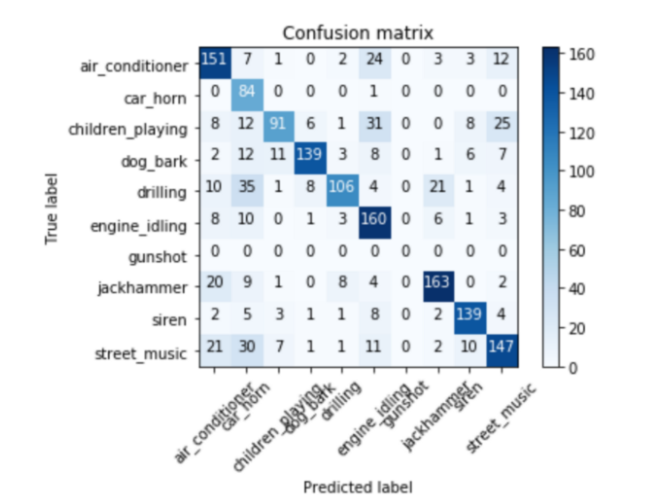
**TensorFlow:**

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming.

**NumPy:**

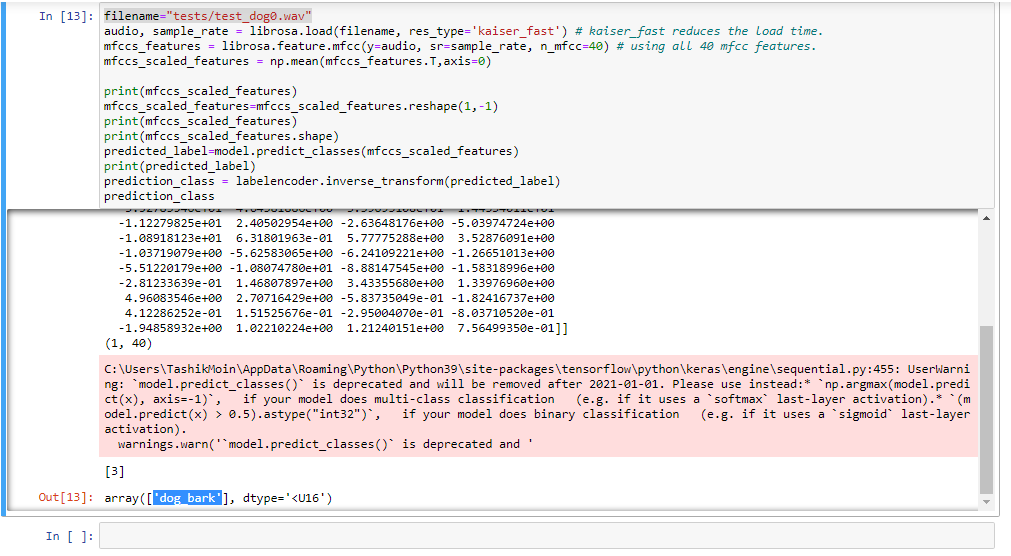
NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.

**Confusion Matrix:**

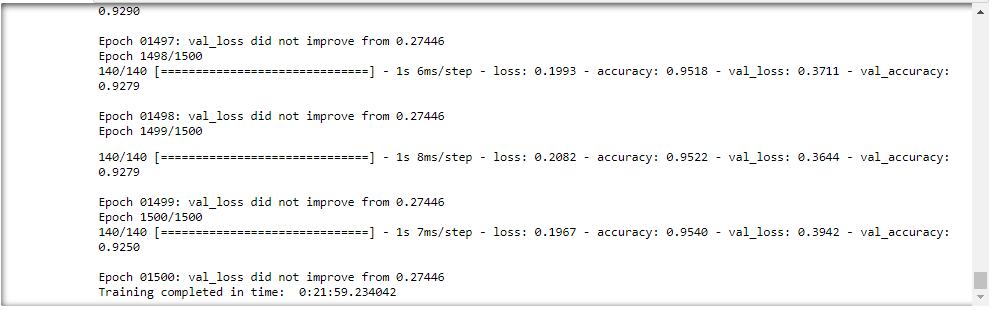


**Results:**

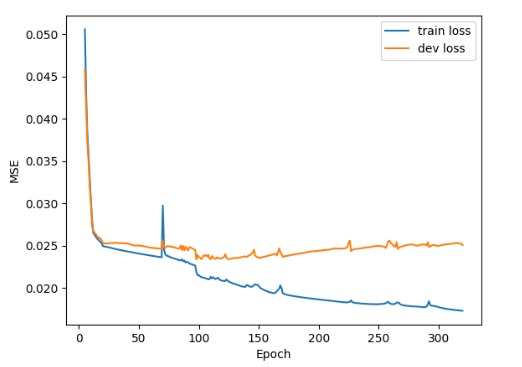
Our trained model obtained a Training accuracy of 98.19% and a Testing accuracy of 91.92%. The performance is very good and the model has generalized well, seeming to predict well when tested against new audio data.

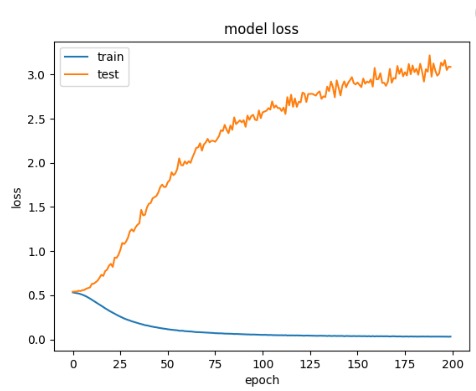


**Accuracy**



**Mean Square Error/ Loss Function**





**Hidden Layers and Output layer**

Three hidden layers and one output layer.



**References**

1. <https://research.google.com/audioset/>
2. <https://ieeexplore.ieee.org/document/7177954>
3. <https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5>