AHSANULLAH UNIVERSITY OF SCIENCE & TECHNOLOGY

Course No: CSE 4108

Course Name: Artificial Intelligence Lab

Section: B   Lab Group: B1

Project 1 Submission

Semester: Fall 2020

Submitted to:

Dr. S.M.A Al Mamun
Department of CSE, AUST.

Md. Siam Ansary
Department of CSE, AUST.

Submitted By:

Name: Sadman Jahin

ID: 17.02.04.115

Email: 170204115@aust.edu

Introduction

Audio Sound detection and classification is a major role in AI. Now a days audio data needs to be detected for many purposes. So, my idea is to analyze our environmental audio data, preprocess them, find out which preprocess techniques yield better results.

Dataset Description

Dataset of this AI project is audio wav files. The first Idea to collect the dataset from personal recordings. But after some record it is found the data contains much noises and model became unpredictable.

Datasets containing audio files downloaded from various sources/websites. Some of the file contains noises which were normalized by software and tools. Dataset also contains recorded sounds. This is a mixed dataset collected manually. Audio Dataset.csv contains information about filename, folder, file class.

Classes in The Dataset

1) Bird Chirp 2) Dog Bark 3) Thunderstorm

4) Children Play 5) Car Horn 6) Cricket Chirp

Dataset contains 120 audio Files

Features

If we visualize audio files, they are collection of frequencies. Audio Files can be analyzed by python library librosa. The figure given below is a visualization of an audio file from the dataset 'Horn2.wav'. librosa.load can load audio files and normalize them plot the Time Domain graph. An audio signal is constantly changing, so to simplify things we assume that on short time scales the audio signal doesn't change much (when we say it doesn't change, we mean statistically i.e., statistically stationary, obviously the samples are constantly changing on even short time scales. If the frame is much shorter we don't have enough samples to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame. So, many feature can be extracted from a audio file considering the frequency level and which way they are crossing the x axis and y axis. Now, we will be discussing about a key feature Mel Frequency Cepstral Coefficient.

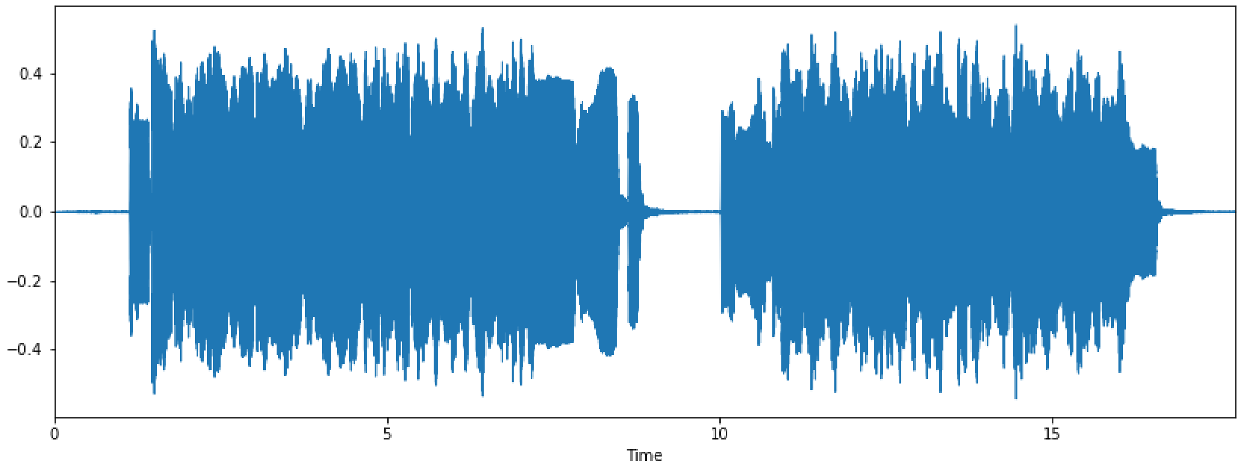
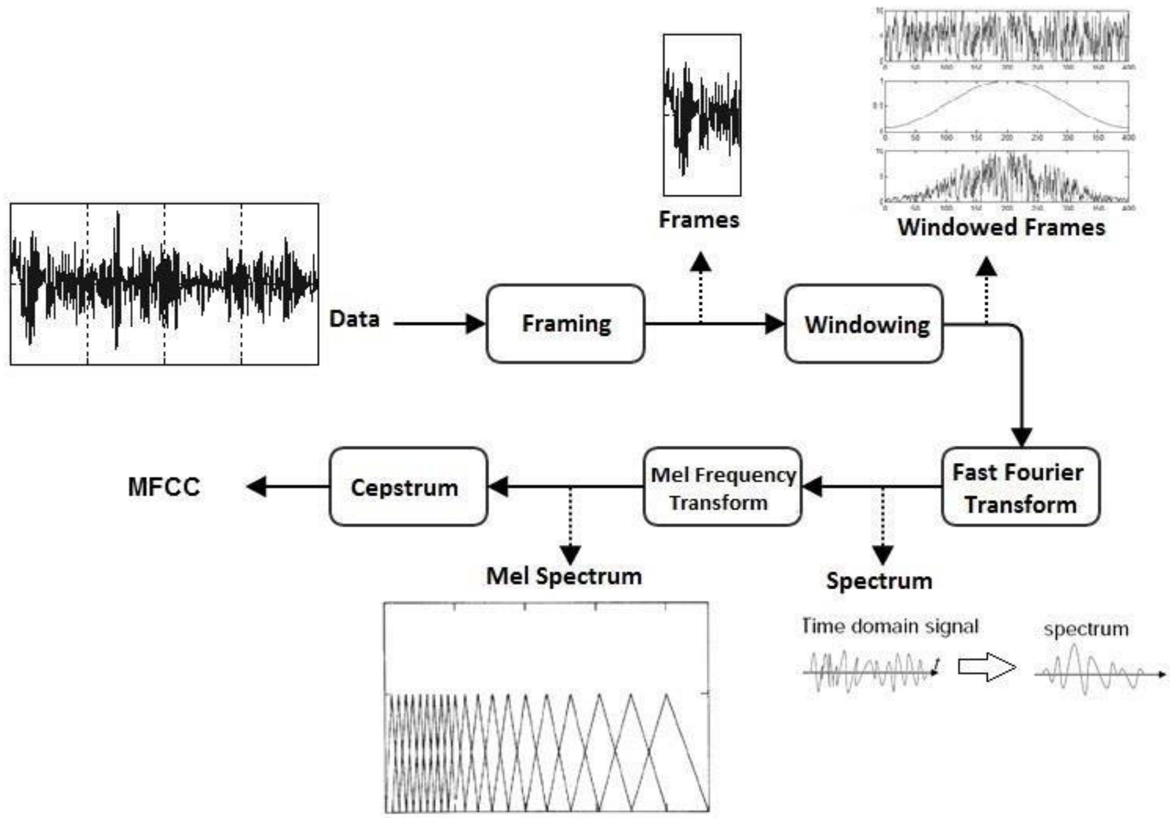
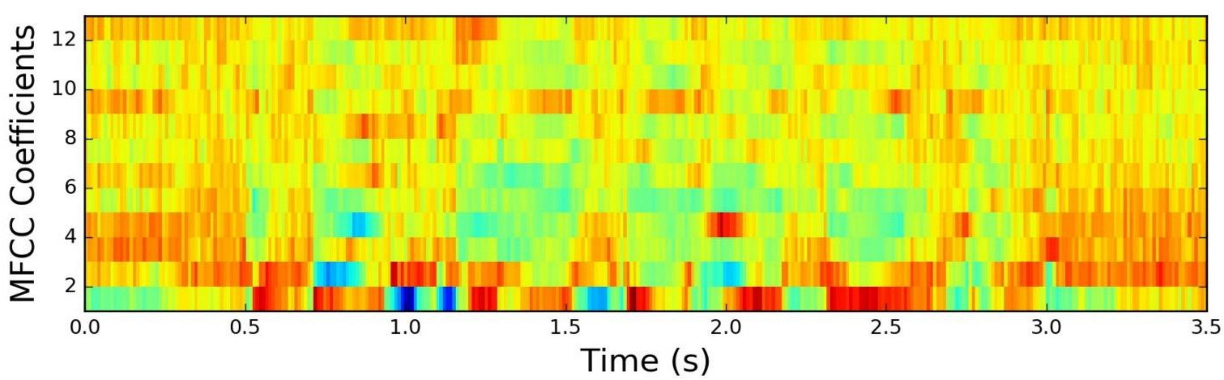


Figure: Time Domain Signal Graph of Horn2.wav

Key Feature: MFCC

MFCC Stands for Mel Frequency Cepstral Coefficient which are the coefficients that collectively make up an MFC. MFC is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency.





MFCC is obtained from framing the graph, First Fourier Transform, Cosine Transform. Python Library librosa has built in functionality to extract feature from audio data.

Feature 2: Mel Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signals spectrograms are sometimes called sonographs, voiceprints, or voicegrams.

A spectrogram can be generated by an optical spectrometer, a bank of band-pass filters, by Fourier transform or by a wavelet transform (in which case it is also known as a scalogram or scalogram.

The Mel scale is a scale of pitches that human hearing generally perceives to be equidistant from each other. As frequency increases, the interval, in hertz, between Mel scale values (or simply Mels) increases. The name Mel derives from melody and indicates that the scale is based on the comparison between pitches.

The Mel spectrogram remaps the values in hertz to the Mel scale.

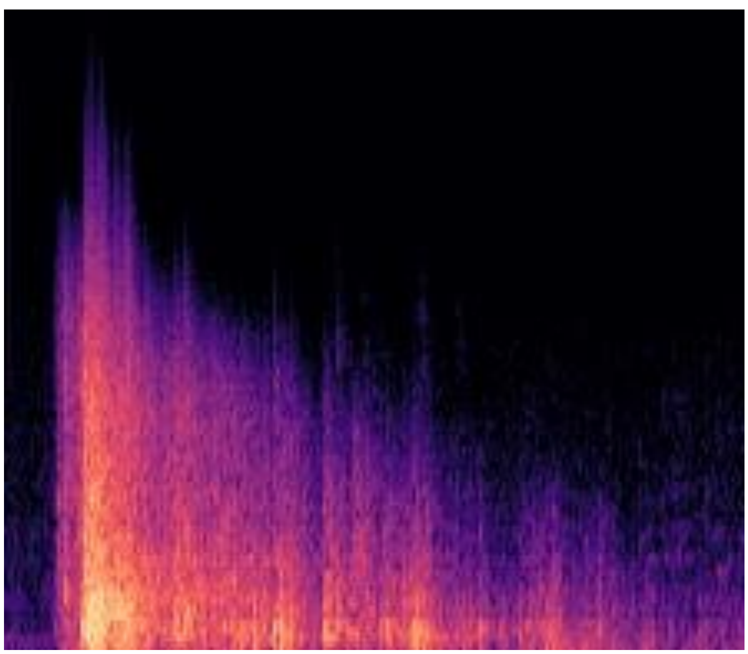


Figure: Mel Spectrogram of Thunderstorm.wav

Description of the Models

1. ***<u>ANN</u>***: An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

2. ***<u>KNN</u>***: **K-Nearest Neighbors** is a machine learning technique and algorithm that can be used for both **regression** and **classification** tasks. K-Nearest Neighbors examines the labels of a chosen number of data points surrounding a target data point, in order to make a prediction about the class that the data point falls into. K-Nearest Neighbors (KNN) is a conceptually simple yet very powerful **algorithm**, and for those reasons, it's one of the most **popular** machine learning algorithms. Let's take a deep dive into the KNN algorithm and see exactly how it works. Having a good understanding of how KNN operates will let you appreciated the best and worst use cases for KNN.

3. ***<u>CNN</u>***: In deep learning, a **Convolutional Neural Network** (CNN, or ConvNet) is a class of artificial neural network, most commonly applied to analyze visual imagery.[1] They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. CNNs are regularized versions of multilayer perceptron. **Multilayer perceptron** usually means fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data.

Result Table

In this project ANN, CNN, KNN model are analyzed and compared.

The result comparison for this models table is given below

Classifiers	Accuracy	Recall	Precision	F1 score
ANN	47%	48%	75 %	47%
KNN	58%	63%	58%	56%
CNN	93%	95%	93%	93%

It is clearly seen that CNN has outperformed other models. It is also needed to be mentioned that the preprocessing with Mel spectrogram has increased the CNN classifier score vastly.

Conclusion

We can clearly see CNN Model trained with MEL-Spectrogram feature works much better than other models. The accuracy obtained is 93%, rest model's accuracy is quite low. Tuning, Normalizing, Reducing Noise from audio data can yield better results. Even so, in other circumstances we would focus more on improving our model. Perhaps we could choose a better algorithm, or maybe we could preprocess our data in finer way. We are looking forward to have better results after selection of good audio data, suitable hyperparameters and best algorithm.