# Single Event Detection-Sound Files

**Aditya Krishna Rao**
Department of Electrical Engineering
IIT Kanpur
adikrao@iitk.ac.in

## 1 Introduction

For this assignment, we are given a dataset of 1000 spectrograms with their labels (100 from each class). We are required to identify to which event the given audio's spectrogram belongs. This report will discuss our approach, method, experiments, and results for the task given.

The parameters for the Mel Spectrograms of audio files are ass follows:

Nfft = 2048  nmels = 128  hop length = 512  windowing function = hann

## 2 Literature Survey

Various signal processing methods and machine learning, such as matrix factorization, dictionary learning, etc., have been applied for sound classification. But as cited in a research paper, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," neural networks, particularly CNN gives the best results. When applied to spectrogram-like inputs, they can capture energy modulation patterns across time and frequency, which is an important trait for distinguishing between different, often noise-like, sounds such as engines and jackhammers. Second, by using convolutional kernels (filters) with a small receptive field, the network should, in principle, be able to successfully learn and later identify spectro-temporal patterns that are representative of different sound classes.

## 3 Method

### 3.1 Data Preparation

We have been given a dataset of 1000 labeled spectrograms that belongs to 10 sample classes. For implementing the CNN model, we need one dimension for its input. Our given dataset of spectrograms has different dimensions, and for implementing CNN, we need to preprocess our data accordingly. For this purpose, after going through certain references, we decided to go with two methods.

- In the first method, we fix the dimension for the CNN model as (128,200). We find a central frame which has maximum amplitude. This is done by simply summing along each frame and finding the maxima. Then we take a width 200 with 100 on each side. Next we average along the nmels to find the average contribution of each frequency. Final data point was of dimensions (128,1).
- In the second method, we decided to take the fixed dimension as (128,400). Then we apply a convolution of window (1,20) with hop of 10. Unlike the previous case, this method takes into account the temporal distribution of frequency and in a way capturing the waveform of the sound pulse, thereby giving more data for the model to train. Final data point was of dimensions (128,2432)

After we have pre-processed the data, we train our model on the dataset of 1000.

### 3.2 Model Deatils

Each model was single layered activated by softmax to give the corresponding classifier probabilities.The loss was calculated by categorical loss function and was minimised to 0.307 over around 15000 iteration. None of the libraries like tersorflow/keras were used and all modeled were prepared from scratch. Libraries used were Numpy and Pandas for vector manipulations.

## 4 Results

I calculated precision, recall and f1 score for each class and then took a average for finding the final value to account for difference in normal f1 due to more samples from a particular class.

### 4.1 Confusion Matrix

For model with dim(128,2432)

[[14. 2. 1. 1. 1. 0. 0. 1. 2. 2.]
[ 4. 10. 0. 1. 1. 1. 0. 0. 1. 0.]
[ 1. 0. 13. 1. 1. 0. 1. 1. 0. 0.]
[ 1. 0. 0. 11. 1. 3. 1. 1. 0. 0.]
[ 1. 0. 0. 1. 12. 0. 0. 1. 0. 3.]
[ 0. 0. 0. 1. 1. 16. 0. 0. 0. 0.]
[ 0. 0. 0. 0. 1. 2. 8. 0. 0. 7.]
[ 2. 1. 2. 4. 1. 1. 0. 12. 4. 0.]
[ 0. 0. 0. 5. 1. 1. 0. 0. 16. 1.]
[ 0. 1. 0. 2. 0. 5. 1. 0. 1. 8.]]

For model with dim(128,1)

[[11. 4. 2. 4. 0. 1. 0. 0. 2. 0.]
[ 3. 11. 1. 0. 1. 1. 0. 0. 1. 0.]
[ 1. 0. 14. 0. 0. 0. 1. 0. 2. 0.]
[ 0. 0. 0. 13. 1. 2. 0. 0. 0. 2.]
[ 2. 0. 0. 1. 11. 1. 0. 1. 1. 1.]
[ 0. 0. 0. 0. 1. 16. 0. 0. 1. 0.]
[ 0. 0. 0. 1. 0. 1. 13. 0. 0. 3.]
[ 3. 1. 4. 3. 1. 1. 0. 10. 4. 0.]
[ 0. 0. 1. 4. 0. 0. 0. 0. 19. 0.]
[ 0. 0. 0. 3. 1. 2. 2. 0. 3. 7.]]

### 4.2 Precision, Recall and F1

Class index in following matrix-
Bark:0, Crying and sobbing:1,Doorbell:2, Knock:3,Meow:4,Microwave oven:5, Shatter:6, Siren:7, Vehicle horn and car horn and honking:8, Walk and footsteps:9 The first element is precision, second is recall and third is F1 score for each tuple present below.
For model with dim(128,1)

[[0.55 0.45833333 0.5 ]
[0.6875 0.61111111 0.64705882]
[0.63636364 0.77777778 0.7 ]
[0.44827586 0.72222222 0.55319149]
[0.6875 0.61111111 0.64705882]
[0.64 0.88888889 0.74418605]
[0.8125 0.72222222 0.76470588]
[0.90909091 0.37037037 0.52631579]
[0.57575758 0.79166667 0.66666667]
[0.53846154 0.38888889 0.4516129 ]]

For model with dim(128,2432)

[[0.60869565 0.58333333 0.59574468]
[0.71428571 0.55555556 0.625 ]
[0.8125 0.72222222 0.76470588]
[0.40740741 0.61111111 0.48888889]
[0.6 0.66666667 0.63157895]
[0.55172414 0.88888889 0.68085106]
[0.72727273 0.44444444 0.55172414]
[0.75 0.44444444 0.55813953]
[0.66666667 0.66666667 0.66666667]
[0.38095238 0.44444444 0.41025641]]