**Sound Event Detection & Classification Engine for Domestic Environment**

Problem Statement-

• Sound event detection in domestic environment can help in developing many new use cases in Voice Assistant solutions, such as generation of NLU and NLG responses based on environment context, trigger alarms in case of abnormal sound event detection etc.

• Goal of this project is to design and develop a sound detection engine that can recognize the overlapping events in domestic environment.

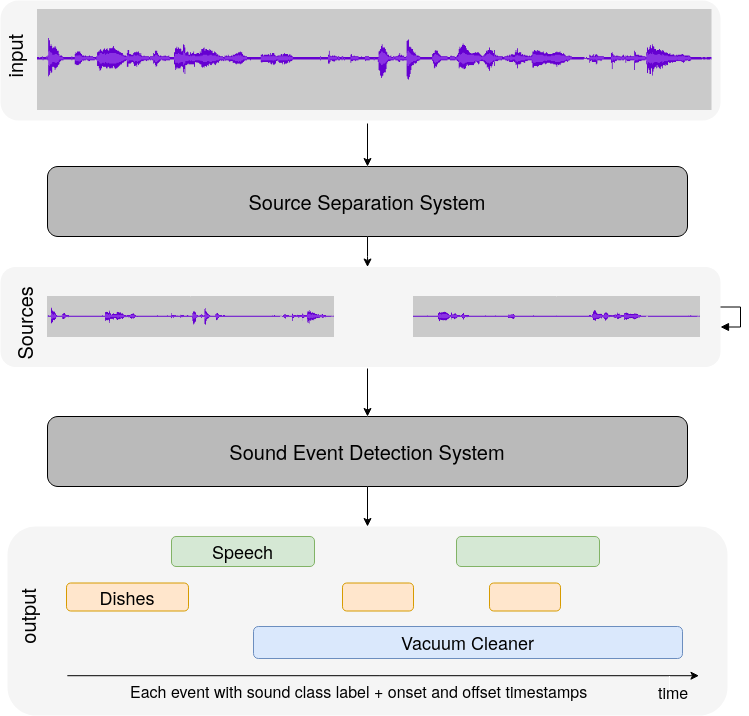
• In domestic environment multiple sound signals such as baby cry, dog bark, tap water, TV playing, Refrigerator etc. are received in sound engine in parallel. With help of extra information such as device’s operating state, human/animal presence etc. the detection accuracy can be enhanced.

• Input to the engine will be sound signal with overlapping sound events and additional information and output will be accurate classification of sound events.

• Engine should classify both speech and non-speech audio accurately and can be trained with new categories of sound data.

Procedure-

1-Block diagram

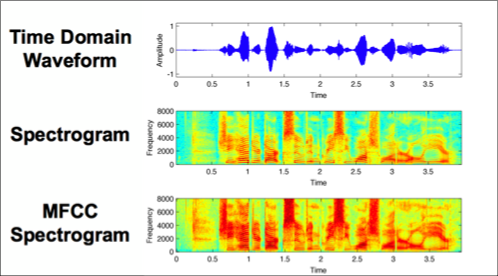
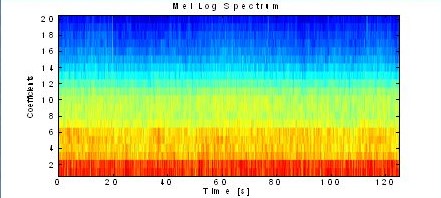


*sound event detection system with a sound separation pre-processing.*

2. Feature engineering

Methods for feature engineering are

* MFCC
* Log-mel
* Mel energy

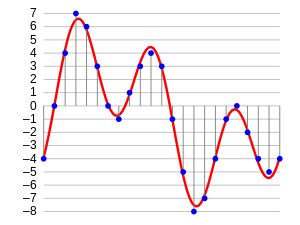
MFCC Mel Log spectrum

3. audio mixing method, duration.

•We have a data set containing sounds of various living beings and non living beings in a domestic environment like birds, animals, sound of honking cars etc.

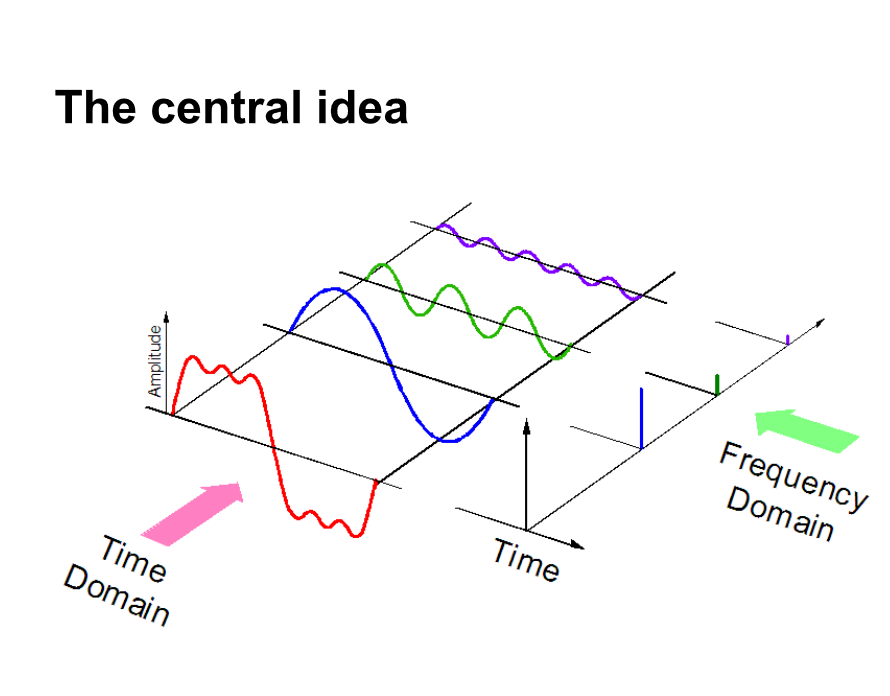
•As with all unstructured data formats, audio data has a couple of pre-processing steps which have to be followed before it is presented for analysis.

•The first step is to actually load the data into a machine understandable format. For this, we simply take values after every specific time steps. For example; we can extract values at half a second. This is called **sampling of audio data,** and the rate at which it is sampled is called the **sampling rate.**



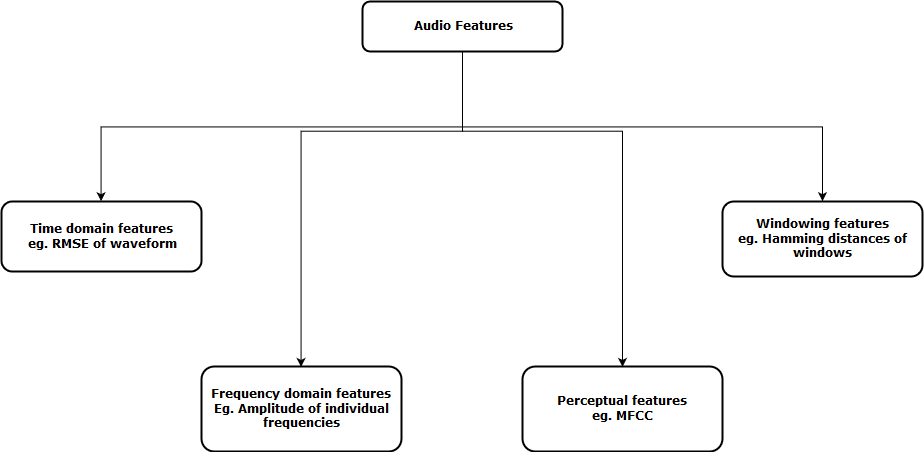
•Another way of representing audio data is by converting it into a different domain of data representation, namely the frequency domain. When we sample an audio data, we require much more data points to represent the whole data and also, the sampling rate should be as high as possible.

•On the other hand, if we represent audio data in **frequency domain**, much less computational space is required. To get an intuition, take a look at the image below



•Here, we separate one audio signal into 3 different pure signals, which can now be represented as three unique values in frequency domain.

•Now the next step is to extract features from this audio representations, so that our algorithm can work on these features and perform the task it is designed for. Here’s a visual representation of the categories of audio features that can be extracted.



4.LSTM with CNN MODEL

•LSTM neural network is a special kind of RNN, that doesn’t suffer from vanishing gradient problem and is able to learn long-term dependencies.

•LSTM consists of a set of subnets, known as memory blocks. Each block includes the memory cell and three units: input, output and forget gates.

•LSTM layer maps the input sequence X = (x1, x2,...xT ) to the output sequence Y = (y1, y2,...yT ) in according to the equations:

it = sig(Wxixt + Wyiyt−1 + bi), (1)

ft = sig(Wxfxt + Wyf yt−1 + bf ), (2)

ct = ft ct−1 + it tanh(Wxcxt + Wycyt−1 + bc), (3)

ot = sig(Wxoxt + Wyoyt−1 + bo), (4)

yt = ottanh(ct), (5)

wherect is the state of the memory cell and it, ft, ot are gate outputs at time t. The network weights W and biases b are tuned during learning to minimize the loss function. In case of a multi-layer structure the input of the next layer is the output of the previous one.

•Our model for sound classification is composed of two LSTM layers followed by dense layer with softmax activation function.

•LSTM produces a sequence, only the last value is propagated to the output layer. The first two layers contain 128 and 64 units, the last layer has 10 units, one per sound class. To reduce overfitting dropout with a rate of 0.25 is applied to the output of the LSTM layers.

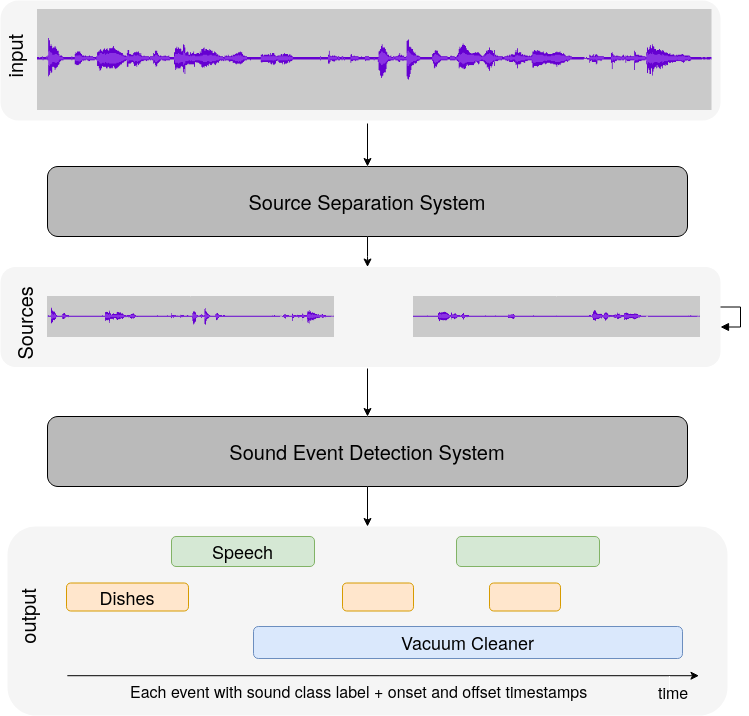
•For training categorical cross-entropy loss function is minimized using Adam optimizer. Because of long training time a full search of hyperparameters is infeasible, thus, the most promising combination was found using single fold evaluation.

•To evaluate the performance of proposed model we use dataset.

•Along with our model we run a baseline CNN . CNN is composed of three convolutional layers followed by two dense layers. Both networks were trained on magnitude mel-spectrogram and CNN model indicated even better performance than was reported in for log-scaled melspectrogram.

6. API for training and testing new sound dataset – In this api will be developed and trained model will be served in the backend. In this new sound data can be provided for testing the model.

New sound dataset will be given as input.



Api