

Major Project Phase - II
Review – 1

Detecting Logging of Forest Trees using Sound Event Detection

Team Members (Team - 17)

19K41A0594 - B.RAJU

19K41A0510 - JINUKALA VAMSHI

19K41A0517 - MOHAMMED RAAMIZUDDIN

19K41A05E9 - BHONAGIRI SHREYA

19K41A05F7 - JUPALLY YOCHITHA

Project Guide

Sallauddin Mohmmad

Assistant Professor

Introduction

- Domain:

- Prevention of illegal logging of forest trees and preserving natural resources.
- As tree cutting generates lot of noise, it can be detected by regularly monitoring the acoustic signals inside the forest.

- Implementation Domain:

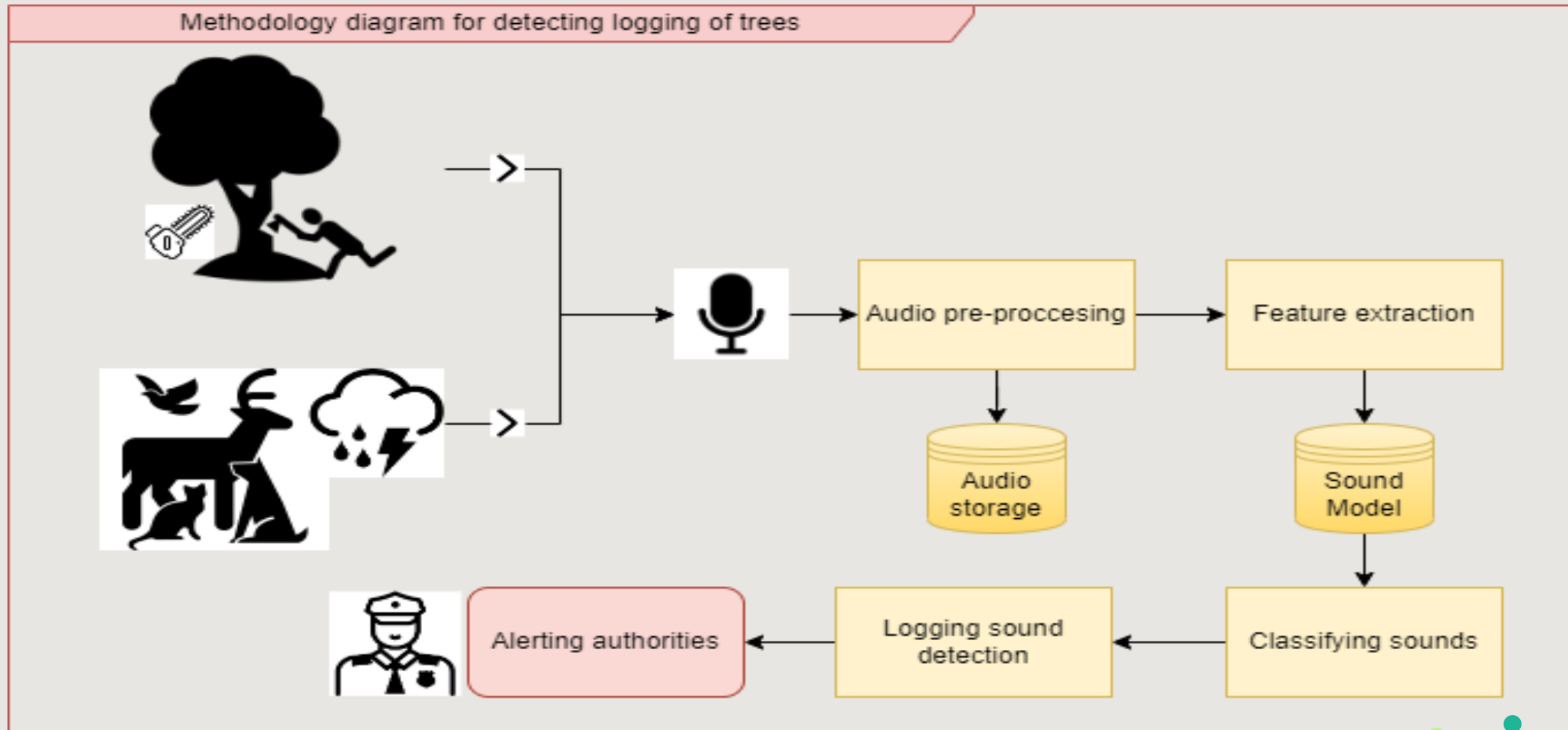
- In the context of our project domain, we will be using a deep learning techniques.
- Deep learning is a type of Artificial Intelligence that imitates the way humans gain certain type of knowledge.

Problem Statement

- In forests, tree cutting activities are illegal but due to shortage of manpower and other resources, governments are not very successful in curbing this menace.
- India is placed on third position for illegally importing logged timber in the world.
- The issue of this level illicit logging must be dealt very seriously as it exhausts the forest assets.
- Monitoring the forest assets visually requires a lot of equipment.
- An acoustic signature can provide valuable information about the activities of any intruder inside the forest.

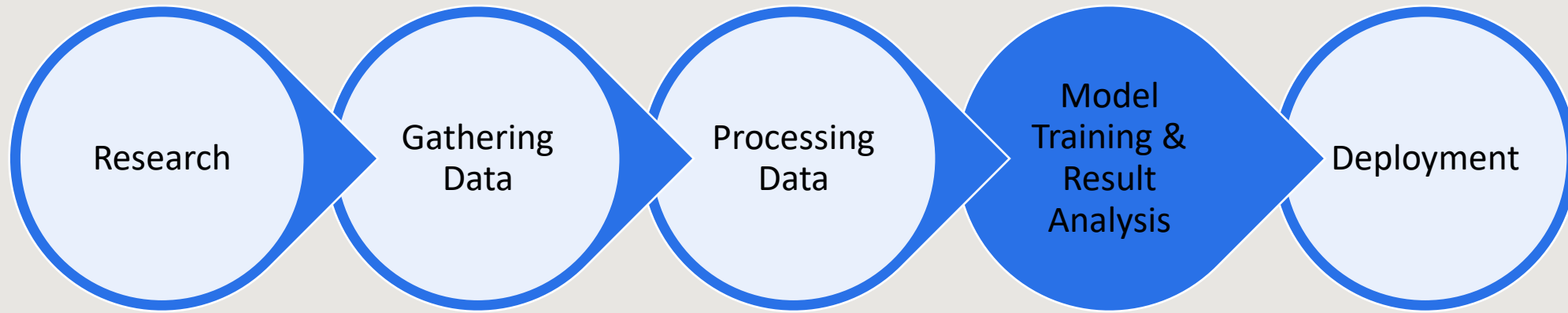
Methodology

- The solution to the mentioned problem can be achieved by using the below workflow :



Block Diagram

Project Timeline







Research Outcomes

- The research done on sound event detection usually comprises of the following steps:
 1. Collecting the sound signals or audio data,
 2. Extracting the crucial features from the audio data,
 3. Making clusters of identical features or labelling the data,
 4. Classifying the features.
- Most of the authors used the coefficients of Mel frequency cepstrum to extract the features which are used for classifying the sounds by CNN or GMM.
- Further, after these features are classified, the environmental audio scene is also identified using the Neural Network.

Dataset

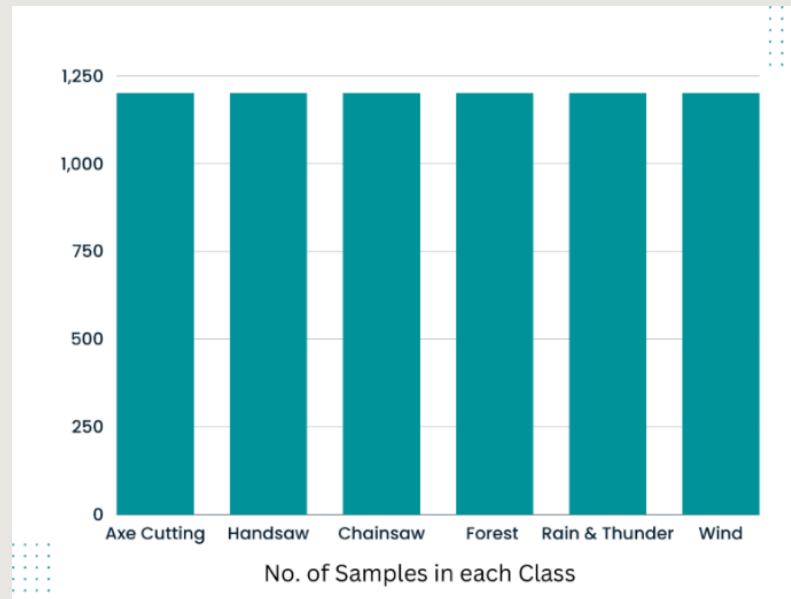
- The dataset that we will be using is an environmental dataset that contains of Tree logging sounds as well as the sounds that can be classified into this class by collecting the sound samples corresponding to it.
- The dataset should also contain some negative classes, complexity of the dataset would be high.
- The Dataset considered consists of six classes (Chainsaw, Handsaw, Axe cutting, Wind, Forest, Rain and Thunder Sounds) that should be categorized according to the samples collected.

 Wind_1	05-02-2023 10:46	WAV File	1,723 KB
 Wind_2	05-02-2023 10:46	WAV File	1,723 KB
 Chainsaw_1	03-02-2023 15:44	WAV File	1,876 KB
 Chainsaw_2	03-02-2023 15:44	WAV File	1,876 KB

Collected Audio Files

Dataset (Contd.)

- The audio data was collected from various sources such as AudioSet and by manually collecting the data. Each class of audio had hours content in it, the audio was split into ten seconds from the hours long file to get more samples.
- Each class has 1200 samples. Therefore, 7200 audio files of ten seconds each for all the classes.

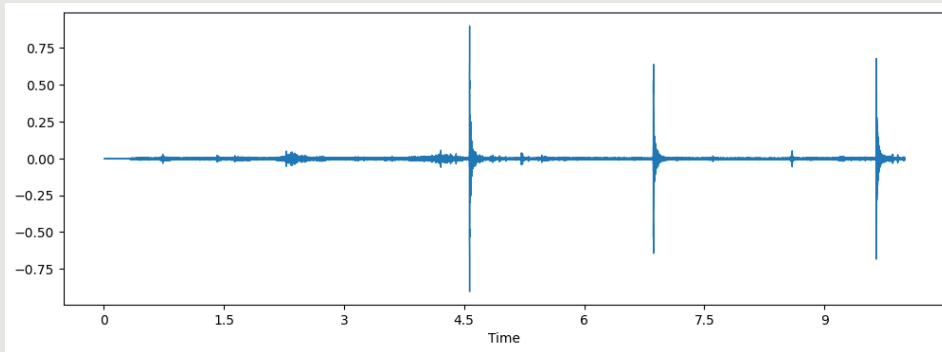


Audio Pre-Processing

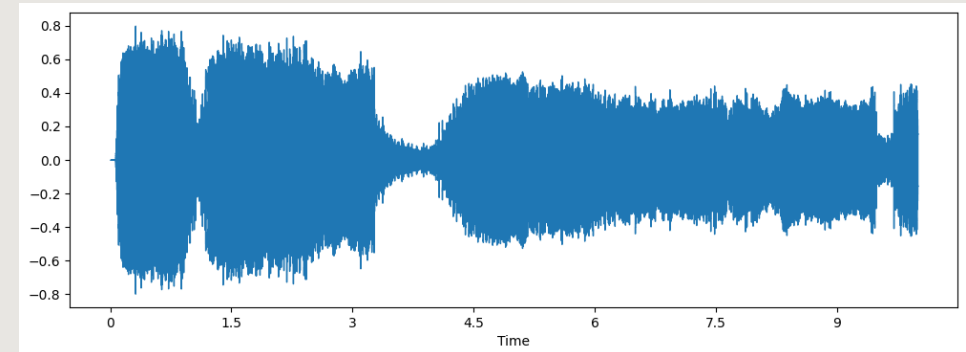
- Sampling and Sampling Frequency determination.
 - sampling is the reduction of a continuous signal into a series of discrete values.
 - The sampling frequency or rate is the number of samples taken over some fixed amount of time.
- Amplitude determination.
 - The amplitude of a sound wave is a measure of its change over a period (usually of time).
- Bit-rate Conversion.
 - Bit-rate is the number of bits per second that can be transmitted along a digital network.
- Audio Channel Manipulation.
 - Audio Channel is a single stream of recorded sound with a location in a sound field.
- Bit-depth Conversion.
 - The audio bit depth determines the number of possible amplitude values we can record for each audio sample.

Audio Waveforms

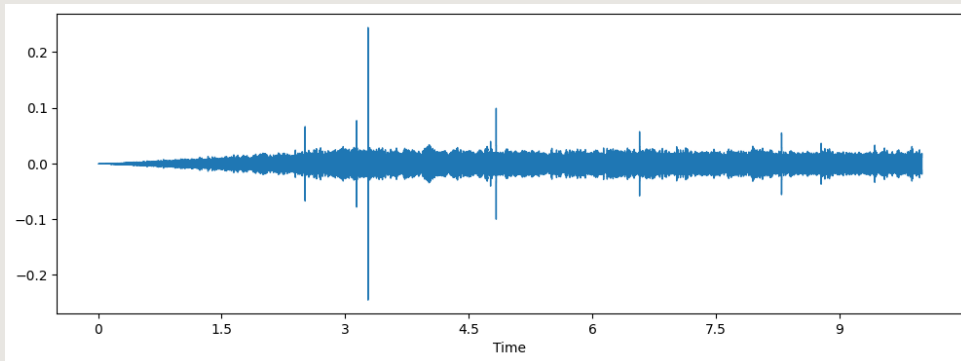
- Below are the audio waveform representations of some audio classes after applying the pre-processing steps:



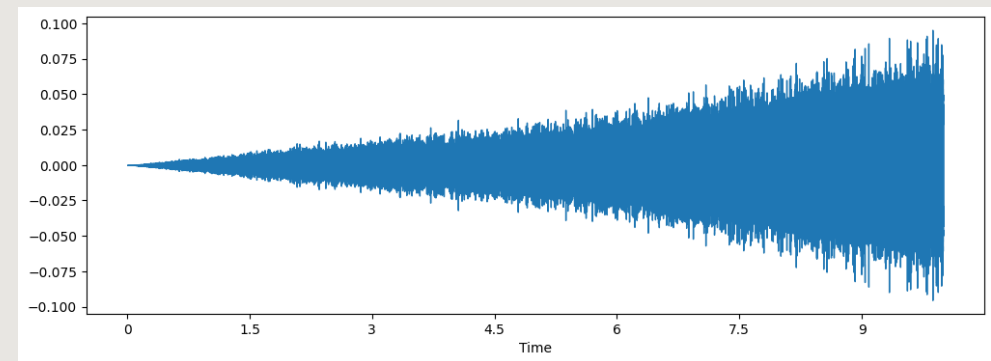
Axe cutting Sounds



Chainsaw Sounds



Forest Sounds



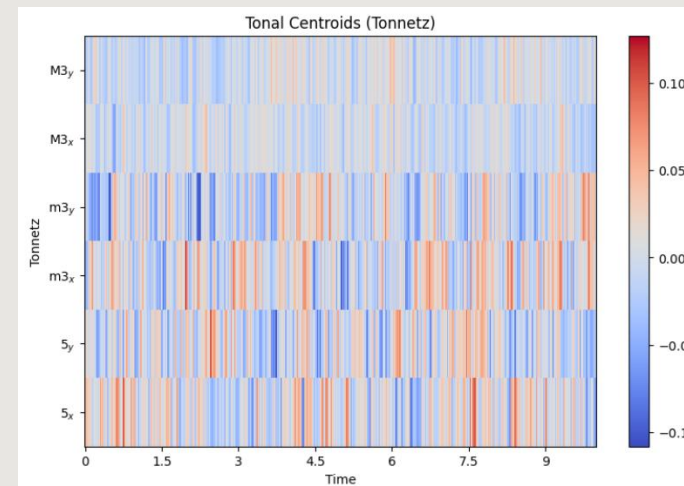
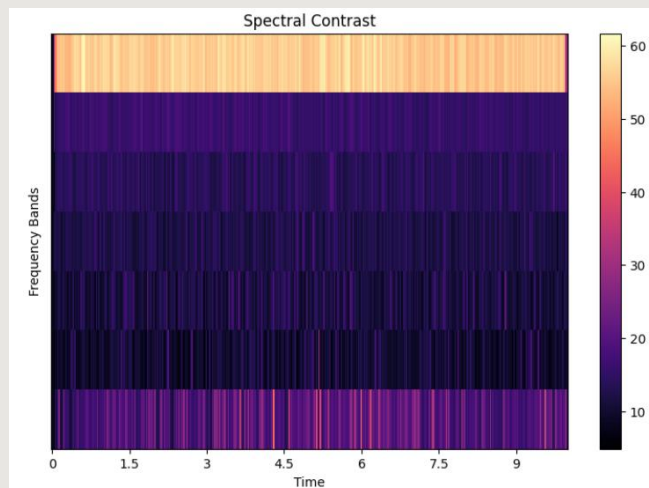
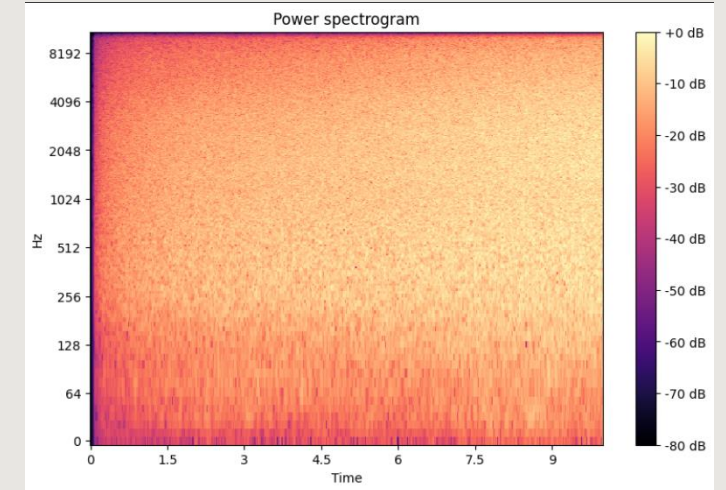
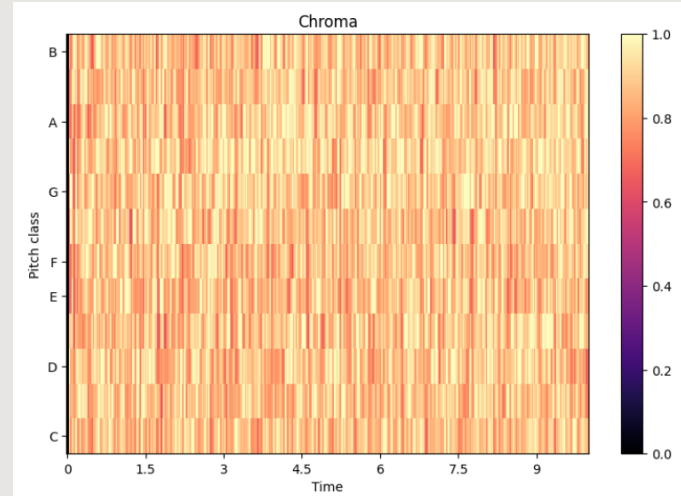
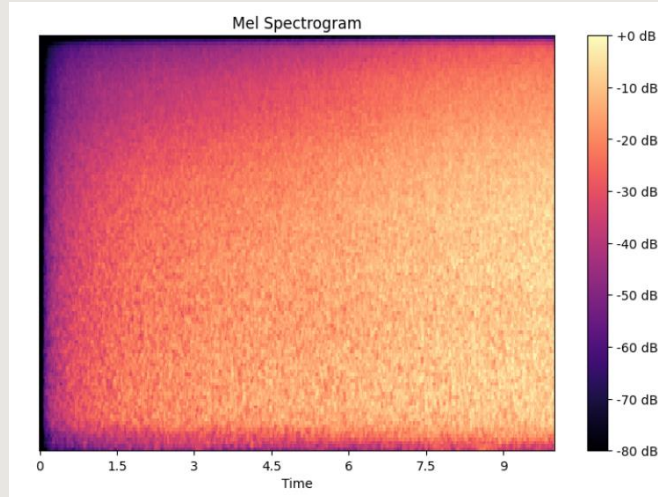
Wind Sounds

Feature Extraction

- MFCC
 - The Mel-frequency Cepstrum coefficient is a representation of the short-term power spectrum of a sound.
- SPECTRAL CONTRAST
 - The Spectral Contrast used to make decibel difference between peaks and valleys, that helps for enhancing of sounds.
- MEL-SPECTROGRAM
 - Mel-Spectrogram is used for rendering the frequencies above certain threshold.
- CHROMA
 - The Chroma value of an audio basically represent the intensity of the twelve distinctive pitch classes that are used to study music.
- TONNETZ
 - Tonnetz is used for computing the tonal centroid features of sound.

Feature Extraction (Contd.)

- Below are the representations of various feature extraction techniques:



Applying Feature Extraction

- The applied feature extraction techniques are:

- Mel-frequency cepstral coefficients (MFCC)
- Chroma (STFT)
- Mel Spectrogram
- Contrast
- Tonnetz

Actual Matrices

```
Sampling Rate : 22050
Shape of Audio File : (220500,)
Shape of MFCC Matrix : (40, 431)
Shape of STFT : (1025, 431)
Shape of Chromagram Matrix : (12, 431)
Shape of Mel Spectrogram Matrix : (128, 431)
Shape of Spectral Contrast Matrix : (7, 431)
Shape of Tonal Centroid Features Matrix : (6, 431)
```

Model Training Features

```
Length of MFCC Features : 40
Length of Chromagram Features : 12
Length of Mel Spectrogram Features : 128
Length of Spectral Contrast Features : 7
Length of Tonal Contrast Features : 6
```

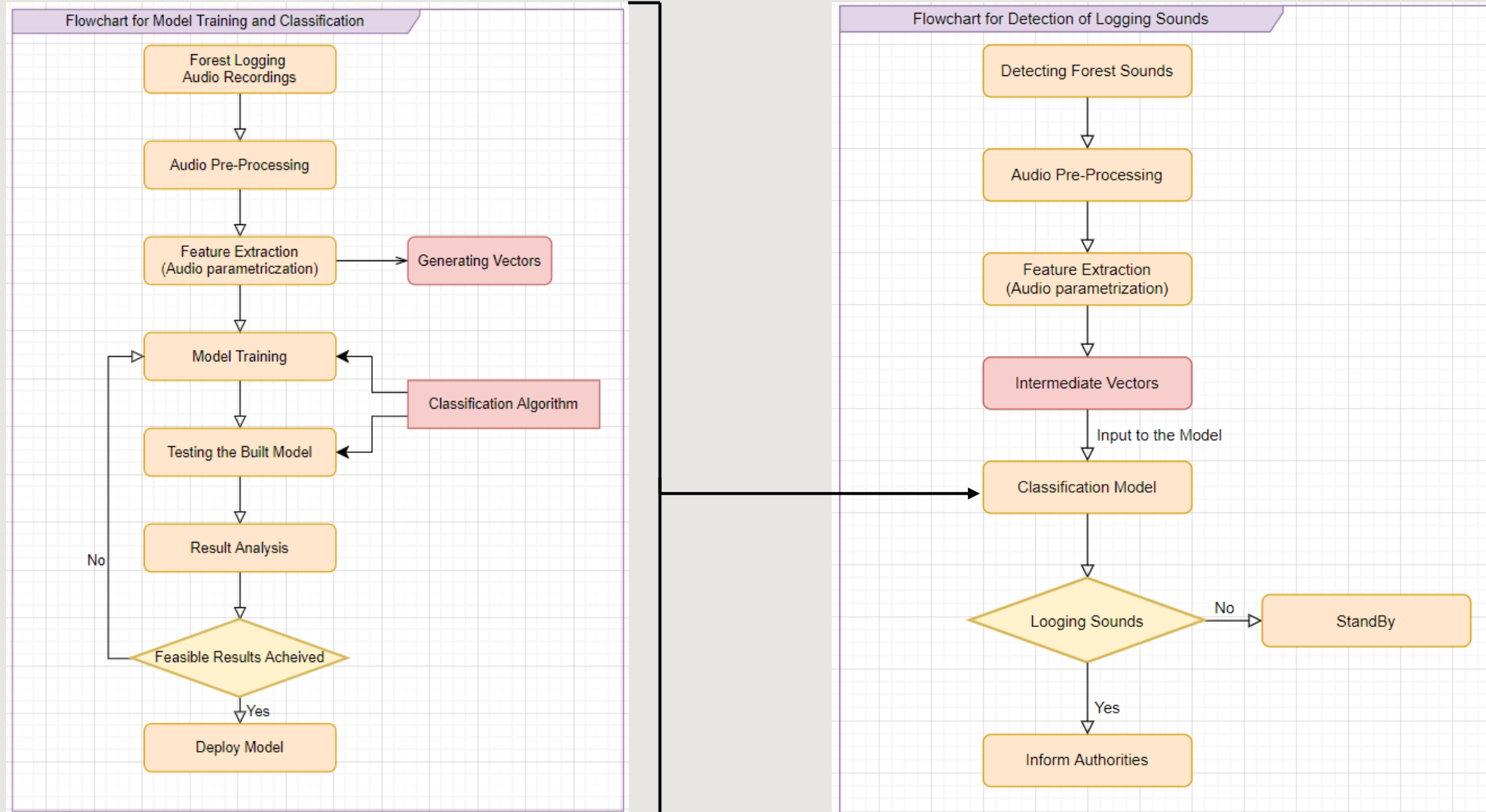
Feature Dimensions

Time-series Dataset

MFCC 0	MFCC 1	MFCC 2	MFCC 3	MFCC 4	MFCC 5	MFCC 6	MFCC 7	MFCC 8	MFCC 9	MFCC 10
-97.7864	86.25643	-0.14897	25.04751	-2.49172	8.204606	-1.12128	6.870169	-10.5541	9.92381	-7.45715
-330.336	48.04	-15.0077	21.51904	-5.75371	1.817559	-13.8778	-5.20551	-10.6022	-1.71919	-9.94648
132.6741	29.82393	-12.1078	21.10336	-19.6643	-2.86029	-10.8468	3.277118	-0.16802	1.372277	-5.92728
-27.0878	24.45839	-25.2705	-4.32539	-23.0208	-14.0775	-22.2997	-15.6152	-20.3026	-10.9955	-15.7231
-78.6029	161.7938	-73.2197	67.81618	-11.1815	11.02398	2.012776	-9.29883	34.67146	-22.7372	6.542136
47.89653	14.56548	-48.7664	35.50513	-13.2396	-21.8665	8.649476	3.224844	-13.0726	8.345982	-5.19494
-104.616	70.90969	3.295435	1.263251	-6.18473	-0.84467	-7.2061	2.220061	-7.23685	4.379762	-4.7575
109.3782	7.088119	11.25363	0.966891	2.606588	8.550988	-3.38388	-7.78959	5.291799	9.438836	-8.16111
29.52477	12.53689	8.920549	-12.1628	3.880223	-8.59329	3.455522	-8.92519	-4.86276	-1.25317	-4.85197
-143.731	182.6462	-64.725	33.94789	-17.1267	-3.02493	15.42898	-12.135	10.38894	-4.71895	0.606314
-18.2287	71.28271	-21.0842	20.02187	9.745628	-26.8885	-0.95378	-12.2228	-19.6097	-4.59399	-7.20963
-491.192	50.91351	-0.63117	19.79546	-11.4158	0.796699	-0.04829	1.791018	5.729762	-2.34494	-0.40979
-64.2932	100.4025	-16.2503	26.18894	5.213824	12.52853	-1.75689	8.556458	-13.1366	9.782	-4.54089
-237.892	206.935	-139.782	29.33392	5.047009	-67.5778	19.12837	-11.8375	-19.2222	25.75368	-8.90576
-217.804	-6.52717	-110.5	33.12815	-82.7203	36.39346	-62.2686	16.61922	-28.3318	4.30646	-7.78515
-117.648	101.3888	-33.3201	20.78649	-9.13334	-0.09123	-6.2287	0.098537	-11.4263	-1.58594	-7.50516
-104.717	24.15383	1.636157	17.13857	-20.2773	-4.03332	-11.6023	-1.01728	-7.00364	2.116731	-9.49313
-148.966	12.31925	-39.0631	24.97993	19.0212	-10.7939	-2.95612	21.48101	16.57922	0.90281	-8.52537
20.22687	1.256859	-18.0231	3.430336	-9.82683	-7.63848	-21.7697	3.580827	-16.3764	6.943383	-12.6126

Extracted Features

Flowchart for Model Training



Models Used

- After extracting the feature vectors as dataset, then we performed the model training. We used various models and compared them.
- The models considered for analysis at this stage are:
 1. Convolutional Neural Network (CNN).
 2. Bi-directional Convolutional Recurrent Neural Network (BICRNN).
- The models developed are custom, the architecture of the networks is not pre-determined, the layers of the networks were determined according to the time-series data generated.
- These models are considered for analyzing the results and improving the efficiency of classification further by altering the architecture or tuning the hyperparameters.

Model Architecture

- The architectures of models are represented below:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 191, 64)	256
conv1d_1 (Conv1D)	(None, 189, 64)	12352
max_pooling1d (MaxPooling1D)	(None, 63, 64)	0
conv1d_2 (Conv1D)	(None, 61, 128)	24704
conv1d_3 (Conv1D)	(None, 59, 128)	49280
global_average_pooling1d (GlobalAveragePooling1D)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 6)	774

=====
Total params: 87,366
Trainable params: 87,366
Non-trainable params: 0

CNN

Model: "sequential_4"

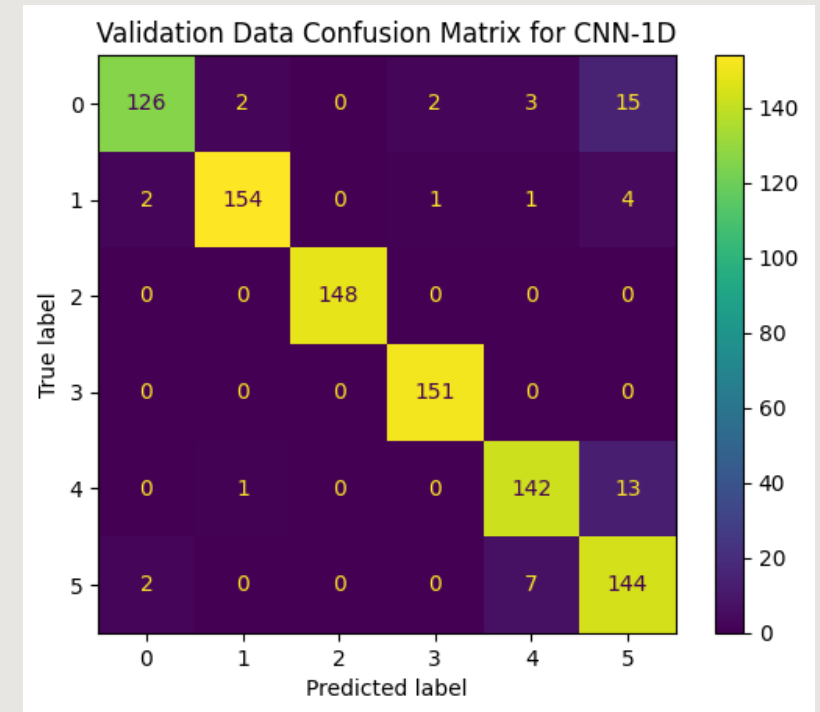
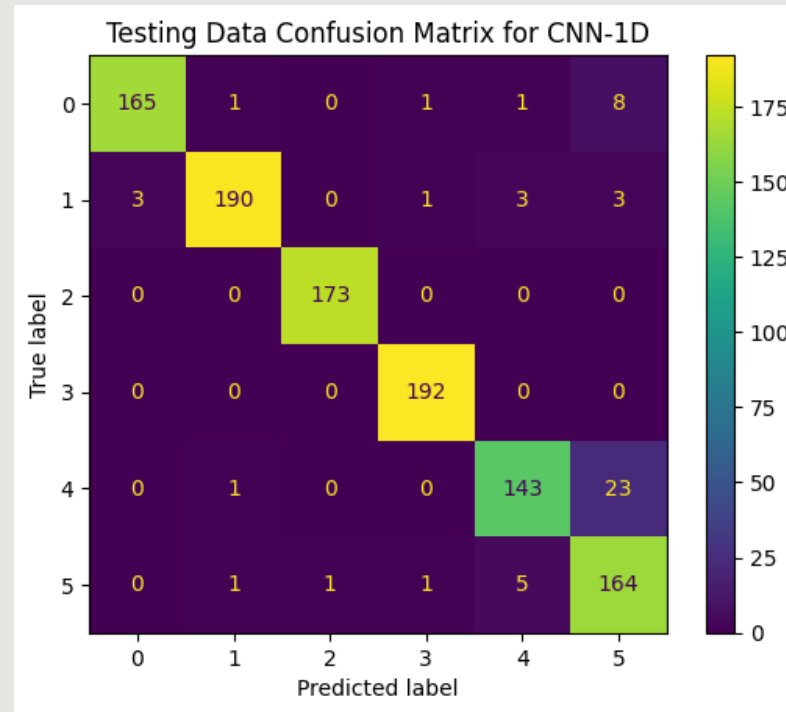
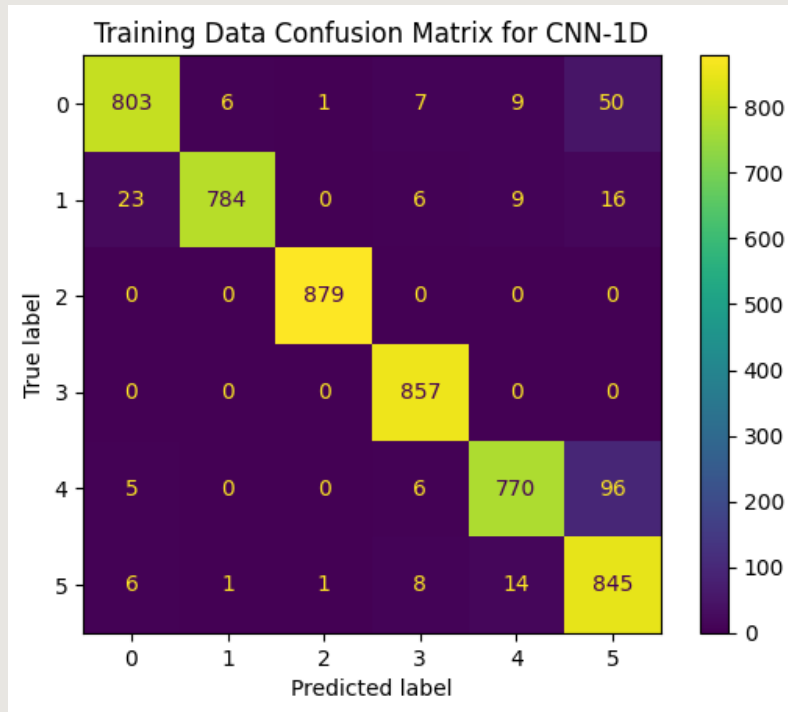
Layer (type)	Output Shape	Param #
conv1d_6 (Conv1D)	(None, 191, 64)	256
conv1d_7 (Conv1D)	(None, 189, 64)	12352
max_pooling1d_2 (MaxPooling1D)	(None, 63, 64)	0
bidirectional (Bidirectional)	(None, 128)	66048
dense_9 (Dense)	(None, 6)	774

=====
Total params: 79,430
Trainable params: 79,430
Non-trainable params: 0

BICRNN

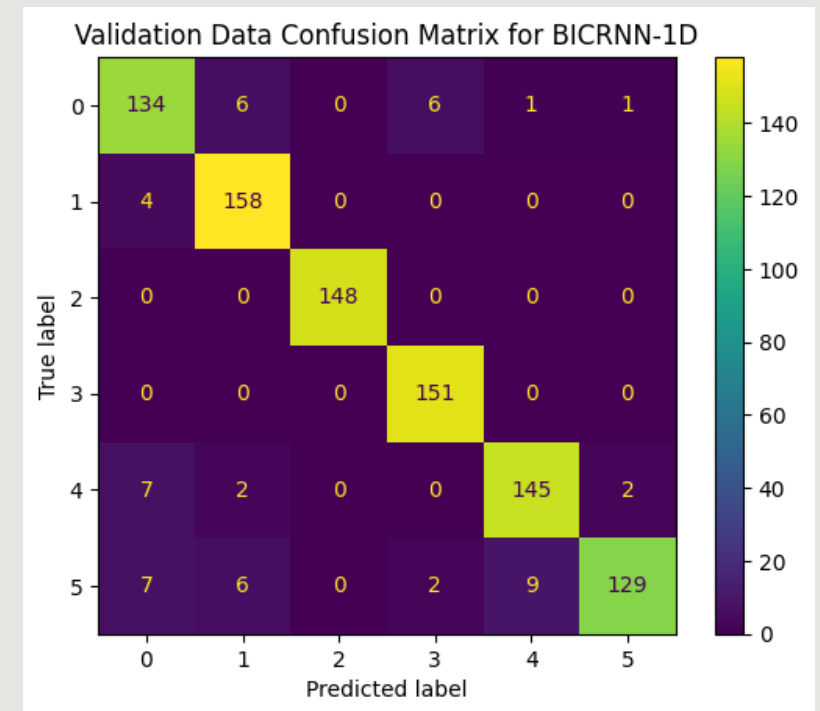
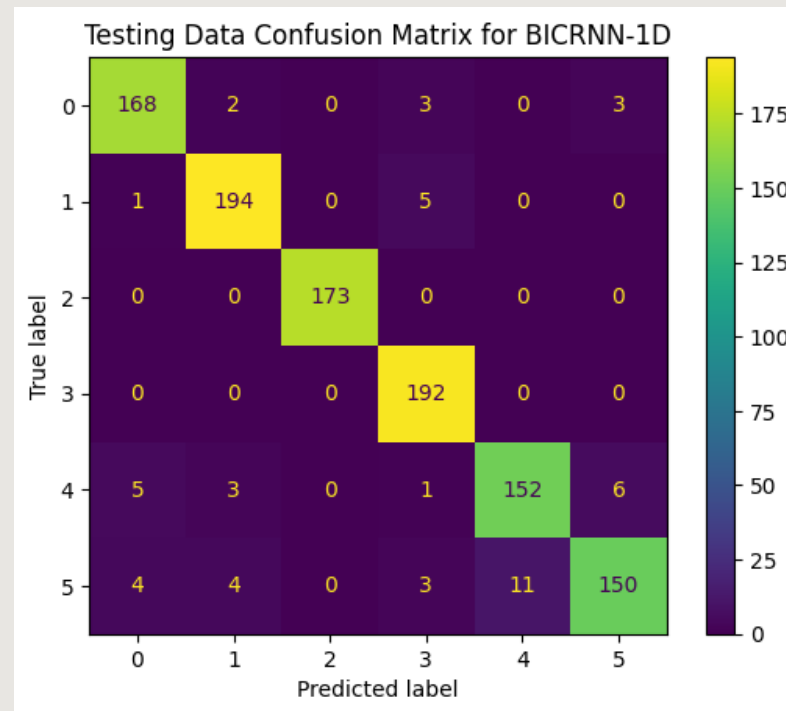
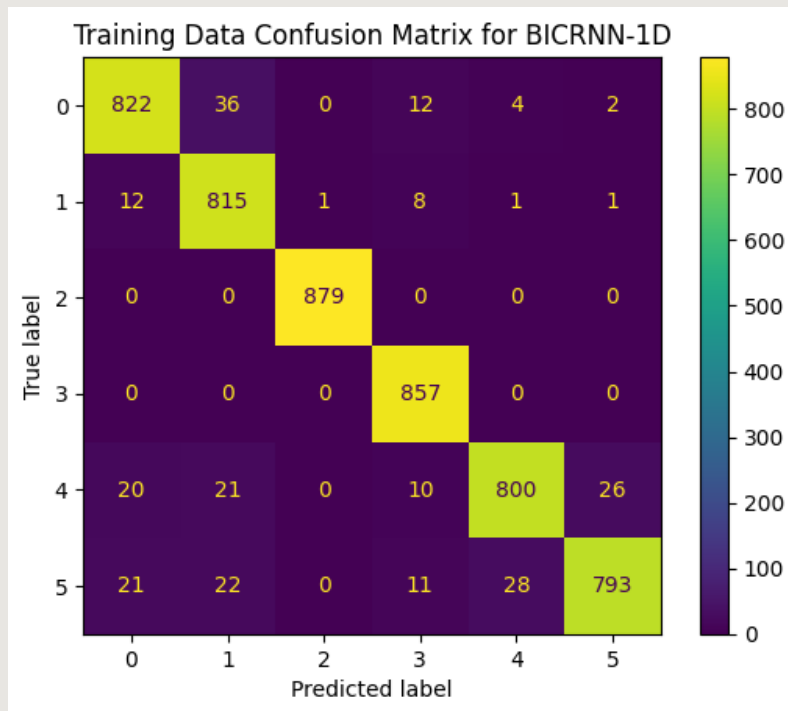
Result Analysis

- The training for each model took approximately ~25 minutes.
- The Confusion matrices of Convolutional Neural network is given below:



Result Analysis (Contd.)

- The Confusion matrices of Bidirectional Convolutional Recurrent Neural network is given below:



Result Analysis (Contd.)

- Accuracy and loss for CNN and BICRNN are determined below:

1. Convolutional Neural Network (CNN):

```
2 - Convolutional Neural Network (CNN 1-D) Metrics for 6 classes is (Axecutting, Chainsaw, Forest, Handsaw, Rain & Thunder, Wind):  
216/216 [=====] - 1s 3ms/step - loss: 0.1680 - accuracy: 0.9509  
Accuracy: 0.9509259462356567  
Loss: 0.16803987324237823
```

2. Bi-Directional Convolutional Recurrent Neural Network (BICRNN):

```
5 - Bi-Directional Convolutional Recurrent Neural Network (BI-CRNN 1-D) Metrics for 6 classes is (Axecutting, Chainsaw, Forest, Handsaw, Rain & Thunder, Wind):  
216/216 [=====] - 1s 7ms/step - loss: 0.1599 - accuracy: 0.9528  
Accuracy: 0.9527778029441833  
Loss: 0.15989729762077332
```

Conclusion

- We presented technological solution to detect tree cutting event through acoustic signal processing.
- We have described and determined the Audio pre-processing techniques which we are going to be used for the development of this system.
- The various Feature extraction techniques are also described by which crucial features will be extracted and used for classifying the data.
- Further, the extracted features are classified using various models and the results are being analysed.
- For better efficiency distance-based algorithms are also being researched upon for implementation.
- Multi-labelled audio classification is also being researched such that the system can be made more accurate.

The slide features a light gray background with decorative elements in the corners. The top-left corner contains a large light blue circle, a small orange circle, a small light blue circle, a small teal circle, and a tiny dark blue circle. The top-right corner features a large green circle, a small orange circle, and a medium blue circle. The bottom-right corner is decorated with a small light green circle, a small teal circle, a medium blue circle, a small pink circle, and a large lime green circle. The text "Thank You" is centered in a dark teal, sans-serif font.

Thank You