EDA-Audio Classification using machine learning and deep learning

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Shivangi Paliwal (201316)

Ankit Kumar Singh (201197)

Under the guidance & supervision of

Dr. Pardeep Kumar

Associate Professor

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan-173234 (India)

TABLE OF CONTENTS

Candidate’s Declaration

The author hereby declares that the work that was submitted in this report, which was titled "**EDA- audio classification using machine learning and deep learning**" partially satisfies the requirements for the awarding of a Bachelor of Technology degree in Computer Science & Engineering / Information Technology. This work was completed under the supervision of **Dr. Pardeep Kumar**, who is a professor in the Department of Computer Science & Engineering and Information Technology at Jaypee University of Information Technology, Waknaghat. The work was conducted from August 2023 to May 2024.

There has been no application for any other degree or certificate pertaining to the subject matter of the report.

(Student Signature with Date)
Student Name: Shivangi Paliwal
Roll No.: 201316

(Student Signature with Date)
Student Name: Ankit Kumar Singh
Roll No.: 201197

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name – Dr. Pardeep Kumar
Designation: Associate Professor
Department: Computer science & Information Technology
Dated::

CERTIFICATION

This certifies that the work submitted in the project report "**EDA- Audio Classification using Machine Learning deep learning**" towards the partial fulfilment of requirements for the award of a B.Tech in Computer Science and Engineering, and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, is an authentic record of work completed by "**Ankit Kumar Singh(201197) and Shivangi Paliwal(201316)**" between July 2023 and May 2024, under the direction of Dr. Pardeep Kumar with the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Ankit Kumar Singh (201197)

Shivangi Paliwal (201316)

The above statement made is correct to the best of my knowledge.

Dr. Pardeep Kumar
Associate Professor
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

Acknowledgement

First and foremost, I want to express my profound thanks and admiration to the all-powerful God for the heavenly gift that has allowed us to successfully complete the project work.

My sincere appreciation and responsibilities are owed to Dr. Pardeep Kumar, who serves as my supervisor in the Computer Science and Engineering Department at Jaypee University of Information Technology in Waknaghat. My supervisor has extensive expertise and a strong interest in deep learning, which will be invaluable as we carry out this research. We owe the completion of this project to his boundless patience, intellectual direction, encouragement, vigorous supervision, constructive criticism, helpful counsel, reading of several mediocre draughts and corrections at every level, and so on.

In addition, I would like to express my deepest gratitude to everyone who has helped me in any way, whether it be directly or indirectly, in order to ensure the success of our project. Considering the specifics of the case, I would want to express my gratitude to the numerous members of the staff, both teaching and non-teaching, who have provided me with useful assistance and made my pursuit possible.

Lastly, I must politely thank our parents for their ongoing assistance and patience.

Ankit Kumar Singh (201197),

Shivnagi Paliwal (201316)

ABSTRACT

Sound is important for every stage of human life. Voice is an important factor in creating automated systems in many areas, from simple surveillance to personal security. Although there are not many machines on the market today, the benefits they provide raise the question of how to use them in practical situations. Deep learning architecture and learning can be used to create a suitable classification system to solve the problem of low performance of traditional systems. In this study, we aim to use machine learning and deep learning to classify environmental audio's using the spectrograms that produce these sounds. We train a neural network using spectrogram sounds from the surrounding environment. Data Analysis (EDA) and Artificial Neural Networks (ANN). In light of the rapid development of deep learning, this article examines the process of deep learning in the case of music. Speech, music and environmental sound are discussed side by side, pointing out the similarities and differences between these fields, and showing general trends, problems, important information and the potential for cultivation of the fields. Meaning representation (particularly log mel spectra and raw waveforms) and deep learning models are analyzed, including neural networks, changes in short-term memory architectures, and various audio-specific neural network models. Then, speech recognition (automatic speech recognition, musical information retrieval, ambient sound detection, localization and tracking) and the importance of deep learning such as synthesis and application areas will be discussed. transformation (designs for space separation, sound enhancement, synthesis of speech, sound and music). Finally, the main issues and future challenges related to deep learning applied to music processing are identified.

Index Terms—. Deep learning, ANN, EDA, voice recognition.

LIST OF FIGURES

LIST OF TABLES

Chapter 01: INTRODUCTION

1.1 Introduction

Audio classification is a multidisciplinary field that combines data analysis (EDA), machine learning, and deep learning techniques to enable classification of audio data. While EDA helps understand the properties of data, ML and deep learning models provide tools to create accurate and effective classifications.

In the field of audio analysis, the powerful Librosa library combined with the use of Mel Frequency Cepstrum Coefficients (MFCC) represents a significant innovation. Deep learning started to gain popularity due to its real-life applications. ANN achieved good results in image recognition. His work in the visual arts is an example of his strength. Although we have extensive image data, it is not enough to monitor wildlife, especially at night, but this can be done with sound detection equipment. This has encouraged people to use AI for audio distribution and achieved good results. Since neural networks operate on images rather than audio, the first step is to convert audio data into image data. This is done by creating a spectrogram from the audio data. Spectrograms of different types of audio signals show different patterns, and neural networks can learn these patterns. In the early days when we were trying to use artificial neural networks for voice classification, having computational resources and large enough data was an issue, but today we have large voices to support deep learning models. After the spectrogram is created, it is given as input to the ANN model and then the output is produced. Audio consists of many types of features, and it is important to preserve as many as possible during spectrogram generation. This is discussed extensively in . This process involves a lot of handling. In this article, the authors show how to create files in a way that makes the process more efficient, eliminating the need to create clips full-time, without affecting the accuracy of the extraction feature.

1.2 Objectives

- Studying tools and techniques for audio classification estimation and audio behaviour using deep learning.
- Implementing different deep learning techniques to estimate different audio and audio behaviour.
- Calculating the concentration of the audio in witness films and analyse audio behaviour is the goal of different audio analysis.
- Creating a model which estimates different audio classes and audio behaviour using deep learning and machine learning.

If the project is successful in achieving these goals, it will have made a significant contribution to the development and practical application of Audio classification technology, which will have improved human-computer interaction and emotional understanding in technologically driven configurations.

1.3 Significance and Motivation

The significance of an audio classification project lies in its ability to add real-world challenges, improve efficiency in various industries, enhance user experiences, and contribute to technological advancements. The applications are diverse, showcasing the versatility and impact of audio classification in our interconnected and technology-driven world.

Embarking on an audio classification project can be a rewarding and motivating endeavor for reasons such as solving real world problems as Audio classification can be applied to solve real-world problems, such as identifying environmental sounds for monitoring ecosystems, detecting anomalies in industrial machinery, or assisting the visually impaired and also motivate to contribute open source contributions by sharing work as open source can contribute to the community as well, it also motivate in participating in audio classification challenges and competitions offering a chance to test skills against others and gain recognition in this field.

Cultural and Social Analysis: To further understand the emotional environment of countries, academics and policymakers may get insights about cultural and social trends by analysing audio emotions in large datasets.

1.4 Tools & Techniques

Python is an excellent option for Audio Classification projects because to its extensive library, flexibility, and vibrant community of machine learning and natural language processing experts. One way to include Python into a Audio Classification project is as follows:

1. Data Collection and Preprocessing: It is possible to collect audio data and prepare it for analysis using Python. When working with audio data, libraries such as 'pyaudio,''soundfile,' and 'librosa' come in handy for tasks like extracting features.
2. Machine Learning and Deep Learning: When building Audio Classification models, Python's abundance of deep learning and ML libraries comes in handy. Standard libraries for scikit-learn and other traditional machine learning methods are available, in addition to those for deep learning frameworks like TensorFlow and PyTorch.
3. Feature Extraction: Python is a powerful tool for extracting useful information from audio signals, including spectrograms, chroma characteristics, and Mel-frequency cepstral coefficients (MFCCs). For feature extraction, libraries such as 'librosa' are often used.
4. Model Training: Machine learning and deep learning models may be trained in Python using audio attributes as input data. You may build and train a variety of models for the Audio Classificaation problem, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models.

5. Real-time Prediction: Python may be used for real-time audio input capture and processing if your project incorporates real-time emotion identification from live audio streams. Libraries like `pyaudio` can help with this.
6. Visualization: Python offers various libraries for data visualization, which can be helpful for visualizing audio data, model training progress, and results. Popular libraries for visualization include `matplotlib` and `seaborn`.

1.5 Technical Requirements

1. Hardware Requirements:
    a. Processing Power: Depending on the complexity of the machine learning models, a computer with a CPU or GPU capable of handling the training and inference processes efficiently.

    b. Memory: Sufficient RAM is essential for loading and processing large datasets, especially for using deep learning models.

    c. Storage: Storage space for audio datasets, model checkpoints, and other project-related files.
2. Software Requirements:
    a. Python: Most Audio Classification projects are implemented in Python due to its extensive libraries for machine learning, signal processing, and audio analysis.

    b. Development Environment: A Python IDE like Google Colab or Kaggle notebook writing and debugging your code.\

    c. Machine Learning Frameworks: Set up the required libraries for deep learning and machine learning, including Keras, scikit-learn, PyTorch, and TensorFlow.

    d. Audio Processing Libraries: Libraries like librosa or soundfile are often used for audio data preprocessing.

1.6 Deliverables/ Outcomes

The precise objectives and project scope will determine the deliverables and results of a Audio Classification project. However, here are some common deliverables and potential outcomes you can expect from an Audio Classification project:

1. Trained Audio Classification Model: The primary deliverable is a trained machine learning or deep learning model capable of recognizing Audio. This model should be able to take audio input and predict the Audio state conveyed in the speech.
2. Accuracy Metrics: An article or report outlining the Audio Classification model's performance, including details on its accuracy, precision, recall, F1-score, and confusion matrices. This provides an evaluation of how well the model performs in recognizing audio’s.
3. Codebase: The original source code of the project, which may include scripts for data preparation, code for training the model, and, if relevant, code for implementing real-time predictions. Proper documentation and code comments included to facilitate understanding and future development.
4. Demo/Prototype: A working demo or prototype showcasing the functionality of Audio Classification system, especially if it involves real-time Audio recognition from live audio Inputs.

Chapter 02: Literature Survey

Table 1: Literature Survey

S.no.	Paper Title	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results	Limitations
1.	Deep learning in audio classification.	Springer/2022	Computer Vision, NLP, EDA/Urban sound.	82.6%	Most of the systems are not ready to be used in real-world environment.
2.	MFCC Based audio classification on using machine learning	IEEE/2021	SVM,MFCC's Random forest, Decision Tree	88.54%	Fixed Frame length.
3.	Rethinking CNN models for audio Classification	Computer vision/2020	Image net models,Deep CNN models	87.42%	Large training and tuning overhead.
4.	A comparison on Data augmentation methods based on deep learning for audio classification	IOP Publishing/2020	DNN,Log-mel, spectrogram	88%.	Less hidden layers of the considered DNN
5.	Sound Classification Using CNN and tensor deep stacking network	IEEE/2019	CNN, spectrogram, tensor deep stacking	79.7%	Lack of resources.

6.	CNN Architecture for large-Scale Audio Classification	IEEE /2017	CNN,DNN	89.5%	Multi modality is not discussed
7.	Audio Recognition	IEEE/2015	Random forest CNN	88.6%	Large difference in accuracy in different datasets.
8.	CNN sound Classification	IEEE/2013	Linear SVM And CNN		Multilingual Database not used

2.1 Overview of Literature Survey

The articles have seemed to revolve on deep learning-based Audo Classification based on the material supplied. Here's an overview and identification of potential key gaps in the literature:

1. Datasets: Researchers have utilized various datasets, such as database, EMDOB, TIMIT Corpus database, The choice of datasets reflects an emphasis on diverse emotional expressions and acoustic variations in Audio.
2. Deep Learning Architectures: Multiple hidden-layer DNNs, CNN, and SVM combinations are among the deep learning architectures utilized. Multimodal approaches are mentioned, combining Audio with other modalities like visual or textual data.
3. Performance Metrics: Performance metrics, such as accuracy, are highlighted in some papers. There are claims of improved accuracy with the use of certain deep learning architectures.
4. Challenges: Challenges in scaling multimodal systems for large-scale applications are acknowledged. Concerns about efficiency for temporally-varying input data and large training and tuning overhead are mentioned.
5. Specific Techniques: Spectrogram feature extraction utilising methods such as deep retinal convolutional neural networks is explored.

2.2 Key Gaps of Literature Survey

1. Multi-Modality Consideration: While some papers touch upon multimodal approaches, there seems to be a gap in a comprehensive discussion of multi-modality in Audio Classification systems. How different modalities (e.g., audio, text, image) can be effectively integrated for improved Audio Classification could be a subject for further exploration.
2. Scalability Challenges: While challenges related to scalability are mentioned, the specific nature of these challenges and potential solutions are not deeply discussed. Further investigation into how to scale up Audio classification systems for real-world, large-scale applications is a potential gap.
3. Temporal Variability Handling: The efficiency concern for temporally-varying input data is mentioned, but there is no detailed exploration of how different architectures handle temporal variations in Audio. This could be an area for deeper investigation.
4. Real-world Deployment Challenges: The challenge of deploying these systems in real-world scenarios is briefly mentioned. Exploring the practical challenges, ethical considerations, and user acceptance aspects could be an area for further investigation.

Closing these gaps could contribute to a more comprehensive and robust understanding of Audio Classification systems, making them more effective and applicable in real-world scenarios. Researchers may consider addressing these gaps in future work to advance the field.

Chapter 03: Feasibility Study, Requirement Analysis and Design

3.1 Feasibility Study

3.1.1 Problem Definition

Develop an audio classification model using both traditional machine learning (ML) and deep learning techniques to accurately categorize audio recordings into predefined classes, enabling applications in speech recognition, music genre classification, and sound event detection

Identify and handle potential issues like missing or corrupted audio files, inconsistent labelling, or noise.

Preprocess the audio data by applying techniques such as resampling, normalizing, and feature extraction (e.g. Mel-frequency cepstral coefficients - MFCCs) to transform raw audio signals into numerical representations suitable for modelling.

Evaluation Metrics: Creating appropriate measuring tools to evaluate the emotion recognition system's performance; examples include confusion matrices, F1-score, and accuracy

The potential of Audio Clssification for better human-computer interaction and generalised emotion understanding has attracted a lot of interest. To overcome these obstacles and progress the area of audio identification , researchers and engineers are hard at work creating trustworthy systems

3.1.2 Problem Analysis

Sound Classification is one of the most widely used applications in Audio Deep Learning. It involves learning to classify sounds and to predict the category of that sound. This type of problem can be applied to many practical scenarios e.g. 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. Problem analysis for Exploratory Data Analysis (EDA) in the context of Audio Classification using Machine Learning (ML) and Deep Learning (DL) involves understanding the challenges and considerations specific to this domain. As we know we were facing in data understanding phase where we collect data from data sources and labelling ,we also consider data preprocessing where we do feature extraction and normalization .In EDA we distributed classes and also perform data visualization and outlier detection. As its inference speed depending on application and model size especially if deployment is on resource constrained devices.

Establishing a feedback loop for iterative improvement based on the performance of deployed model .

1. Ambiguity and Subjectivity:
    a. Emotions in speech can be ambiguous and subjective, making it difficult to achieve high agreement among human annotators, let alone automated systems.
2. Overfitting and Generalization:
    a. Robust methods for validating and testing models are necessary to avoid frequent problems like overfitting to training data and underperformance on novel, unseen data.
3. Real-Time Processing::
    a. Implementing Audio Classification in real-time applications like virtual assistants or human-computer interfaces demands low-latency solutions.
4. Imbalanced Datasets:
    a. Datasets may have imbalanced audio class distributions, making it challenging to train models that perform well on minority emotions.

5. Transfer Learning:
    a. It may be hard to get models trained on one dataset to work well on another, particularly as there is not a big dataset available for that particular application.

Addressing these challenges and complexities is crucial for advancing the field of Audio classification and making it more accurate, reliable, and applicable to a wide range of practical applications.

3.1.2 Solution

Solving the challenges associated with audio classification requires a multidisciplinary approach involving data collection, feature engineering, model development, and ethical considerations. Here are some potential solutions to address the challenges in Audio Classification.

1. Data Augmentation and Diverse Datasets:
    a. Create larger and more diverse emotion datasets to improve model generalization.
    b. Augment existing datasets by adding noise, altering pitch, or introducing different accents and languages to make the model more robust.
2. Balancing Imbalanced Datasets: Use techniques like oversampling, undersampling, or artificial data production to address class imbalance concerns.
3. Feature Engineering: To capture a larger collection of emotional signals, explore advanced feature extraction approaches including Mel-frequency cepstral coefficients (MFCCs), chroma features, and prosodic features.
4. Solving the challenges in Audio Classification is an ongoing process, and researchers and practitioners continue to explore innovative approaches and solutions to enhance the accuracy, robustness, and practical utility of audio classification systems.

3.2 Requirements

3.2.1 OS

Operating system acts as an interface between software and hardware, in our case Operating system of choice was Microsoft windows 10 , with its easy feature, handy command prompt to run terminal commands and fast and secure development it was an easy choice, also it was the only OS available with us, but it does not changes the fact that even if we had alternatives like linux or mac we would have selected them, by importing system library in python we can easily integrate system file, read and write commands or read system date and time or other features necessary for the development of this prototype for our major project. open() can be utilised to simply open files in the file directory of our computer and os.path command cas be used to change its director. We can also create temporary files for intermediate usage and delete them later once we have obtained final results.

3.2.2 Numpy

The Python package NumPy is used to manipulate arrays. For usage in the discipline of linear algebra, functions, the Fourier transform, and matrix operations are also given. Numpy, an acronym for "Numerical Python," is a package that contains multidimensional array objects and several methods for handling such arrays. For logical and mathematical operations on arrays, NumPy is commonly used. Additionally, it covers various array methods, indexing strategies, etc. You may use it for nothing because it is an open source project. NumPy is the name given to Python used in math.

The goal of NumPy is to offer array objects that are up to fifty times faster than typical Python lists. The NumPy array object is symbolised by the nd array object. Unlike lists, NumPy arrays are stored in a single unified region of memory, making it relatively easy for programs to access and modify them.

3.2.3 Librosa

Python's Python Librosa module is used for audio and music analysis. Librosa is basically used when we work with audio data like in music generation(using LSTM's), Automatic Speech Recognition.

3.2.4 Keras

For creating and analysing efficient and user-friendly deep learning models, Keras is open source and simple to use. We introduce Theano and TensorFlow, two frameworks for fast numerical computation, to design and train neural network models. It uses independent machine learning toolkits as well as C#, Python, and C++ libraries.

Although incredibly powerful tools for building neural networks, Theano and TensorFlow are also difficult to comprehend. TensorFlow or Theanobased deep learning models may be easily and quickly constructed using the fundamental Keras framework. With Keras, deep learning models can be defined quickly. Of course, Keras is the best choice for applications that call for deep learning.

3.2.5 Matplotlib

Library used for visualisation of the data and can further be used for static and interactive visualisations.For 2D displays of arrays, Matplotlib is a fantastic Python visualisation package. A multi-platform data visualisation package called Matplotlib was created to deal with the larger SciPy stack and is based on NumPy arrays.

One of visualisation's biggest advantages is that it gives us visual access to vast volumes of data in forms that are simple to understand. There are numerous plots in Matplotlib, including line, bar, scatter, histogram, etc

3.2.6 MFCC

Deep learning object detection model MFCC has become well-known for its high accuracy and quick inference speed. Without the aid of area proposal networks or anchor boxes, it is a single-stage detector that can predict item bounding boxes and class probabilities from input photos alone. With a lightweight design and few parameters, MFCC is simple to train and deploy on devices with limited resources. It is especially helpful for detecting the presence of individuals in crowds because it can do it rapidly and precisely. Additionally, MFCC supports a number of backbone topologies, including S, M, L, and X, which present various tradeoffs between accuracy and speed.

3.2.7 Scikit-learn

A well-liked machine learning toolkit for Python called Scikit-learn offers a variety of tools for preprocessing data, classifying objects, predicting future outcomes, grouping data, and choosing models. It offers a user-friendly interface and is built on top of other scientific computing libraries like NumPy, SciPy, and matplotlib, making it simple to apply machine learning techniques to practical issues. Support vector machines, decision trees, random forests, k-means clustering, and principal component analysis are just a few of the supervised and unsupervised learning methods that are supported by Scikit-learn. Additionally, it offers performance indicators and cross-validation as model evaluation tools. Generally speaking, scikit-learn is a robust and adaptable Python machine learning package that has grown in popularity among data scientists and machine learning professionals.

3.3 Project Design and architecture

1. In in ERD (1) or Project design (1) firstly see literature review on Audio classification after that we collect our dataset and we do feature extraction using EDA then classify ANN to evaluation of classifier performance and end with conclusion & future work.

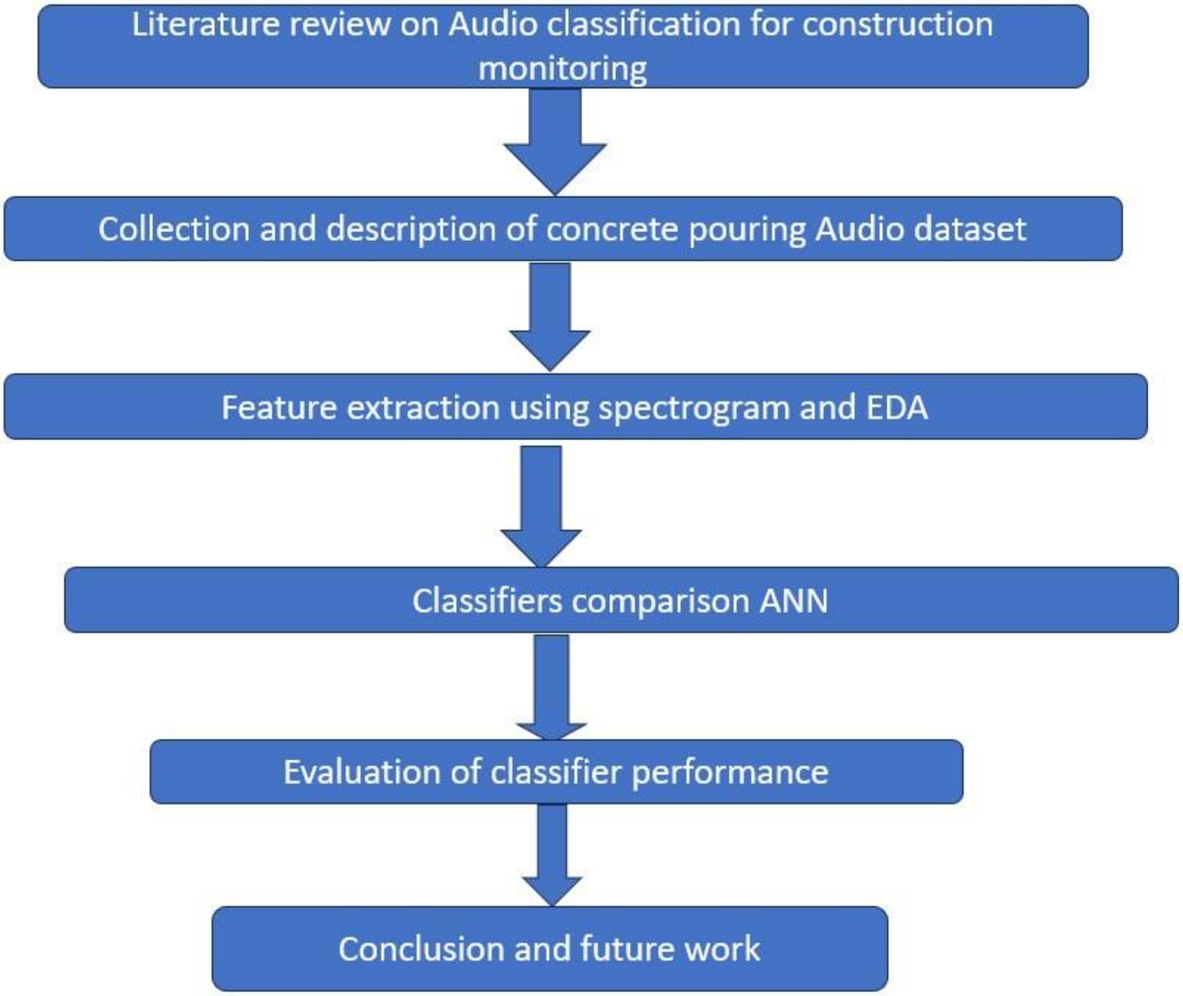

Literature review on Audio classification for construction monitoring
Collection and description of concrete pouring Audio dataset
Feature extraction using spectrogram and EDA
Classifiers comparison ANN
Evaluation of classifier performance
Conclusion and future work

Figure 1a: Project Design 1 /ERD 1

2. In in ERD (2) or Project design (2) So firstly take audio signal and preemphasis it with tensor flow and keras using MFCC to spectrogram the harmonic and percussive component with the Artificial neural network (ANN).

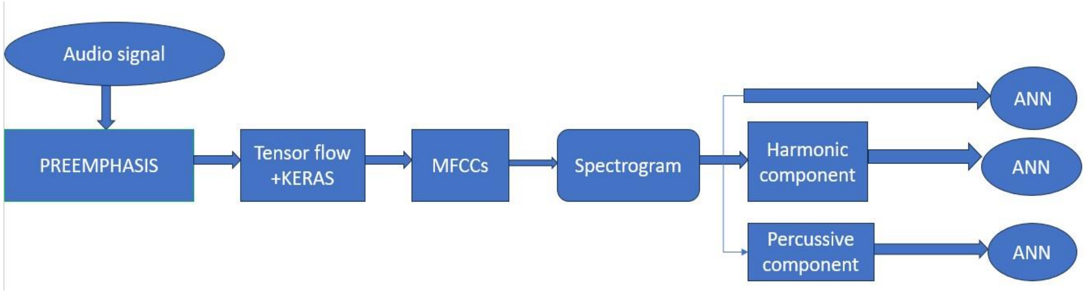


Figure 1b: Project Design 2 /ERD 2

3- In ERD (3) or Project design (3) use spectrogram with ANN with dataset(Animal ,Bird, Insect) on multilevel feature with librosa and EDA to abstract our result or accuracy **.**

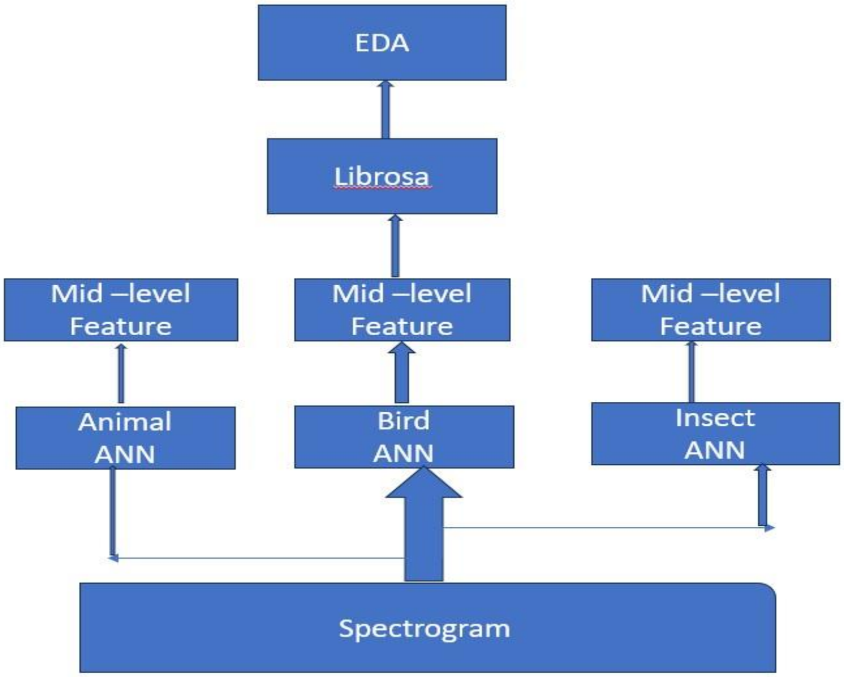


Figure 1c: Project Design 3 /ERD 3

Chapter 04: Testing

4.1 Dataset Used

4.1.1 Urban Sound Dataset

Datasets kept by the Urban Sound Database of Audio classification include:

Dataset Size: The file contains 8732 recorded urban sound excerpts (<= 4 seconds) from 10 categories: air conditioning, car horns, children playing, dogs barking, drilling, interesting idling, gun shot, jackhammers, sirens and street music. These categories are derived from the Urban Sound Taxonomy.

You can find a detailed description of the dataset and how it was compiled in our article. All quotes are taken from the recording published at www.freesound.org. The previously saved files have been divided into ten parts (folders called floor1-fold10) to aid reproduction in the above paragraph and comparison with automatic classification results.

In addition to audio excerpts, a CSV archive containing metadata about each excerpt is also provided. There are 3 ways to extract features from audio files: use the mffc file of the audio file use the spectrogram image of the audio and convert it into a content file (same as image). This can be easily done using Librosa mel spectogram function. Combine these two functions to create a better model. (Reading and extracting files takes a lot of time).I chose to use the second method.

Tag names are converted into taxonomy files for distribution.

CNN was used as an important technique to classify data.

The following are some advantages of using the urban sound dataset:

1. It is a huge and difficult dataset.
2. Numerous different audio scenarios are covered.
3. It is simple to use and well-organised.
4. Many modern audio counting algorithms have been trained and tested using it

4.1.2 UCSD Dataset.

Audio of congested pedestrian pathways make up the UCSD Anomaly Detection Dataset. In 2010, scientists at the University of California, San Diego (UCSD) developed it. 50 audio segments totaling 30 seconds each make up the dataset. A stationary camera set at a height and looking down on pedestrian pathways was used to record the recordings. The walkways can have varying densities of people, from few to many. Only pedestrians are shown in the video at its regular setting. Odd things happen for one of two reasons:

the movement of non-pedestrians (such as bikes, skaters, and small carts, dogs) through the walkways. abnormal pedestrian movement patterns, such as those involving running, jumping, or collisions, human voice. Annotations at the pixel and frame levels make up the dataset's ground truth. Whether an anomaly is present at a certain frame is indicated by the annotations at the frame level. The regions with abnormalities are identified using the pixel-level annotations. Researchers studying anomaly identification in crowded environments can benefit greatly from the UCSD Anomaly identification Dataset. One of the first datasets to offer frame- and pixel-level annotations for anomaly identification in congested environments, it offers both types of information. A number of cutting-edge anomaly detection algorithms have been trained and evaluated using the dataset.

The following are some advantages of utilising the UCSD Anomaly Detection Dataset:

- It is a huge and difficult dataset.
- Numerous different audio scenarios are covered.
- It is simple to use and well-organised.
- Several cutting-edge anomaly detection algorithms have been trained and tested using it.

4.1.3 Examining Dataset

A Data Quality Assessment, a distinct phase in the data collection process, is used to confirm the origin, amount, and effect of any data items that do not adhere to accepted data quality standards. quality of data lifecycle. To assure the quality of the data, the

Data Quality Assessment can be performed once after a certain amount of time as part of an ongoing project or only once.

The quality of your data might rapidly decrease with time, even with stringent data collection protocols that clean the data as it enters your database. Due to factors like people moving homes, changing phone numbers, and passing away, the information you have might soon become outdated.

Finding records that have become erroneous, as well as the source of the data and any possible implications that inaccuracy may have had, is made easier with the use of a data quality evaluation. With the help of this evaluation, it may be rectified and potential new issues discovered.

4.1.4 Take US

Database is a collection of more than 240 audio footage of voice violence, with each video of dimension 224 X 224. Dataset is designed to train models which classify crowd's behaviour as violent and non-violent. Violent Flow Database include videos of length ranging from 1 second to seconds, while average duration of a video in this dataset is 3.6 seconds.

4.2 Flow graph of the Major Project Problem

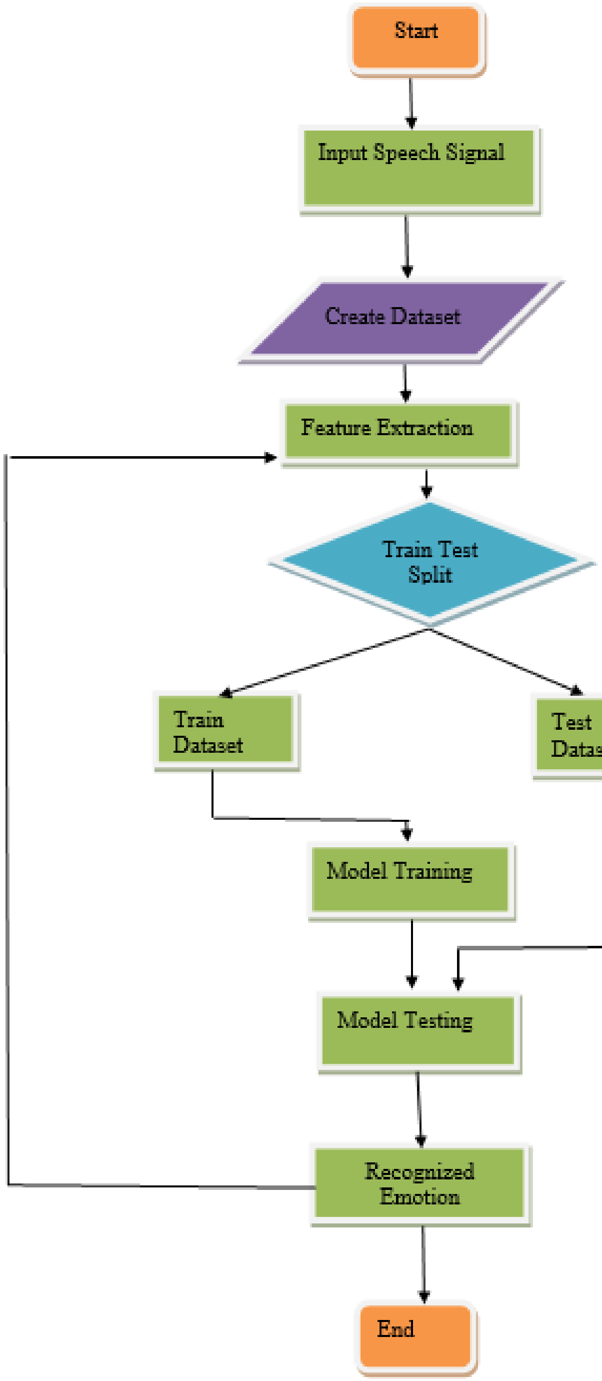


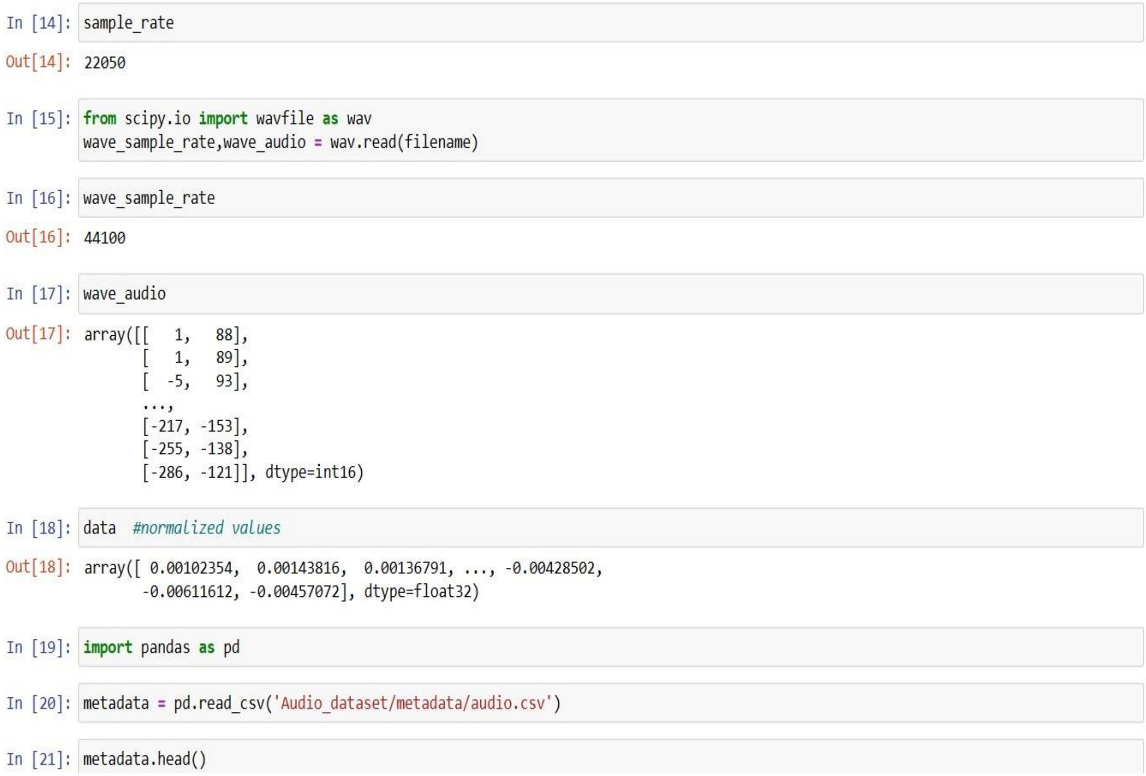
Figure 2 : Flow Graph of Major Project Problem

The flow graph of the Major Project Problem as described in Figure 2 lays down the framework of processing input, extracting features, training the model and finally evaluating its performance. Detailed description of entire project implementation with different stages of implementation is given in the subsequent section.

4.3 Screen shots of the various stages of the Project

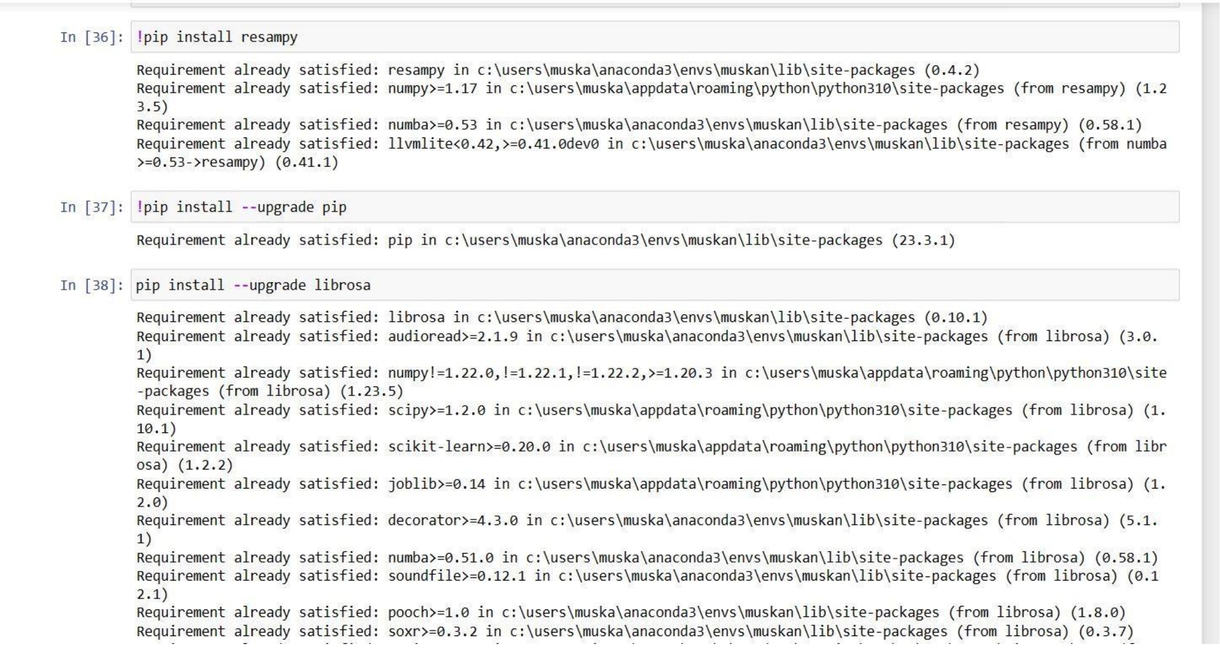
Step 1: Importing necessary libraries and data collection

The necessary libraries are imported and then the dataset is downloaded from the Kaggle[6].



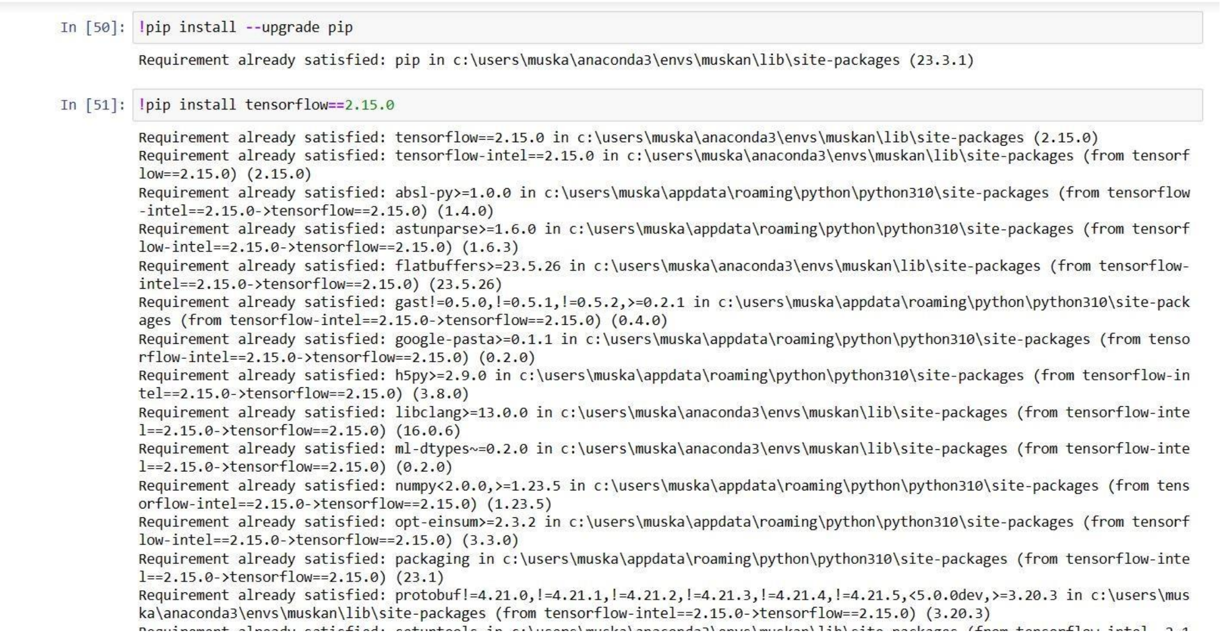
```
In [14]: sample_rate

Out[14]: 22050

In [15]: from scipy.io import wavfile as wav
         wave_sample_rate,wave_audio = wav.read(filename)

In [16]: wave_sample_rate

Out[16]: 44100

In [17]: wave_audio

Out[17]: array([[   1,   88],
                [   1,   89],
                [  -5,   93],
                ...,
                [-217, -153],
                [-255, -138],
                [-286, -121]], dtype=int16)

In [18]: data  #normalized values

Out[18]: array([ 0.00102354,  0.00143816,  0.00136791, ..., -0.00428502,
                -0.00611612, -0.00457072], dtype=float32)

In [19]: import pandas as pd

In [20]: metadata = pd.read_csv('Audio_dataset/metadata/audio.csv')

In [21]: metadata.head()
```

Code Snippet 1: In this code we implemented sample rate and in code datasets in this part. We also implemented wave sample rate and wave audio part.

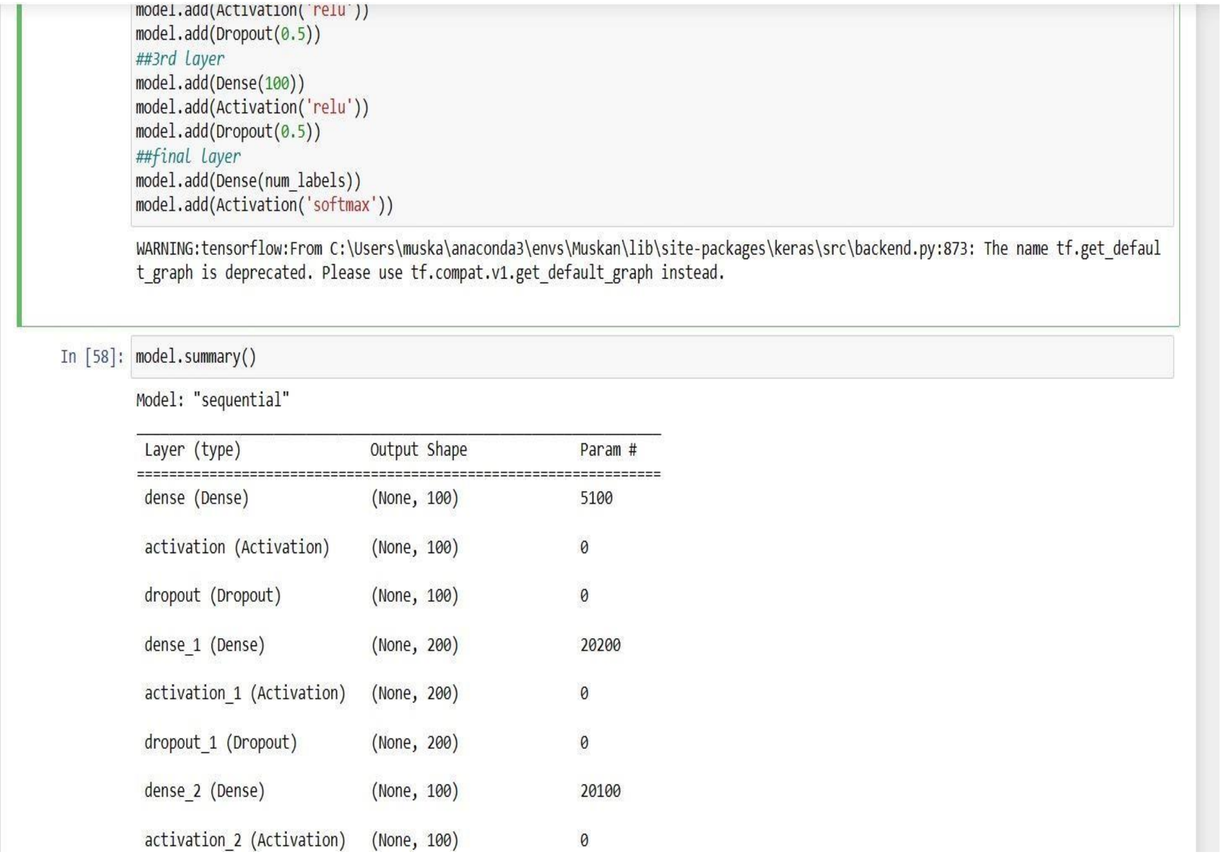


```
In [36]: !pip install resampy

         Requirement already satisfied: resampy in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (0.4.2)
         Requirement already satisfied: numpy>=1.17 in c:\users\muska\appdata\roaming\python\python310\site-packages (from resampy) (1.2
         3.5)
         Requirement already satisfied: numba>=0.53 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from resampy) (0.58.1)
         Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from numba
         >=0.53->resampy) (0.41.1)

In [37]: !pip install --upgrade pip

         Requirement already satisfied: pip in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (23.3.1)

In [38]: pip install --upgrade librosa

         Requirement already satisfied: librosa in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (0.10.1)
         Requirement already satisfied: audioread>=2.1.9 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (3.0.
         1)
         Requirement already satisfied: numpy!=1.22.0,!=1.22.1,!=1.22.2,>=1.20.3 in c:\users\muska\appdata\roaming\python\python310\site
         -packages (from librosa) (1.23.5)
         Requirement already satisfied: scipy>=1.2.0 in c:\users\muska\appdata\roaming\python\python310\site-packages (from librosa) (1.
         10.1)
         Requirement already satisfied: scikit-learn>=0.20.0 in c:\users\muska\appdata\roaming\python\python310\site-packages (from libr
         osa) (1.2.2)
         Requirement already satisfied: joblib>=0.14 in c:\users\muska\appdata\roaming\python\python310\site-packages (from librosa) (1.
         2.0)
         Requirement already satisfied: decorator>=4.3.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (5.1.
         1)
         Requirement already satisfied: numba>=0.51.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (0.58.1)
         Requirement already satisfied: soundfile>=0.12.1 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (0.1
         2.1)
         Requirement already satisfied: pooch>=1.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (1.8.0)
         Requirement already satisfied: soxr>=0.3.2 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from librosa) (0.3.7)
```

Code Snippet 2 :In this part install librosa and resampy part for our code performance.



```
In [50]: !pip install --upgrade pip

         Requirement already satisfied: pip in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (23.3.1)

In [51]: !pip install tensorflow==2.15.0

         Requirement already satisfied: tensorflow==2.15.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (2.15.0)
         Requirement already satisfied: tensorflow-intel==2.15.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from tensorf
         low==2.15.0) (2.15.0)
         Requirement already satisfied: absl-py>=1.0.0 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tensorflow
         -intel==2.15.0->tensorflow==2.15.0) (1.4.0)
         Requirement already satisfied: astunparse>=1.6.0 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tensorf
         low-intel==2.15.0->tensorflow==2.15.0) (1.6.3)
         Requirement already satisfied: flatbuffers>=23.5.26 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from tensorflow-
         intel==2.15.0->tensorflow==2.15.0) (23.5.26)
         Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\muska\appdata\roaming\python\python310\site-pack
         ages (from tensorflow-intel==2.15.0->tensorflow==2.15.0) (0.4.0)
         Requirement already satisfied: google-pasta>=0.1.1 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tenso
         rflow-intel==2.15.0->tensorflow==2.15.0) (0.2.0)
         Requirement already satisfied: h5py>=2.9.0 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tensorflow-in
         tel==2.15.0->tensorflow==2.15.0) (3.8.0)
         Requirement already satisfied: libclang>=13.0.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from tensorflow-inte
         l==2.15.0->tensorflow==2.15.0) (16.0.6)
         Requirement already satisfied: ml-dtypes~=0.2.0 in c:\users\muska\anaconda3\envs\muskan\lib\site-packages (from tensorflow-inte
         l==2.15.0->tensorflow==2.15.0) (0.2.0)
         Requirement already satisfied: numpy<2.0.0,>=1.23.5 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tens
         orflow-intel==2.15.0->tensorflow==2.15.0) (1.23.5)
         Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\muska\appdata\roaming\python\python310\site-packages (from tensorf
         low-intel==2.15.0->tensorflow==2.15.0) (3.3.0)
         Requirement already satisfied: packaging in c:\users\muska\appdata\roaming\python\python310\site-packages (from tensorflow-inte
         l==2.15.0->tensorflow==2.15.0) (23.1)
         Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in c:\users\mus
         ka\anaconda3\envs\muskan\lib\site-packages (from tensorflow-intel==2.15.0->tensorflow==2.15.0) (3.20.3)
```



```
model.add(Activation('relu'))
model.add(Dropout(0.5))
##3rd layer
model.add(Dense(100))
model.add(Activation('relu'))
model.add(Dropout(0.5))
##final layer
model.add(Dense(num_labels))
model.add(Activation('softmax'))
```

```
WARNING:tensorflow:From C:\Users\muska\anaconda3\envs\Muskan\lib\site-packages\keras\src\backend.py:873: The name tf.get_defaul
t_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

```
In [58]: model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 100)               5100

 activation (Activation)     (None, 100)               0

 dropout (Dropout)           (None, 100)               0

 dense_1 (Dense)             (None, 200)               20200

 activation_1 (Activation)   (None, 200)               0

 dropout_1 (Dropout)         (None, 200)               0

 dense_2 (Dense)             (None, 100)               20100

 activation_2 (Activation)   (None, 100)               0
```

Code Snippet 3: In this part implemented a RELU and softmax to accurately run code part .After a model part was run to see a code implementing

Step 2: Data Augmentation

In data augmentation, we generate additional polymerized data samples by inserting tiny disturbances into our original training set. Audio polymerization may be achieved by the use of noise injection, time shifting, pitch and speed manipulation, and other similar techniques. The Figure 4 given below shows and compares Mel spectrogram of clean speech as well as augmented noise speech based on MFCC coefficients. Our goal is to improve our model's generalizability and make it resistant to such disturbances. The original training sample's label must be preserved when adding the disturbance for this to function.

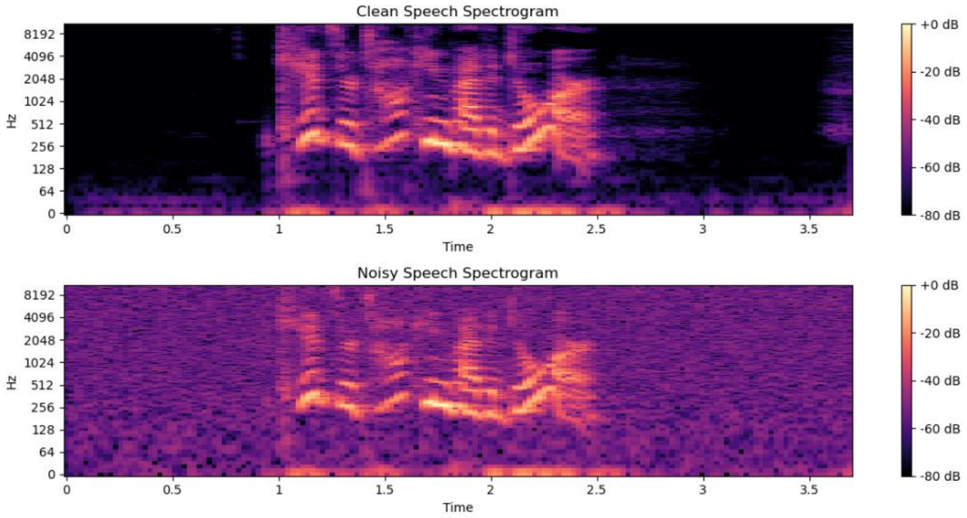


Figure 3: Mel Spectogram of Original and noisy signal

Step 3: Feature Extraction

Audio Classification sometimes makes use of feature extraction using Mel-frequency cepstral coefficients (MFCC) over multiple time frames as represented by graph in Figure 5. The feature vectors that represent the speech signal are the MFCC coefficients that are produced and are stored in csv file.

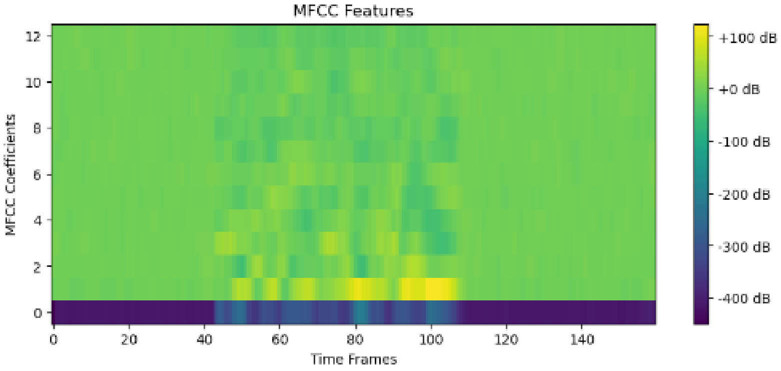


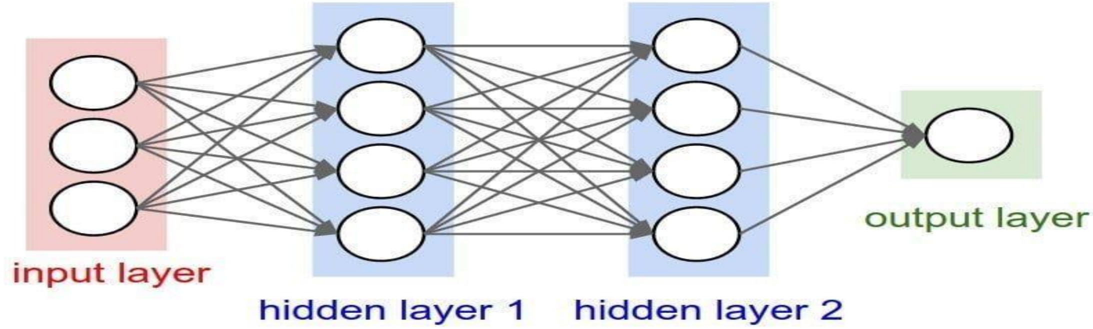
Figure 4: MFCC Feature Extraction

4.4 Models Utilised

4.4.1 ANN(Artificial neural network)

An artificial neural network known as ANN excels at identifying and interpreting patterns. ANN has so far proven to be the most useful for categorising images.

A ANN model may employ various numbers and sizes of filters. The key element that enables us to find the pattern is these filters. Artificial neural networks, or ANNs for short, are specialised neural network models which are created for use with two dimensional visual input, however they may also be used for one-dimensional and three-dimensional data. The hidden layer, which gives the network its name, is a key part of a artificial neural network. The most often utilised layers in ANN models are inner and outer layers. ANN is a useful solution for problems involving photo categorization since it performs better with data that are represented as grid structures.



. Figure 5: Basic ANN architecture

4.4.2 Functioning of ANN

Artificial neural networks are made up of many layers of artificial neurons. Artificial neurons are mathematical operations that compute the weighted sum of a collection of inputs and output an activation value, much like their biological counterparts.

The behaviour of each neuron is governed by its weight. When given pixel values, ANN's artificial neurons can recognise a variety of visual properties.

Each layer in a produces a number of activation maps when you feed it an image. Activation maps draw attention to the image's most important components. Each neuron receives the colour values of a pixel patch as input and multiplies them by its weights before adding them all up and sending the sum via the activation function

4.4.3 Inception V3

This model was developed as a combination of several ideas and has proven to be more than 78% accurate on the ImageNet dataset.

Over time, researchers have evolved. According to Figure below, this model contains fewer than 25 million parameters. Convolutions, average pooling, max pooling, concatenations, dropouts, and entirely linked layers are the structure that make up a model itself. The model heavily employs batch normalisation, which is also applied to the inputs for activation. Softmax is used to compute the loss. We imported this pretrained model for the project using Keras and transfer learning.

Additionally, you chose the imagnet weights, provided a unique 299 x 299 input shape for the model, and changed the include top value to false. A dropout layer, a flatten layer, and a thick layer were the next layers we added using sequential.

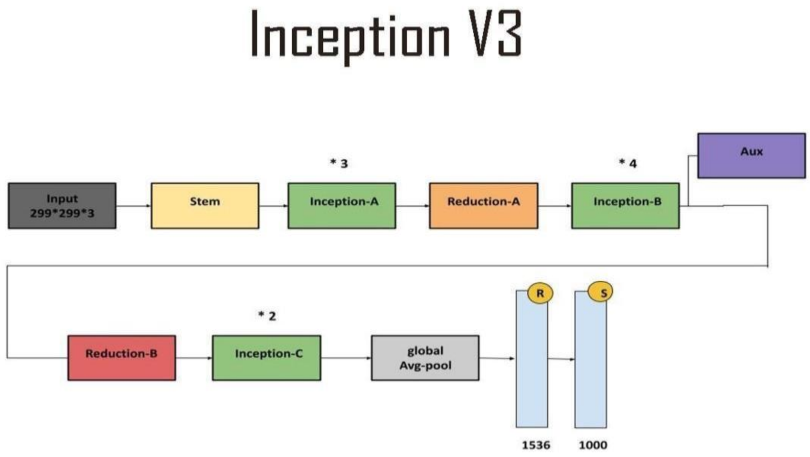


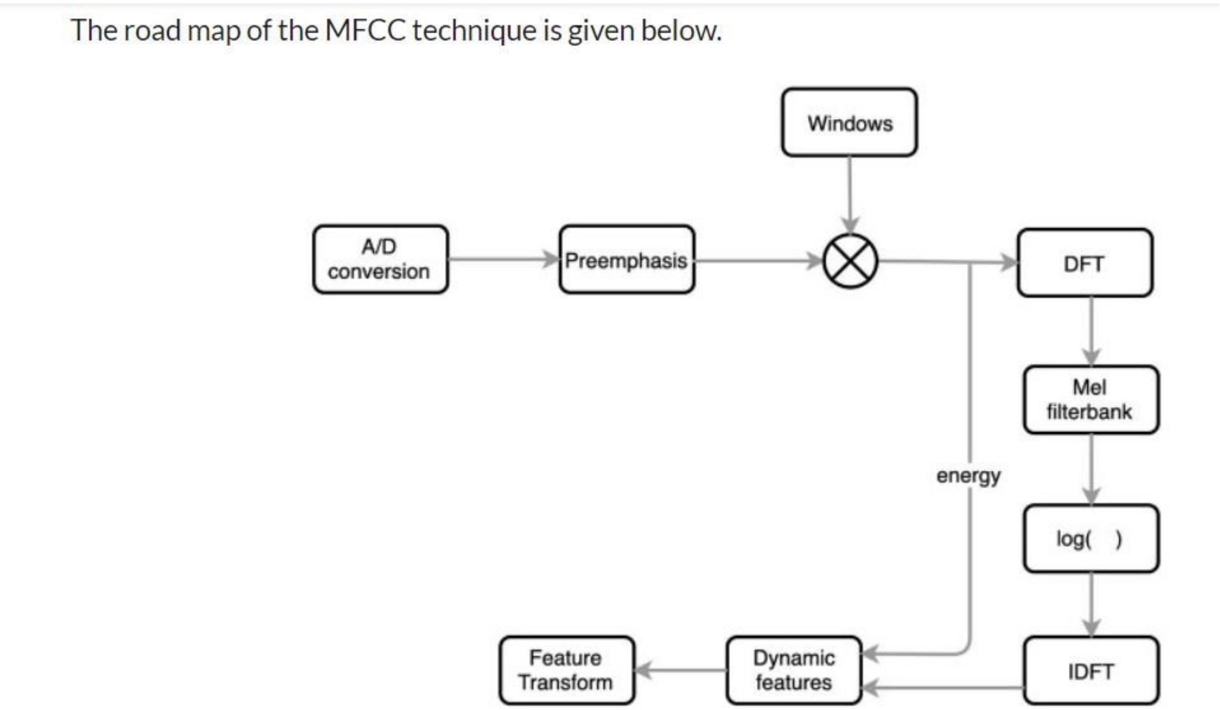
Figure 6: Inception V3

4.4.4 MFCC

MFCC is a widely used technology in speech and audio processing. MFCCs are used to represent the properties of sounds in a way suitable for many machine learning applications, such as speech recognition and music analysis.

Simply put, MFCC is a set of coefficients that capture the shape of the power spectrum of an audio signal. These are achieved by first converting sound to frequency using techniques such as Discrete Fourier Transform (DFT) and then using the Mel scale to estimate human sound hearing. Finally, cepstrum coefficients are calculated based on the Mel scale spectrum.

MFCCs are particularly useful because they highlight features of the audio signal that are important for human perception while providing less relevant information. This makes them useful for tasks such as speaker identification, emotion recognition, and converting speech to text.



The road map of the MFCC technique is given below.

Figure 7: MFCC

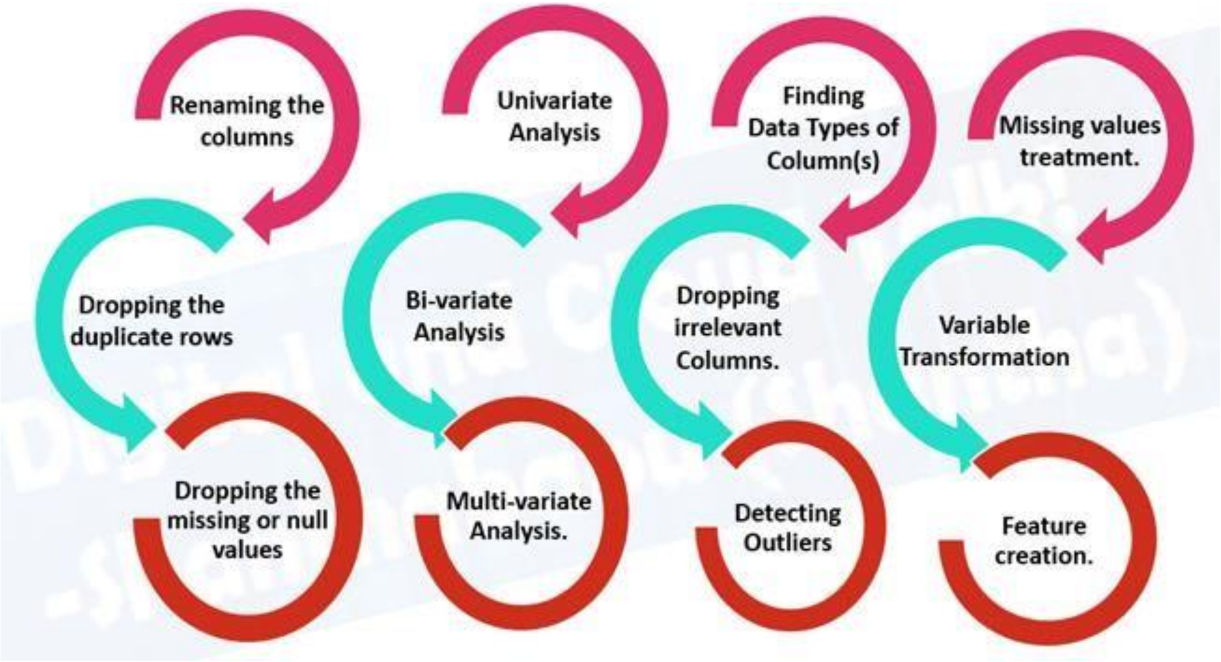
4.4.5 EDA

Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.

The Foremost Goals of EDA :

1. Data Cleaning: EDA involves examining the information for errors, lacking values, and inconsistencies. It includes techniques including records imputation, managing missing statistics, and figuring out and getting rid of outliers.

2. Descriptive Statistics: EDA utilizes precise records to recognize the important tendency, variability, and distribution of variables. Measurements such as mean, median, mode, standard deviation, range and percentage are commonly used.



Renaming the columns
Dropping the duplicate rows
Dropping the missing or null values
Univariate Analysis
Bi-variate Analysis
Multi-variate Analysis.
Finding Data Types of Column(s)
Dropping irrelevant Columns.
Detecting Outliers
Missing values treatment.
Variable Transformation
Feature creation.

Figure 8: EDA-Architrcture

4.4.6 Librosa

A Librosa is a useful audio and audio research library that helps programmers create applications that use Python to manage audio and music data rendering. This Python package for music and audio testing is used when we process audio files without automatic speech recognition (using Lstm), as in the music era.

This library is easy to use and can solve simple problems as well as advanced tasks related to music and music. It is open source and can be accessed without restrictions under the ISC license.

The library supports some concepts related to the processing and extraction of audio recordings, such as charge in the circuit, recording of different spectrogram depictions, symphonic percussion track separation, conventional spectrogram attenuation, superimposed and translated audio, spatio-temporal audio processing, etc. , continuous presentation, compound consonant percussion segmentation, hold synchronization, etc.

Librosa helps identify the audio signal and add the extracted object using different symbols.

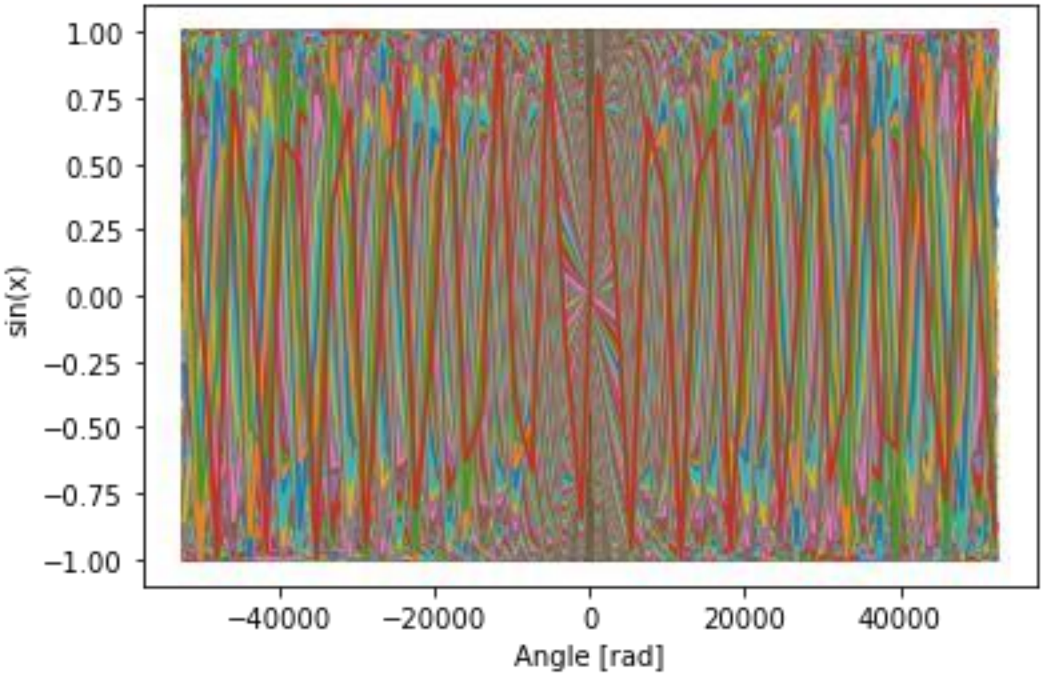


Figure 9: Librosa Architecture

4.4.7 SVM

SVM works best when the dataset is small and complex. It is usually recommended to use logistic regression first and see how it works, if it does not give good accuracy you can use SVM without kernels (kernels will be discussed in detail in the next section). Logistic regression and kernelless SVM have similar performance, but depending on your specifications, one may outperform the other. Support vectors: These are the points closest to the plane. A dividing line will be defined with the help of data points. Distance: It is the distance between the hyperplane and the closest observation (support vector) to the hyperplane. In SVM, larger margins are considered positive. There are two types of margins: hard margins and soft margins

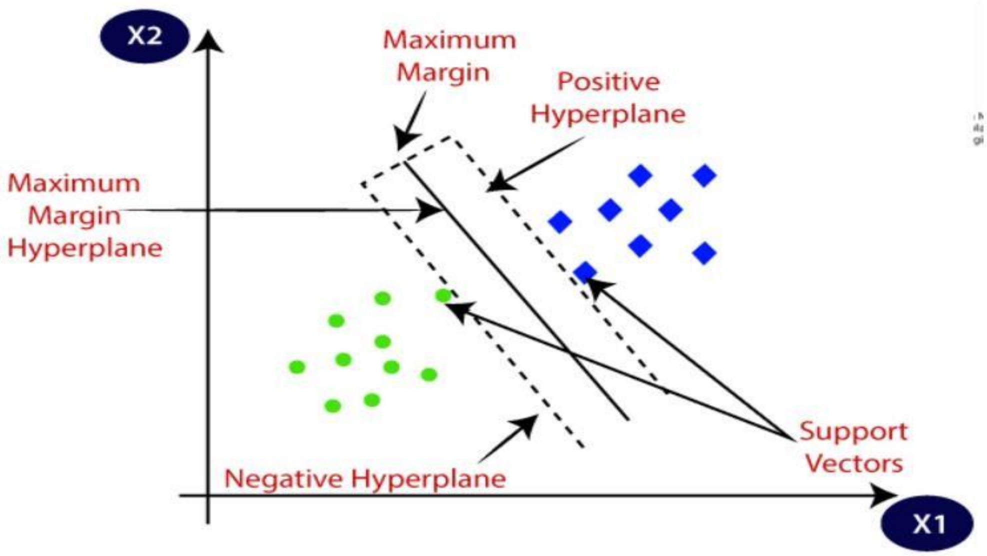


Figure 10: SVM Architecture

4.4.8 ANN Architecture

The neural network has many layers, each layer has a specific function, and as the complexity of the model increases, the number of layers also increases, so it is called a multilayer perceptron. The purest form of a neural network has three layers: input layer, hidden layer and output layer. The input process collects the input signal and sends it to the next layer and finally the output process gives the final prediction. These neural networks need to be trained using some data training and machine learning algorithms before given specific problems.

ANN Layers :

Since these layers recur often, our neural networks are deep, so, these types of structures are known as deep neural networks.

a. Input given are raw pixel values.
b. Hidden layer: The input layers transform the results given by the neuron layer as output.
c. Output layer: The maximum score generated of the input digits may be found after concentrating on the scoring class.

As we delve further and deeper into the levels, there is a notable increase in complexity. The trip would be valuable, though, because while precision could advance, time consumption, regrettably, does as well.

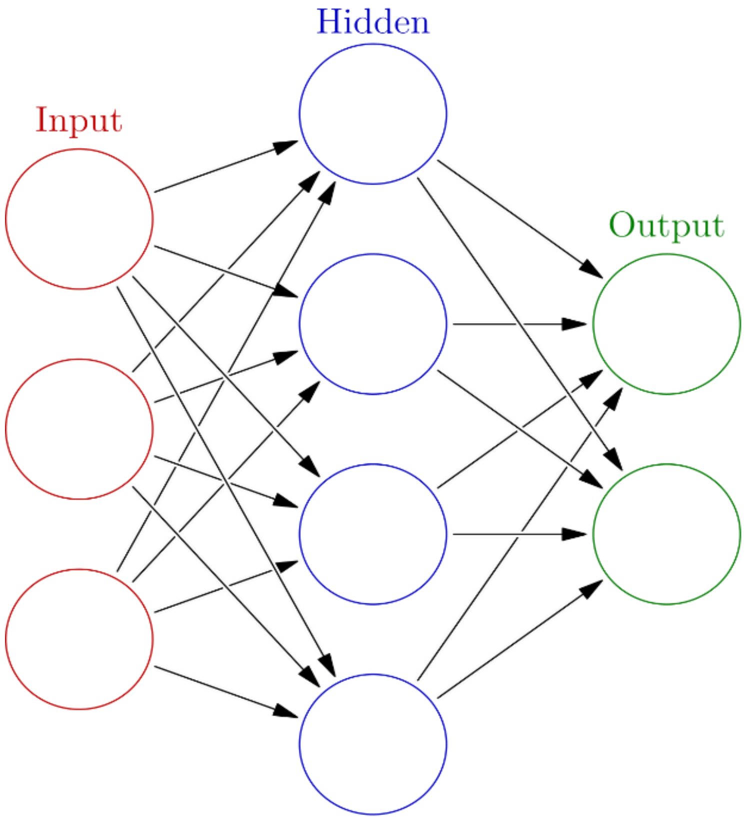


Figure 11: ANN Layers

1. INPUT Layer : It accepts input from programmers in many different formats.
2. Hidden Layer : The hidden layer is between the input layer and the output layer. It does all the calculations to find hidden features and patterns.

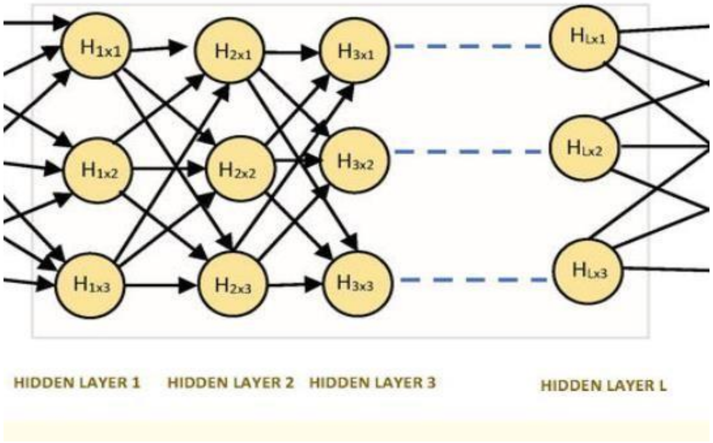


Figure 12: Hidden Layers

3. Output Layers: The Inputs are subjected to multiple transformations using a secret algorithm and ultimately an output is created using this algorithm. Artificial neural networks take inputs and calculate the weighted sum of the biased inputs. This calculation is expressed in the form of a transfer function.

Working of ANN

Artificial neural networks can best be represented as weighted maps in which neurons form nodes. The relationship between the output neuron and the input neuron can be viewed as an edge indicator with weight. Artificial neural networks receive input signals in the form of patterns and images in the form of vectors from other sources. Then, for each n inputs, these inputs are numbered with the symbol x(n). Each input is then divided into its weights (these weights are the elements the neural network uses to solve a particular problem).

Generally speaking, these weights generally represent the strength of connections between neurons in the neural network. All weight entries are collected in the calculation unit. If the weight number is equal to zero, a bias is added to make the output non-zero or make the power work. Bias has the same concept and its weight is equal to 1. The values of the weighted objects here can vary from 0 to positive infinity. Here, some maximum value is taken as a standard to keep the response within the desired range and the number of weighted items is transmitted to the activation function. Functionality refers to the process of change used to achieve the desired outcome. There are many types of dynamic functions, but most of them are linear or nonlinear functions. Some of the most commonly used functions include binary, linear, and Tan hyperbolic sigmoid activation functions.

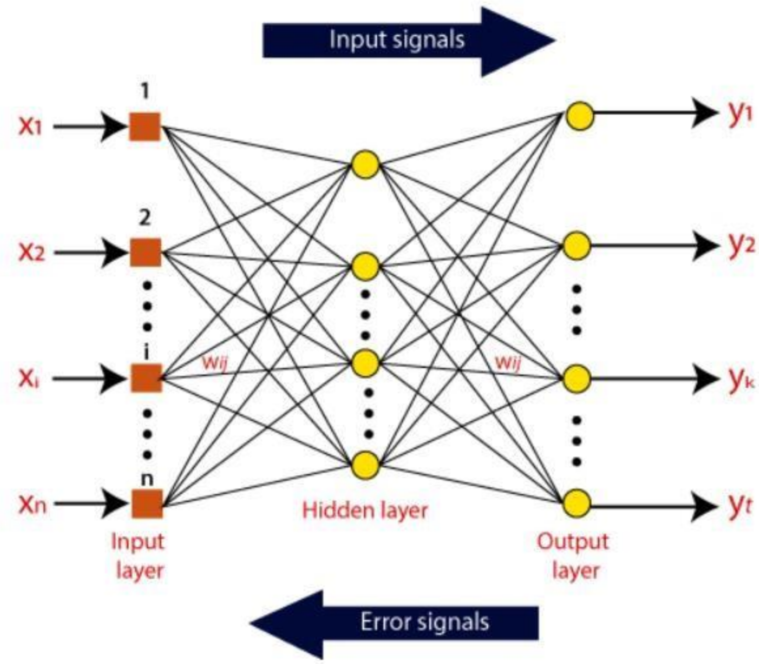


Figure 13: Working of ANN

4.5 Key Challenges

Audio classification using machine learning (ML) and deep learning (DL) poses many challenges that developers must solve to create good and accurate models. Here are some key challenges and solutions:

1. **Variability of audio files:**
    a. Challenges: The quality of audio files can vary across background noise, recorded events, and speakers.
    b. Solution: Augmentation technique can be used to increase the diversity of the training process. This involves applying a change to the audio file, such as changing the pitch, extending the duration, or adding background noise. This helps the model better generalize to unseen variations.
2. **Limited Labelled Data:**
    a. Challenges: Collecting large labelled data to train deep learning models can be challenging, especially for certain tasks or special projects.
    b. Solution: Transfer learning can be implemented using pre-trained models on large data sets. Developing these models on small data sets specific to the task at hand can produce good results using less data.
3. **Computational Complexity:**
    a. Challenges: Training deep learning models for audio classification can be computationally expensive, especially for large neural network architectures.
    b. Solution: Optimization methods such as model pruning, quantization, and compression can help reduce computational requirements. Additionally, leveraging hardware accelerators like GPUs or TPUs can significantly speed up the training process.
4. **Handling Long Sequences:**
    a. Challenge: Audio data is often presented as a sequence of samples, and capturing long-term dependencies in the data can be challenging for some models.

    b. Solution: Recurrent Neural Networks (RNNs) or attention mechanisms can be employed to handle long sequences effectively. These architectures allow the model to consider information from different parts of the input sequence when making predictions.

5. **Class Imbalance:**
    a. Challenge: In audio classification tasks, some classes may have significantly fewer examples than others, leading to imbalanced datasets.
    b. Addressing Solution: Techniques such as oversampling the minority class, under sampling the majority class, or using class weights during training can help balance the impact of different classes on the model.
6. **Real-time Processing:**
    a. Challenge: For applications that require real-time audio classification, the model must make predictions quickly.
    b. Addressing Solution: Model quantization, simplification of architectures, and efficient implementation using hardware accelerators can help achieve real-time processing. Additionally, sliding window approaches or streaming models can be employed for continuous audio classification.
7. **Interpretable Representations:**
    a. Challenge: Deep learning models are often considered as "black boxes," making it challenging to understand how they arrive at certain predictions.
    b. Addressing Solution: Techniques such as layer-wise relevance propagation (LRP) or attention mechanisms can provide insights into which parts of the input contribute to the model's decision. Employing interpretable models or techniques can enhance the trustworthiness of the system

4.6 Testing Strategy:

Testing is a critical aspect of the development process for audio classification using machine learning (ML) and deep learning (DL) models. A well-designed testing strategy helps ensure model reliability, accuracy, and robustness. Here are the key elements to consider in your testing strategy:

1. Data distribution: Training, validation, and test sets: Divide the dataset into three parts: a training set for model training, a validation set for hyperparameter tuning, and a test set for final evaluation. Ensure that the distribution of classes is representative in each set.
2. Evaluation metrics: Class-specific metrics: Due to potential class imbalance in audio datasets, use metrics such as precision, recall, and F1 score for each class to comprehensively evaluate performance.
3. Overall metrics: Consider using overall accuracy, confusion matrices, and ROC curves to assess model performance across classes.
4. Cross Validation: K-fold cross-validation: Perform K-fold cross-validation to assess model performance across different data distributions. This helps ensure that model performance is consistent and not overly influenced by a particular subset of the data.
5. Durability testing: Noise and Distortion: Test the model's performance at different noise levels and different types of distortion to assess its resilience to real-world changes in sound quality.
6. Variability in Speakers and Accents: Evaluate the model's ability to generalize across speakers and accents.
7. Time analysis: Time-based testing: If the application involves real-time processing, test the performance of the model over time to ensure that it meets all latency requirements. Temporal generalization: Evaluate how well the model generalizes to unseen data recorded at different times.
8. Adversary Testing: Counterexamples: Test the model against adversarial examples created to trick the system. This helps to identify weak points and potential weaknesses of the model.
9. Environmental conditions: Test the performance of the model in different environmental conditions, especially if the application is designed to run in different settings.

10. Testing on unseen data: Evaluate the model on data that was not present in the training or validation sets to assess its generalization capabilities.
11. Model Interpretability: For interpretability, use techniques such as feature importance analysis or model-agnostic interpretability tools to understand how the model makes decisions.
12. System integration: If the audio classification model is part of a larger system, perform integration testing to ensure seamless interaction with other components.
13. Performance monitoring: Implement continuous monitoring of model performance in production to detect any degradation over time. This can include tracking accuracy, latency and other relevant metrics.
14. Bias assessment: Evaluate the model for biases related to gender, ethnicity, or other sensitive attributes to ensure fair and unbiased predictions.
15. Comprehensive reporting: Document the testing process, including datasets used, testing methodologies and results. Provide a comprehensive report detailing the strengths and weaknesses of the model.

By incorporating these testing strategies, developers can ensure that the audio classification model is robust, reliable, and capable of providing accurate predictions in a variety of scenarios and conditions. Continuous monitoring and regular re-evaluation are necessary to maintain the effectiveness of the model over time.

4.7 Test cases and Outcomes:

Test cases are essential to evaluate the performance of multi-class classification projects. Here are some examples of test cases and their expected results.

Test case 1: Data preprocessing Objective: Ensuring proper preprocessing of model input data. Apply MFCC extraction or spectral-oriented generation to raw audio files. Normalize features. Expected results: Processed dataset ready for model training with standard input features.

Test case 2: Model training Aim: to validate the training process of a multi-class classification model. Model training using the training dataset. Monitor casualty and accuracy metrics during training sessions.

Expected result: The model converges by reducing losses and increasing accuracy in the training set.

Test case 3: Model evaluation Objective: To evaluate the performance of the model on unseen data. Use the trained model to predict classes on separate validation/test datasets. Calculation of evaluation criteria (accuracy, precision, recall, F1 score) for each class. Expected Results: Obtain satisfactory performance metrics that reflect the model's ability to generalize to new data.

Test Case 4: Error Analysis Objective: Identify and understand the weak points of the model. Analyze classified samples (samples where the predicted class is different from the actual class). Review confusion matrices and class performance metrics. Expected results: Gain insight into which classes the model is having trouble with and potential reasons for misclassification.

Test Case 5: Performance with unbalanced data .Aim: To evaluate the performance of models for unbalanced classes. See performance metrics for datasets with unbalanced class distribution. Use techniques such as oversampling, under sampling and stratified weighting. Expected results: Improved performance for less represented classes without significantly reducing overall accuracy.

Chapter 05: Results

5.1 Discussion on the Results Achieved

As mentioned in the previous section, the proposed design (called ANN) achieved an accuracy of 82% when classifying sounds on the UrbanSound8K dataset. However, this value is lower than previously published testing of these data.

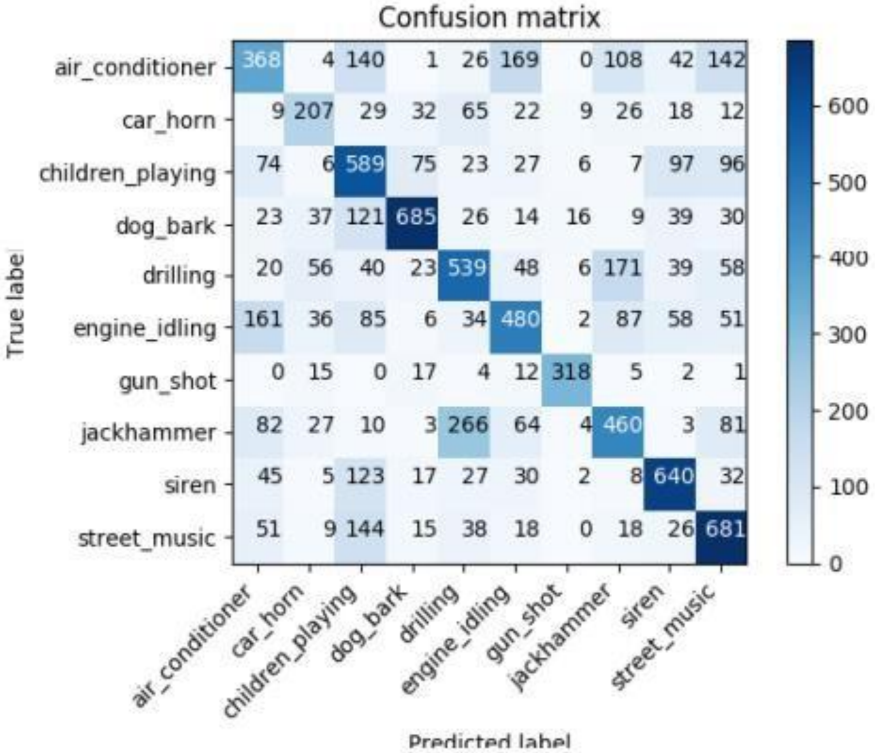
Table 2: Performance Report

System	Features	Accuracy
SVM	MFCC	89.75%
SKM	Log-Mel	81.76%
ANN	Log-Mel	85.78%
SB-ANN	Log-Mel	76.88%

As mentioned earlier, one of the advantages of using the peer-to-peer approach is that it allows the network to create resources appropriate to the problem. As reported in the literature, this can solve or reduce like problems. Citing documentation, it is explained which of the three sets of bands are most confusing due to their timbral similarities: cold weather that idles engines; drill bits; and children are playing with music. The ANN model also has the same problem, as shown in the confusion matrix in Figure 5.2.

The class referred to is the class that is most easily confused with ANN and its confusion matrix is similar to the matrix prepared in the literature. Also, in this case, there is a lot of confusion between classes because model cannot find the correct model. Regarding the reasons for the poor performance are not clear. A similar formula was applied to the sound naming problem and good results were obtained. As with normalization experiments, one of these reasons could be the limitation of training data in the data used to train ANN.

set.



	air_conditioner	car_horn	children_playing	dog_bark	drilling	engine_idling	gun_shot	jackhammer	siren	street_music
air_conditioner	368	4	140	1	26	169	0	108	42	142
car_horn	9	207	29	32	65	22	9	26	18	12
children_playing	74	6	589	75	23	27	6	7	97	96
dog_bark	23	37	121	685	26	14	16	9	39	30
drilling	20	56	40	23	539	48	6	171	39	58
engine_idling	161	36	85	6	34	480	2	87	58	51
gun_shot	0	15	0	17	4	12	318	5	2	1
jackhammer	82	27	10	3	266	64	4	460	3	81
siren	45	5	123	17	27	30	2	8	640	32
street_music	51	9	144	15	38	18	0	18	26	681

Figure 14: Confusion Matrix of CNN Model

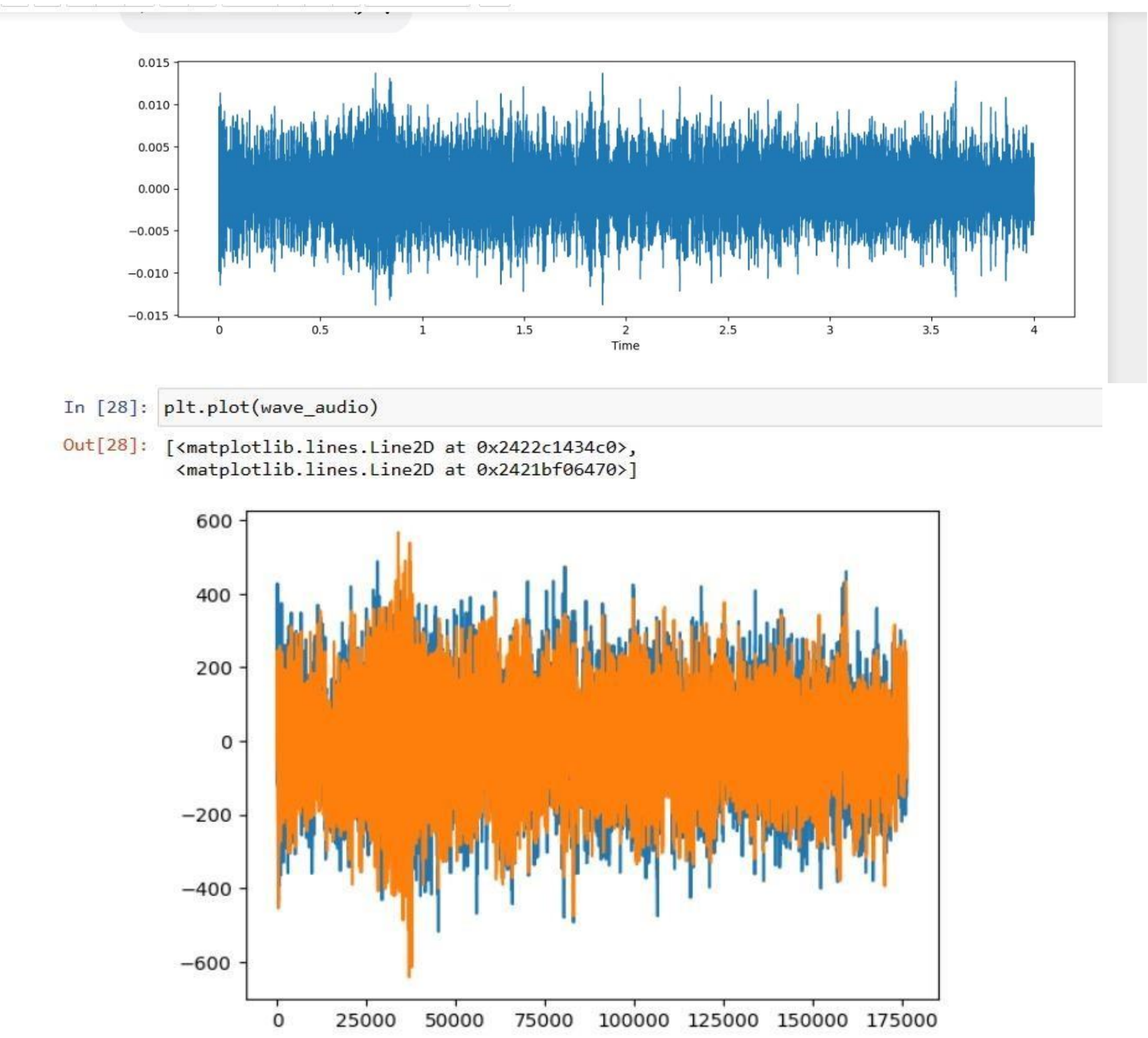
The requirement for a large amount of training data is a well known problem in deep learning, given the high dimensionality of raw waveform data [50], this requirement might be more strict in end-to-end approaches to audio.

5.2 Interpretation Of Result

Inspired by two lines of evidence which inspired this article, the network can find the frequency decomposition from the original waveform. Compared to these models, ANN cannot achieve good results in noise classification.

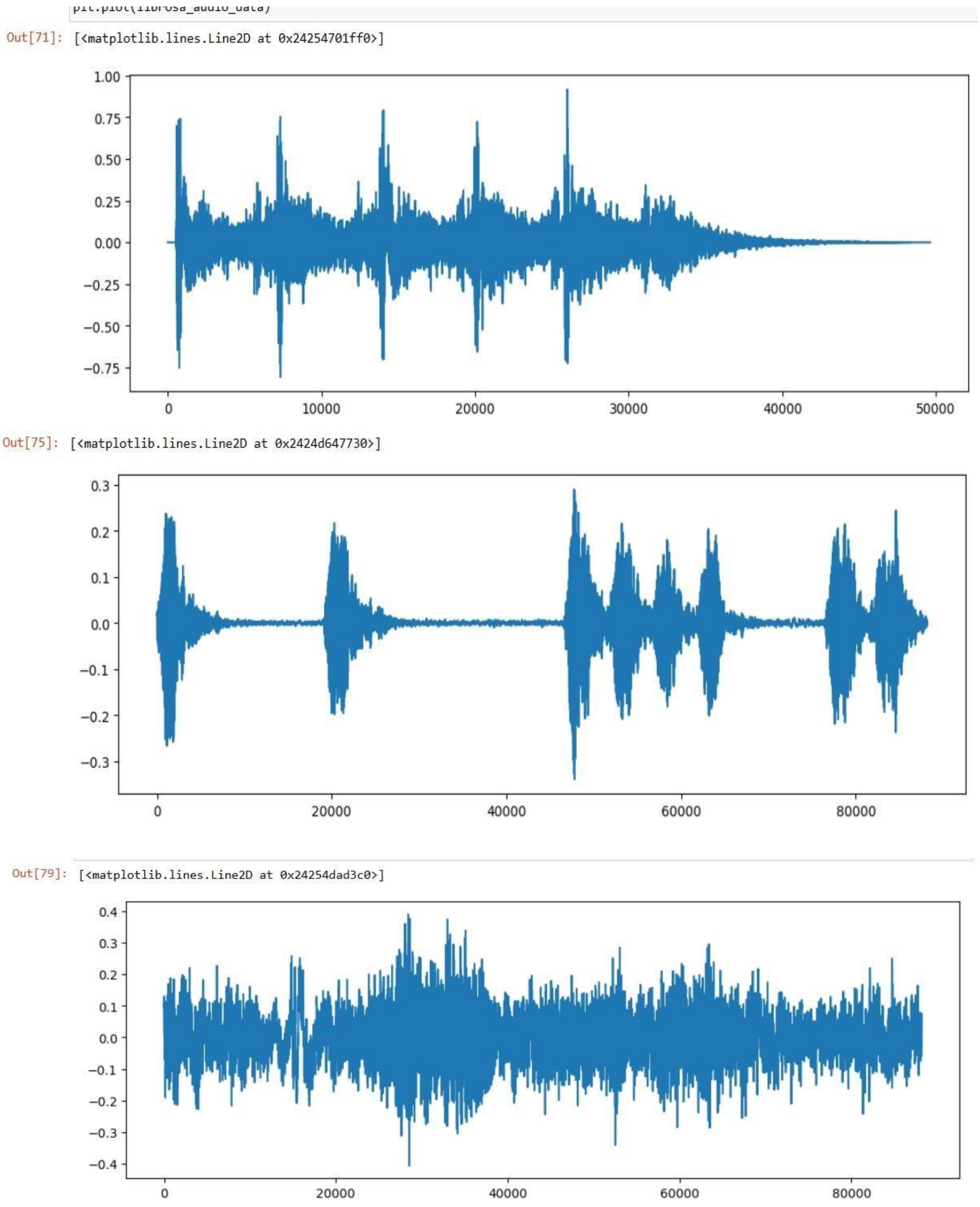
Therefore, the filters of this network do not think that there is a clear picture in the information about. One of the reasons for this significant difference is the data used in each case, especially

Sound Net, which uses data from 2 million videos. The fact that the filters were very noisy, as shown in Figure 5.3, inspired to conduct research on different weighted initialization parameters for random initializations used gammatone pulse response to initialize weights and this may be useful in their work.



```
In [28]: plt.plot(wave_audio)
Out[28]: [<matplotlib.lines.Line2D at 0x2422c1434c0>,
          <matplotlib.lines.Line2D at 0x2421bf06470>]
```

Figure 15:   Wave analysis of input audio files



```
plt.plot(librosa_audio_data)
```

```
Out[71]: [<matplotlib.lines.Line2D at 0x24254701ff0>]
```

```
Out[75]: [<matplotlib.lines.Line2D at 0x2424d647730>]
```

```
Out[79]: [<matplotlib.lines.Line2D at 0x24254dad3c0>]
```

Figure 16: Frequency and Amplitude analysis of input audio files

5.3 Comparison With Existing Solution:

In our next experiment, we compare the performance of DNN with the two classifiers described in Section 3 .Table 3 shows the classification results of DNN classifier compared to GMM and SVM classifiers. We can see that the performance of GMM is worse than the two classifiers, while the performance of SVM and DNN classifiers is comparable. The weakness of GMM classifiers for sound perception tasks that we found in our experiment is consistent with the findings in [18]. The DNN classifier achieved the best performance in most of the single-class results, except for the music class, where SVM performed better.

Table 3:

Audio class	GMM	SVM	ANN
Crowd	24.48	19.25	17.89
Traffic	23.85	16.75	15.71
Animal	15.48	10.18	8.65
AVEARGE	21.27	15.39	14.08

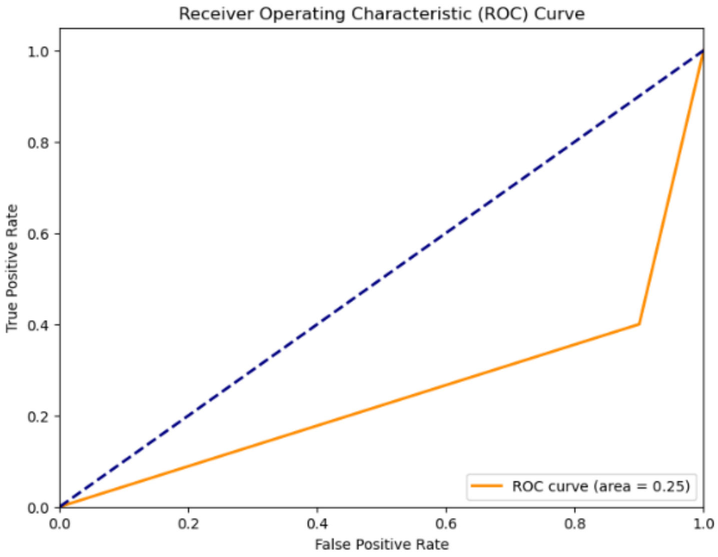
We do simple fusion at the fractional level. The new score is created by adding the last score obtained from ANN to the last score of SVM. This new fusion score is used to calculate the EER. Fusion results are shown in Table 6. From these results, we learn that the fusion score of ANN and SVM classifiers provides a 6.5% improvement over ANN alone. Most of the benefits in fusion are achieved through improvements in music classification. Similarly, if we evaluate the relative improvement in fusion score compared to SVM, most of the improvement is achieved in the Applause group, with the overall relative improvement being 8.0%.

Table 4:

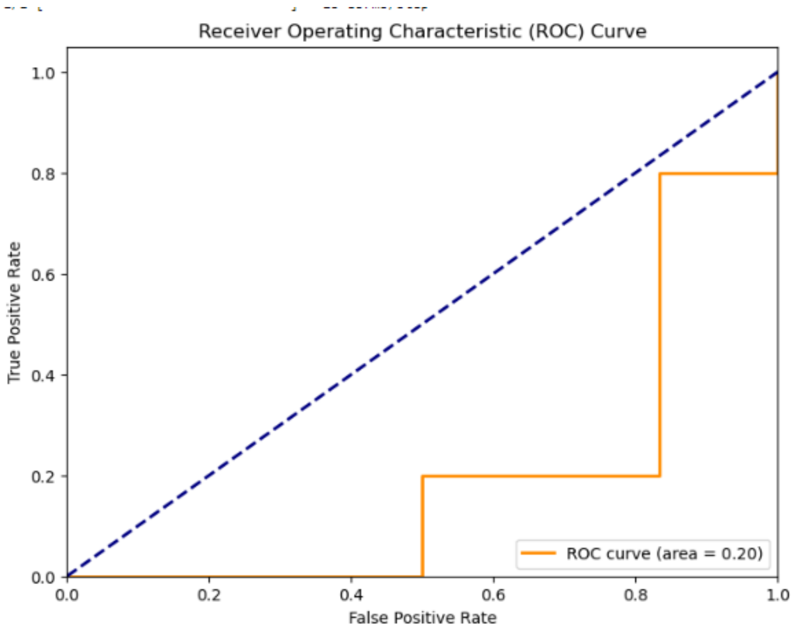
Audio classes	ANN+SVM
Crowd	18.04
Traffic	14.88
Animal	17.99
AVEARAGE	16.97

Comparative Analysis using ROC

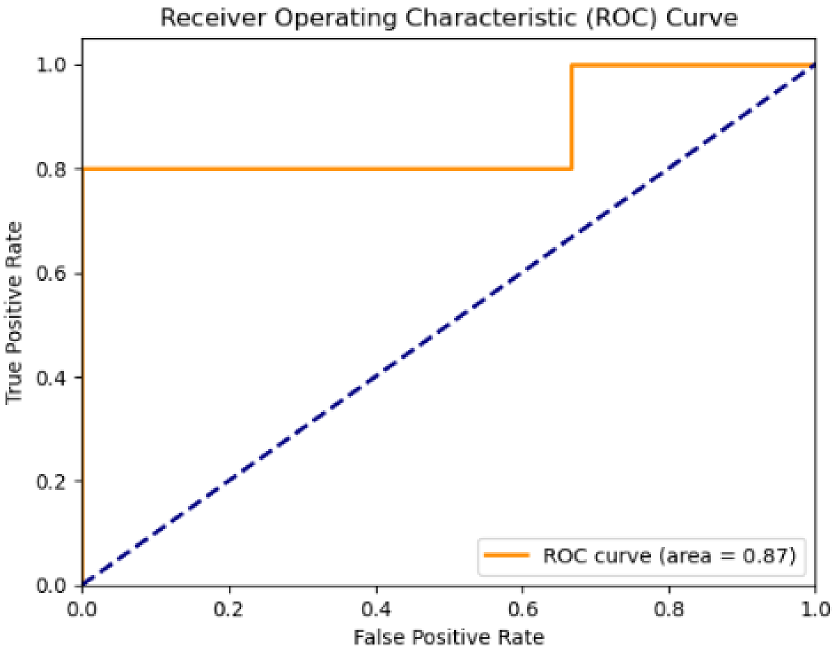
ROC curves of all the four models are plotted as shown in Figure 20 to perform comparative analysis. ROC or the Receiver Operating Characteristic Curve is a graph that depicts performance of classification models at all thresholds. ROC curve is basically tradeoff between True Positive Rate and False Positive Rate. If area under ROC curve is less than 0.5 denotes classifier is not working well, else it denotes classifier is making correct predictins more than half of the time.



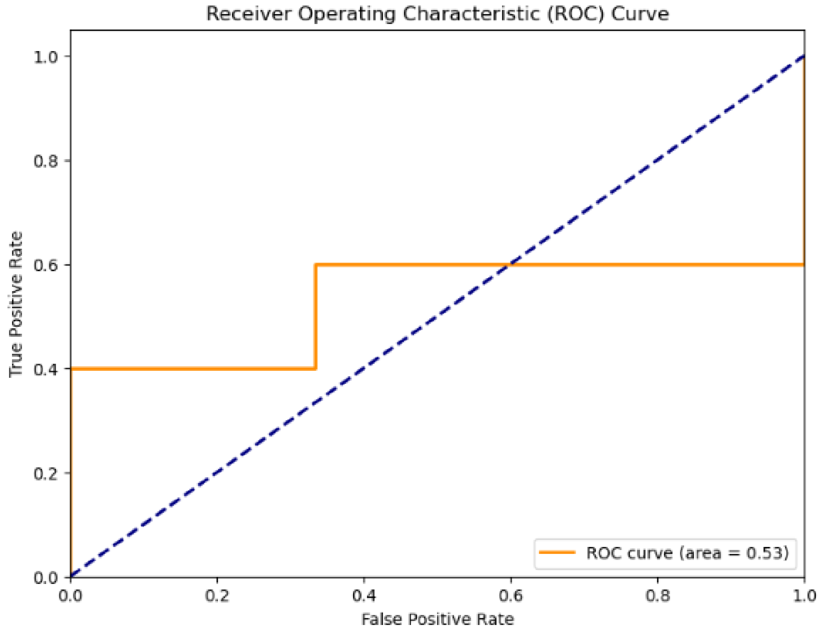
a.


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.20)

b.


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.87)

c.



d.

Figure 17: ROC Curve of a. Decision Tree b. LSTM c. GRU and d. CNN models

As Figure 17 demonstrates GRU model has the highest ROC area under curve value of 0.87 which makes it most suitable for our final demonstration on a random audio signal as made in previous chapter.

Chapter 06: Conclusion and Future Scope:

6.1 Application of the Major or Project

Audio Classification has various applications across different domains due to its potential to enhance human-computer interaction, improve user experience, and provide valuable insights into emotional states. Here are some notable applications of Audio Classification:

The system is able to find audio in a variety of lighting situations. There are some situations in which perfect accuracy is not possible. One of the causes is the size and form of the structuring element. When we vary the size and shape of the structuring element, the detection of the number of audio may also change. The non audio model's restriction, which identifies both audio and background objects, is another factor. Making a distinction between audio and non-audio regions can help to further reduce false positives.

6.2 Limitation of the Project

Despite the potential utility of Audio Classification there are a number of obstacles and restrictions that must be taken into account:

1. Subjectivity and Cultural Variability: Emotions are subjective and may be expressed differently by different people and cultures. A audio classification model's ability to generalise depends on how successfully it was trained on a certain cultural or linguistic group.
2. Limited Training Data: Training audio classification models with labelled information is hard, because existing datasets could not represent all cultural and emotional variances. The results may be biased and less applicable to a wider population.

3. Speech Variability: The accuracy of audio classification models may be affected by variations in audio patterns caused by things like accent, speech rate, and personal variations.
4. Lack of Standardization: The assessment of audio classification systems is not yet supported by any established standards. It is challenging to assess the performance of various models since different research may use different datasets, feature extraction techniques, and evaluation measures.
5. Real-time Processing: Many applications rely on real-time processing, however many audio classification models, particularly on devices with limited resources, may not be able to handle real-time situations due to their high computational demands.

Overcoming these limitations will need ongoing research and development in the field of audio classification . If we want audio classification systems to be more useful and successful, we need to make improvements to data gathering, model robustness, cross-cultural adaption, and ethical concerns.

6.3 Future Work

The results of the experiments in this thesis point to several directions and options for further experimentation with the network architecture:

Recurrent Networks: Some deep learning methods that rely on handcrafting also use recurrent techniques after several convolutional blocks . In addition, it will also expand layers such as long-term memory into the design.

Gated Activation Unit: Recent architectures proposed the so-called gated activation unit, which combines two different weights into two activation functions to achieve interaction with the input. Additional network upgrades may use these instead of ReLU.

Combining original waveforms with other features: Given the failure of the request end-to-end approach, additional features must be applied to the waveforms. These features can be based on stereo data and are easily obtained by waveform subtraction and require technical skill or technical skill to calculate It does not require spectral features.

6.4 Conclusion:

I have implemented a Multi-Column Convolutional Neural Network (MFCC) to precisely calculate the audio of different types , from different perspective and scenarios. I have used urban sound dataset, which contains a total of 8710 participants annotated under two section, in order to compare my result with other studies and techniques. As of right now, this dataset contains the most marked heads for audio counting. On all test datasets, our approach surpasses cutting-edge audio counting techniques

Additionally, the excellent generalizability of the suggested model is demonstrated by the fact, that just by tuning parameters and final few layers, this model can be used for a number of applications across many sectors.

Implementation of 3D Artificial Neural Network can predict the audio behaviour with an accuracy of 61.2%, and the architecture has proven to be effective in large gatherings. Various neural network architectures were evaluated to find the appropriate model to describe sound events for UrbanSound8K. The starting point of the architecture is the network proposed in because it is the simplest network using raw waveforms as input. Since then, the hyperparameters analyzed are step size, number of convolution filter, number of convolution blocks, normalization with dropout, Regularization .After evaluating several blocks, ANN, a contentbased design, is recommended. phase, the convolution phase and the entire coupling phase. Unfortunately ANN is more demanding than basic.

References

[1] B. Vimal, M. Surya, Darshan, V. S. Sridhar and A. Ashok, "MFCC Based Audio Classification Using Machine Learning," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021.

[2] K. J. Piczak, "Environmental sound classification with convolutional neural networks," 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA, 2015.

[3] A. Copiaco, C. Ritz, S. Fasciani and N. Abdulaziz, "Scalogram Neural Network Activations with Machine Learning for Domestic Multi-channel Audio Classification," 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, United Arab Emirates, 2019..

[4] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey and P. Tiwari, "Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network," in IEEE Access, vol. 7, pp. 7717-7727, 2019.

[5] A. Hamza et al., "Deepfake Audio Detection via MFCC Features Using Machine Learning," in IEEE Access, vol. 10, pp. 134018-134028, 2022, doi:

10.1109/ACCESS.2022.3231480.

[6] J. Acharya and A. Basu, "Deep Neural Network for Respiratory Sound

Classification in Wearable Devices Enabled by Patient Specific Model Tuning," in IEEE Transactions on Biomedical Circuits and Systems, vol. 14, no. 3, pp. 535-544, June 2020, doi: 10.1109/TBCAS.2020.2981172.

[7] M. Esposito, G. Uehara and A. Spanias, "Quantum Machine Learning for Audio Classification with Applications to Healthcare," 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), Corfu, Greece, 2022, pp. 1-4, doi: 10.1109/IISA56318.2022.9904377 Recognition Workshops, CVPR Workshops 2009. 1446-1453. 10.1109/CVPRW.2009.5206771.

[8] I. McLoughlin, H. Zhang, Z. Xie, Y. Song and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 540-552, March 2015.

[9] S. B. Shuvo, S. N. Ali, S. I. Swapnil, M. S. Al-Rakhami and A. Gumaei, "CardioXNet: A Novel Lightweight Deep Learning Framework for Cardiovascular Disease Classification Using Heart Sound Recordings," in IEEE Access, vol. 9, pp. 36955-36967, 2021.

[10] T. -E. Chen et al., "S1 and S2 Heart Sound Recognition Using Deep Neural

Networks," in IEEE Transactions on Biomedical Engineering, vol. 64, no. 2, pp. 372380, Feb. 2017.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In Proc. IEEE ICASSP 2017, New Orleans, LA, 2017.

[12] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, et al. An exemplarbased nmf approach to audio event detection. In Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on, pages 1–4. IEEE, 2013.

[13] D. Giannoulis, A. Klapuri, and M. D. Plumbley. Recognition of harmonic sounds in polyphonic audio using a missing feature approach. In 2013 IEEE International

Conference on Acoustics Speech and Signal Processing. Institute of Electrical and Electronics Engineers (IEEE), may 2013.

[14] C. Clavel, T. Ehrette, and G. Richard. Events detection for an audio-based surveillance system. In Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, pages 1306–1309. IEEE, 2005.

[15] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23(12):2136–2147, 2015.

[16] M. J. Kim and H. Kim. Automatic extraction of pornographic contents using radon transform based audio features. In 2011 9th International Workshop on ContentBased Multimedia Indexing (CBMI). Institute of Electrical and Electronics Engineers (IEEE), jun 2011.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition.

Neural computation, 1(4):541–551, 1989.

[18] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, volume 2, pages II–1941. IEEE, 2002.

[19] K. J. Piczak. Environmental sound classification with convolutional neural networks. In Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on, pages 1–6. IEEE, 2015.

[20] K. J. Piczak. Esc: Dataset for environmental sound classification. In

Proceedings of the 23rd ACM international conference on Multimedia, pages 1015– 1018. ACM, 2015.

[21] L. R. Rabiner and B.-H. Juang. Fundamentals of speech recognition. 1993. [22] R. Ranft. Natural sound archives: past present and future. Anais da Academia Brasileira de Ciências, 76(2):456–460, jun 2004.

[23] J. Salamon and J. P. Bello. Feature learning with deep scattering for urban sound analysis. In Signal Processing Conference (EUSIPCO), 2015 23rd European, pages 724–728. IEEE, 2015.

[24] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3):279–283, 2017.

[25] J. Salamon and J. P. Bello. Unsupervised feature learning for urban sound classification. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 171–175. IEEE, 2015.

SPP-1

ORIGINALITY REPORT

17%	15%	3%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	projekter.aau.dk Internet Source	3%
2	www.analyticsvidhya.com Internet Source	2%
3	pdfs.semanticscholar.org Internet Source	2%
4	www.geeksforgeeks.org Internet Source	2%
5	Submitted to University of Bedfordshire Student Paper	1%
6	Submitted to SASTRA University Student Paper	1%
7	fastercapital.com Internet Source	1%
8	cse.anits.edu.in Internet Source	1%
9	ir.juit.ac.in:8080 Internet Source	1%

10	Submitted to Universiti Teknologi Petronas Student Paper	1 %
11	Submitted to Federal University of Technology Student Paper	<1 %
12	Submitted to Universiti Sains Malaysia Student Paper	<1 %
13	Lidong Yang, Jiangtao Hu, Zhuangzhuang Zhang. "Audio Scene Classification Based on Gated Recurrent Unit", 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019 Publication	<1 %
14	assets.researchsquare.com Internet Source	<1 %
15	www.ijraset.com Internet Source	<1 %
16	Submitted to Indian Institute of Technology, Kanpur Student Paper	<1 %
17	cland.lsce.ipsl.fr Internet Source	<1 %
18	tel.archives-ouvertes.fr Internet Source	<1 %

19	Submitted to Virginia Polytechnic Institute and State University Student Paper	<1 %
20	Submitted to Wawasan Open University Student Paper	<1 %
21	Submitted to Liverpool John Moores University Student Paper	<1 %
22	Submitted to University of Essex Student Paper	<1 %
23	link.springer.com Internet Source	<1 %
24	"Intelligent Computing & Optimization", Springer Science and Business Media LLC, 2022 Publication	<1 %
25	eudl.eu Internet Source	<1 %
26	ActEd Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ___________________________ __Department: ________________ Enrolment No _________

Contact No. ______________________________E-mail. ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com