321726

SOUND CLASSIFICATION

A PROJECT REPORT

Prepared at

BISAG-N

ISO 9001:2008
ISO 27001:2013
CMMI LEVEL-5

Submitted by

Rushikesh Palnitkar

Enrolment No: 191310132080

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

ICT Department

Adani Institute of Infrastructure Engineering, Ahmedabad

adani Institute of Infrastructure

Gujarat Technological University, Ahmedabad

May 2023

321726

Adani Institute of Infrastructure Engineering, Ahmedabad

Shantigram Township, Nr.Vaishnodevi Circle, Sarkhej - Gandhinagar Hwy, PO, Adalaj, Gujarat 382421

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Sound Classification** has been carried out by **Rushikesh Palnitkar** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Information and Communication Technology, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2022-23.

Dr. Anuj Kumar Singh
Internal Guide

Dr. Ajay Kumar Vyas
Head of the ICT Department

Gujarat Technological University

Adani Institute of Infrastructure Engineering

Bhaskaracharya National Institute for Space Applications and Geo informatics
MeitY, Gover nment of India
Phone: 079 23213081 Fax: 079 23213091
mail: info@bisag.gujarat.gov.in, website: https://bis n.gov.in/

IS 9001:2008
IS 27001:2013
MMI LEVEL

321726

CERTIFICATE

This is to certify that the project report compiled by ***Mr. Priyansh Ranpura,Mr. Soham Patel and Mr. Rushikesh Palnitkar*** *students of 8th Semester* ***BE - ICT from Adani Institute of Infrastructure Engineering, Gujarat Technological University, Ahmedabad*** *have completed their final Semester internship project satisfactorily. To the best of our knowledge this is an original and bonafide work done by them. They have worked on model creation on "****Sound Classification****", starting from January 23rd, 2023 to May 10th, 2023.*

During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.

We wish them every success.

Prem Patni
External Co-Guide
BISAG- N, Gandhinagar

Punit Lalwani
CISO,
BISAG-N,
Gandhinagar

321726

Adani Institute of Infrastructure Engineering, Ahmedabad

Shantigram Township, Nr. Vaishnodevi Circle, Sarkhej - Gandhinagar Hwy, PO, Adalaj, Gujarat 382421

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled **Sound Classification** submitted in partial fulfilment for the degree of Bachelor of Engineering in Information And Communication Technology to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at **BISAG-N** under the supervision of **Prem Patni, Dr. Anuj Kumar Singh** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student	Sign of Student
Rushikesh Palnitkar	_____

ACKNOWLEDGEMENT

Knowledge is continuous process. At this moment of our substantial enhancement, we rarely find enough words to express our gratitude towards those who were constantly involved with us during our project. At the outset, we wish to express our sincere gratitude to all who have helped us to complete this project in at most accomplished manner.

I would like to express our gratitude towards faculty member of college for their kind cooperation and encouragement which help us in completion of this project. I would like to express our special gratitude and thanks to industry persons for giving me such attention and time.

First, we are very grateful to our internal guide **Dr. Anuj Kumar Singh** who has guided us to accomplish our project and giving their wide experience of knowledge. We are also very thankful to the **Adani Institute of Infrastructure Engineering,** for allowing us to complete the project within four walls of the computer lab. Finally, we would like to acknowledge and thanks in large measures to all our fellow friend & guides for their support.

- Rushikesh Palnitkar

ABSTRACT

Sound classification has attracted considerable and varied interest in recent years. The majority of the developments in sound categorization techniques included in this research date from the last ten years. The selection of features and classifiers has a substantial impact on how reliable a sound classification system is, similar to speech recognition systems. From early works in content-based audio classification to more recent developments in applications like sound event recognition, audio surveillance, and environmental sound recognition, the features and classifiers used in sound classification systems are examined from a broader perspective. Audio classification is the process of classifying specific sounds into groups, such as speech recognition and environmental sound classification. We have taken the UrbanSound8k dataset which is publicly available for building a suitable model. We have gained some basic domain knowledge regarding audio processing, as in that there are multiple characteristics of an audio file, such as the audio sample rate, audio file duration, file format, whether the audio is mono(single channel) or stereo(double channel) etc. We also learned that audio processing is better performed by analysing the Mel Spectrograms or Mel Frequency Cepstral Coefficients, as they give us the different components which compose a particular audio signal, rather than just analyzing the amplitude of the signal at various times and capturing its patterns. Based on the relevant extracted features, we have tried different ways to create a robust model, we have tried 3 ways to train our model: artificial neural network, convolutional neural network, as well as a convolutional neural network using pytorch library. We also got to know about the very immensely critical role of using GPU in such deep learning techniques, we obtained knowledge about installing the necessary packages and drivers required to get the GPU ready for usage by our deep learning models, that is, if the GPU was CUDA-enabled. GPU enables significantly higher processing/computing power which, judging by the immense advancements in this field, could very well become the basic requirement for these purposes. Finally we concluded that a CNN is obviously better suited for this problem statement, as it can effectively capture the different, complex spatial dimensional characteristics which ANN cannot.

Keywords: Audio Classification, Neural Network,ML,DL,pytorch,tensorflow

LIST OF FIGURES

LIST OF ABBREVIATION

ML : Machine Learning
DL : Deep Learning
ANN : Artificial Neural Network
CNN : Convolutional Neural Network
ESC : Environmental Sound Classification
CPU : Central Processing Unit
API : Application Programming Interface
GPU : Graphics Processing Unit
TPU : Tensor Processing Unit
LSTM : Long Short-Term Memory
MFCC : Mel Frequency Cepstral Coefficients

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

This is a brief introduction to the basics of the project, like domain information, purpose, objective, scope, limitations etc.

1.1 PROJECT DETAILS

The project Titled “Audio Classification” is the ML model developed for classification of audio. Audio classification has attracted considerable and varied interest in recent years. The majority of the developments in audio classification techniques included in this research date from the last decade. The selection of features and classifiers has a substantial impact on how reliable a sound classification system is, similar to speech recognition systems. From content-based audio classification to advancements in applications like sound event recognition, and environmental sound recognition, the features used in sound classification systems are examined from a broader perspective.Strong audio classification has enormous potential advantages. From surveillance to house security, it's important to be able to tell apart sounds like a crying baby, air conditioning, or a breaking glass. This project focuses on everyday sounds that make up the background of daily life. These sounds are more susceptible to noise and randomness in both the periodic and aperiodic senses than specialized sounds like speech and music, which already have cutting-edge classification and recognition techniques.Thus, environmental sounds are any sound that is not speech or music and that is created in settings that are typical of day-to-day life. Our effort in this project is primarily focused on classifying and representing such sounds.

1.2 PURPOSE OF PROJECT

The aim of audio classification is to enable machines to automatically recognize and differentiate between various types of audios, such as music, speech, and environmental noises.Powerful capabilities for content management can be

provided by audio classification. The administration of audio can be greatly improved if an audio clip can be automatically categorized and then saved in a structured database. If the content is known, audio coding methods and modules like "librosa" may be enhanced.

1.3 OBJECTIVE OF PROJECT

The goal for this project is Training and testing of two neural networks, including an ANN and a CNN that classify audio samples.

1.4 PROJECT SCOPE AND LIMITATIONS

We used UrbanSound8k dataset for our project. It is a standard dataset. It has certain limitations which are imbalance class distribution and varying sampling rates. Apart from these, there are 10 classes in this dataset with each audio file with upto 4 seconds in length. We had a chance to work with the ESC50 dataset. It has 50 classes. Each class contains 40 audio files of 5 seconds in length. It has the same sampling rate as UrbanSound8k dataset. We have trained and tested our model based on UrbanSound8k. The UrbanSound8K has 10 classes, consisting of 8732 audio files. We have used supervised learning approach for our models. Training and testing the model via unsupervised learning is also applicable.For our project implementation, we had limited knowledge of industrial ML models like how to create them and how to implement them, which affected our progress. We used YouTube playlists to learn the theoretical and practical implementation of ML and DL Models.If determined the correct resources accordingly, we keep have overcome few of the restraints and the breadth of our research would have raised. Extraction of the spectrograms in addition to the pre-processing of the visual and audio entertainment transmitted via radio waves signal accepted continuously plenty come into sight our research that is due to our disadvantage of the information about occupied machines of the signals. So, we had to sanctify our opportunity for the study purpose (10 weeks) in addition we anticipated.

CHAPTER 2 - DETAILS OF SOUND CLASSIFICATION

This chapter deals with the actual theoretical concepts involved in our project.

2.1 GENERAL CLASSIFICATION

Automatic categorization can be executed in abundant ways. The applicable data maybe treated in more or less active ways to tell the dossier's informative conditions. The influence with that attributes of unidentified samples are analysed to equate various classes varies with classifiers as well.Automatic categorization can be executed in abundant ways. The vacant data maybe treated in more or less persuasive ways to disclose the dossier's informative values. The effectiveness accompanying that attributes of unidentified samples are analysed to identify between various classes changes among classifiers also.The two variables that are measured for a group of cells are size and redness. Normal cells are less red and smaller. When compared to decisions based on a single feature, choosing a decision boundary that is a straight line greatly increases classification accuracy.

2.2 FEATURE EXTRACTION

The effectiveness of the classification method depends on how accurately the first step is completed. The feature vector, y, which is made up of a number of features, should be as discriminatory between the classes taken into consideration as possible. The feature vectors should, in theory, be able to distinguish all measured samples from various classes. Although this is not actually possible, the feature extraction step's goal is to learn as much as possible about the observed sequence, x.

In order to extract as much discriminative information as possible from the observed sequence, various audio processing algorithms extract features. The distinction between music and speech would make any attribute that indicates the amount of energy present in the GHz range irrelevant. When the frequency range is too high, that characteristic cannot distinguish between speech and music. The spread of the

frequency spectrum between 0 and 22050 Hz, however, is a feature that provides discriminative information and can be used for classification.The classification accuracy depends on the make-up of the feature vector, y. A feature vector that is well-composed makes classification easier, which also makes the classifier's construction easier. Consequently, the context determines which features to extract. The below figure shows the above described relation between feature extraction and classification.

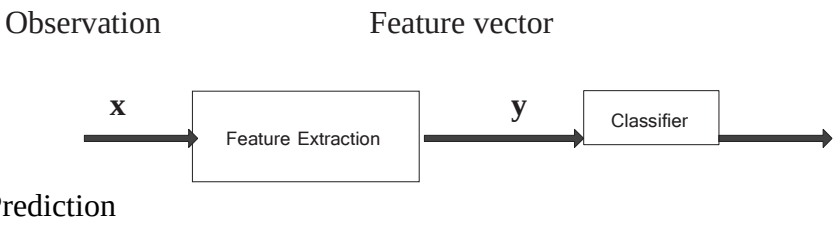


Figure 2.1: Basic Scheme of automatic classification

2.3 LEARNING

The classifier's decision bounds must represent the classes taken into consideration for categorization of additional samples to be possible. Learning, which is the act of using a collection of samples to adjust the classifier to the desired task, results in relevant decision limits. Depending on the algorithm the classifier uses, there are several tuning options. A training set of samples must typically be provided to the classifier in order for it to build decision boundaries.Generally speaking, depending on the sort of classifier that is being addressed, there are two techniques to learn classifiers. Supervised learning is the initial approach to classifier learning. In this technique, the pre-classified samples that make up the training set are used by the classifier to create decision boundaries. Pre-classification is manually carried out. Pre-classification can be challenging in some situations.To pre-classify cell samples as normal or malignant in our prior example, considerable knowledge and skill are required. When the distinctions between various classes are ambiguous and subjective, the pre-classification in the classification of music genres can be

problematic. Unsupervised learning refers to a training strategy without manual pre-classification. With this technique, the classifier creates clusters from unclassified samples in the training set. The classifier may create decision boundaries since the clusters are tagged into various groups.

2.4 CLASSIFICATION

Once trained, the classifier can be used to categories fresh samples. There are many different classification algorithms employed with differing levels of complexity and efficiency because a perfect categorization is rarely feasible. There are many different statistical, instance-based, and clustering-based classifier methods in use.Dealing with feature value variation for samples belonging to particular classes is a classifier's main challenge. The difficulty of the categorization problem could cause this variance to be significant. Decision boundaries should be selected based on combinations of the feature values to maximise classification accuracy.

2.5 ESTIMATION OF CLASSIFIERS PERFORMANCE

To estimate the classifier's accuracy, it is vital to analyse how well a certain categorization scheme works. A classifier's accuracy on future data can be predicted using a variety of techniques when it is being constructed. These techniques enable performance comparison with other classifiers.To increase the generality of predicted error rates, cross-validation is performed. A labelled data set is divided into two pieces to determine the error rates. The training set is utilized in one portion, while the validation set or testing set is used in the other. The classifier is trained using the training set, and its performance is assessed using the testing set.The "10-fold cross-validation" method, which splits the data set into 10 pieces, is a popular version. Ten classifications are carried out, with a different testing set used each time. The final estimate is only the mean value of the ten estimated error rates. By repeating the training process with various training and testing sets, this methodology further generalizes the projected error rates.

CHAPTER 3 - DATABASE AND PREPROCESSING

This chapter discusses the performance issues related to the type and/or structure of databases/datasets.

3.1 Database

Some datasets have been used in the past by researchers to study both supervised and unsupervised learning. These datasets include brief environmental sound samples that were taken from Free sound, a collection of field recordings that includes more than 160,000 audio clips. ESC-10, ESC-50, ESC-US, UrbanSound8k, and ESC-US are a few of the well-known datasets.Except for the ESC-US dataset, which is unlabelled and devoid of any metadata and thus appropriate for unsupervised learning techniques like clustering and anomaly detection, all of these datasets have minimal background noise and are manually annotated. 2000 audio recordings are contained in the ESC-50 dataset, which is separated into 50 classes and arranged into 5 broad categories. In contrast, the ESC10 dataset is a subset of the ESC-50 dataset, which is larger. These two datasets each include audio snippets, each lasting around five seconds.We chose to work with UrbanSound8k instead of the ESC datasets because, based on our preliminary research, we were convinced that our Deep Learning models would perform better with huge data. Urban Sound 8K is an audio dataset that contains 8732 labeled sound excerpts (<=4s) of urban sounds from 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music. The below shown figure is a representation of the comparison between accuracy and dataset size of ML and DL models.

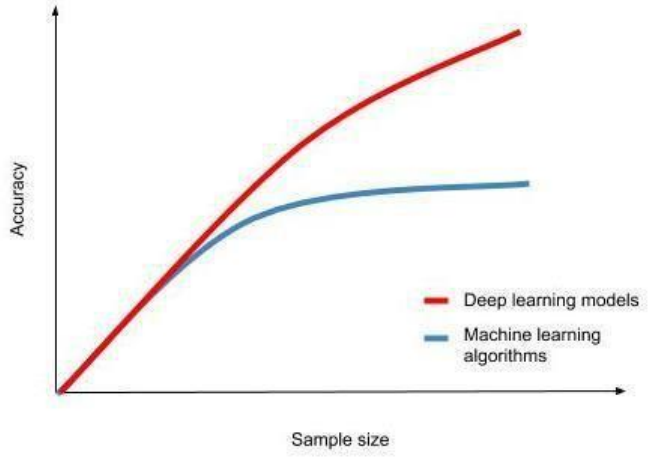


Figure 3.1: Comparison of performance of Deep and ML models w.r.t sample size

Since there is no urban taxonomy to aid in the labelling and creation of various datasets, categorising audio data into distinct classes continues to be one of the major challenges in urban sound research. A taxonomy of urban sounds is created that fits three basic requirements: it must detail low-level sounds, satisfy previously established taxonomies, and contain sounds that contribute to noise pollution in urban settings. Below mentioned figure shows the described taxonomy concepts.

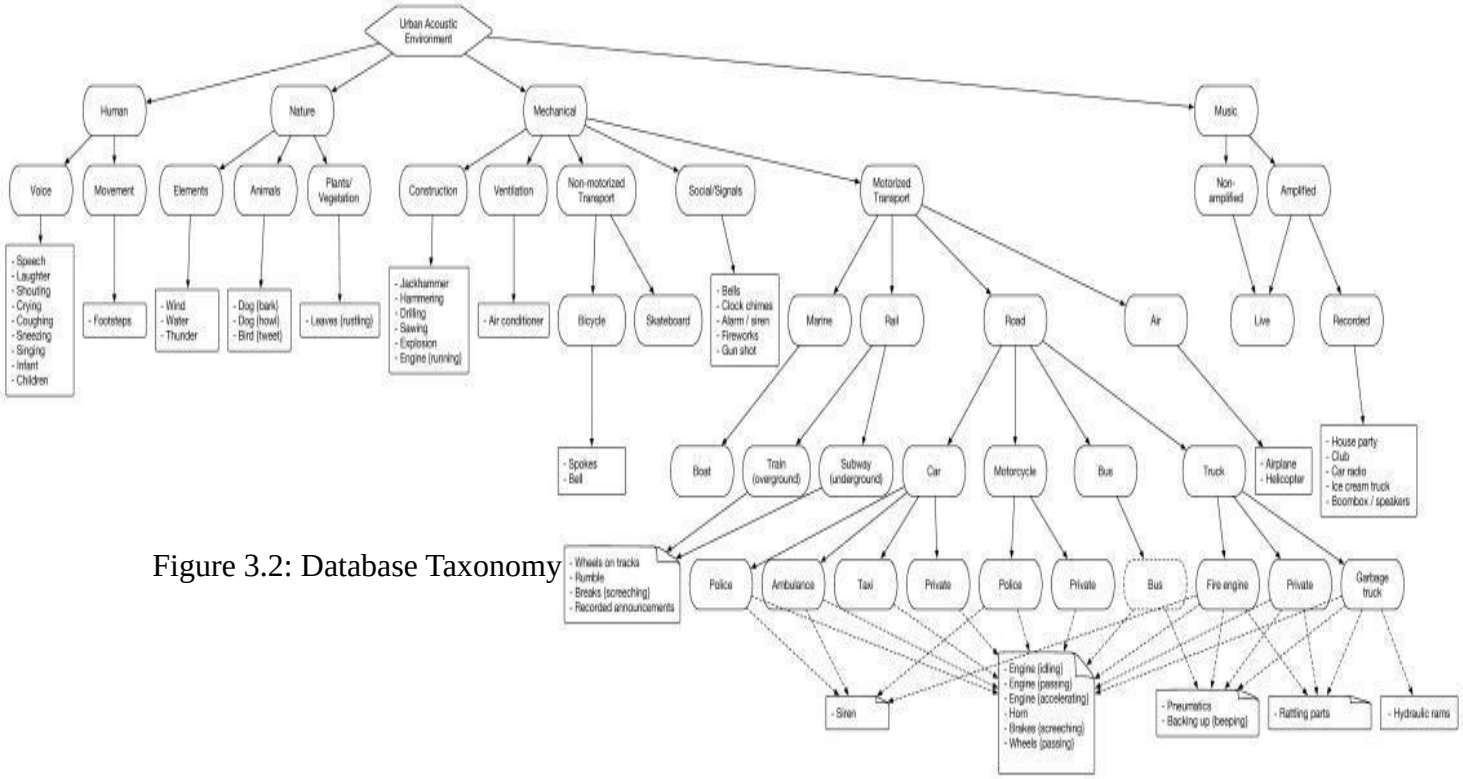


Figure 3.2: Database Taxonomy

Additionally, the following taxonomy, with a duration of 27 hours, forms the foundation of the UrbanSound8K dataset. The dataset includes 8732 labelled audio slices with different audio lengths and sampling rates. The ten categories are: street music, air conditioner, car horn, kids playing, dog barking, drilling, engine idling, gunshot, jackhammer, and children's laughter. The majority of classes contain about 1000 slices, however the gunshot and car horn have just 429 and 374 slices, respectively. As a result, some of the baseline machine learning methods used on the dataset show less accuracy for car horn and gunshot. As wav files were manually labelled, there are fewer classes than in other datasets. Eight columns make up the UrbanSound8K csv file: slice le name, fsID, start, end, salience, fold, classID, and Class name.

- Slice file name - Determines the name of the wav files in the folders.
- FsID - Each wav file is given a unique ID for identification.
- Start and end - Labels the start and end time of every occurrence in the 27 hours of audio.
- Salience - Indicating whether the sound was in the background or foreground of the recordings.
- Fold - Speci es the folder to investigate for each wav file. In each folder attributes are selected based on correlation to circumvent overfitting of training data.
- ClassID and class name - Each class is given a name along with an ID from 1-10.

In order to avoid unrealistically high categorization accuracy, the slices are distributed randomly among the 10 folders. UrbanSound8K subset, a division of the bigger dataset, has been developed and can be utilized for numerous studies on sound source identification. The subset also includes 10 folds with relatively less wav files than the other folds. Finally, UrbanSound8k was chosen to run our model because it has a large and meaningful dataset. Below shown figure is the directory structure of our dataset.

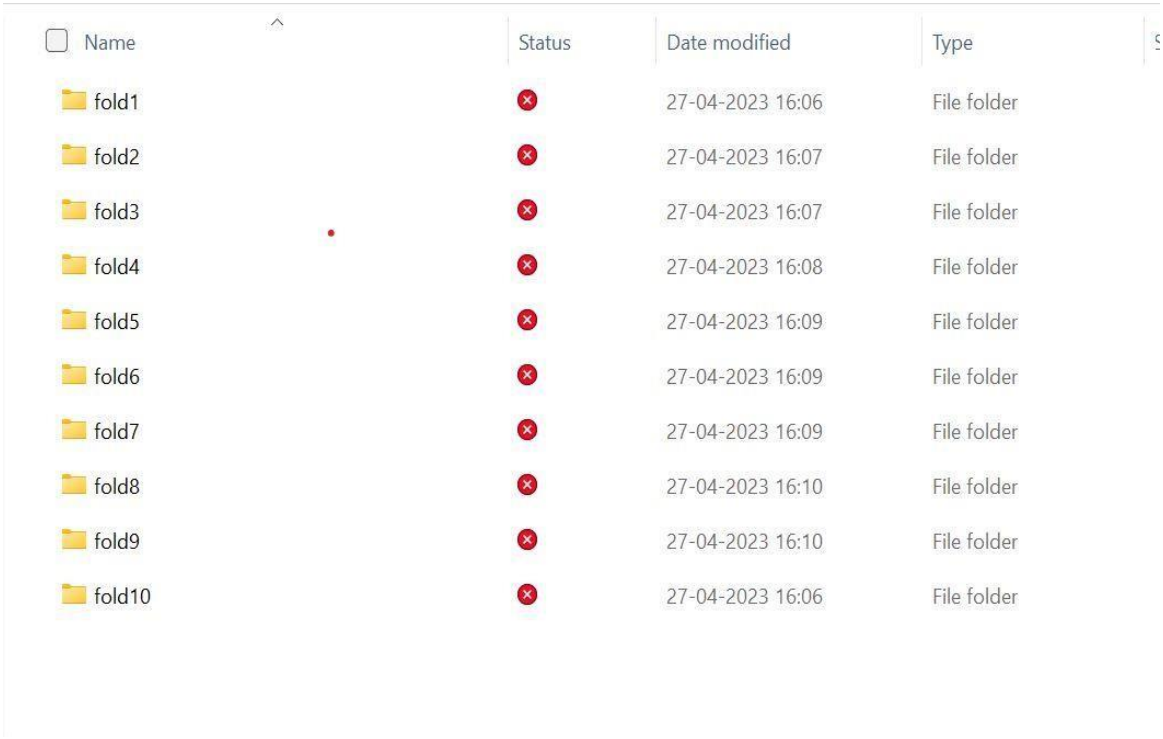
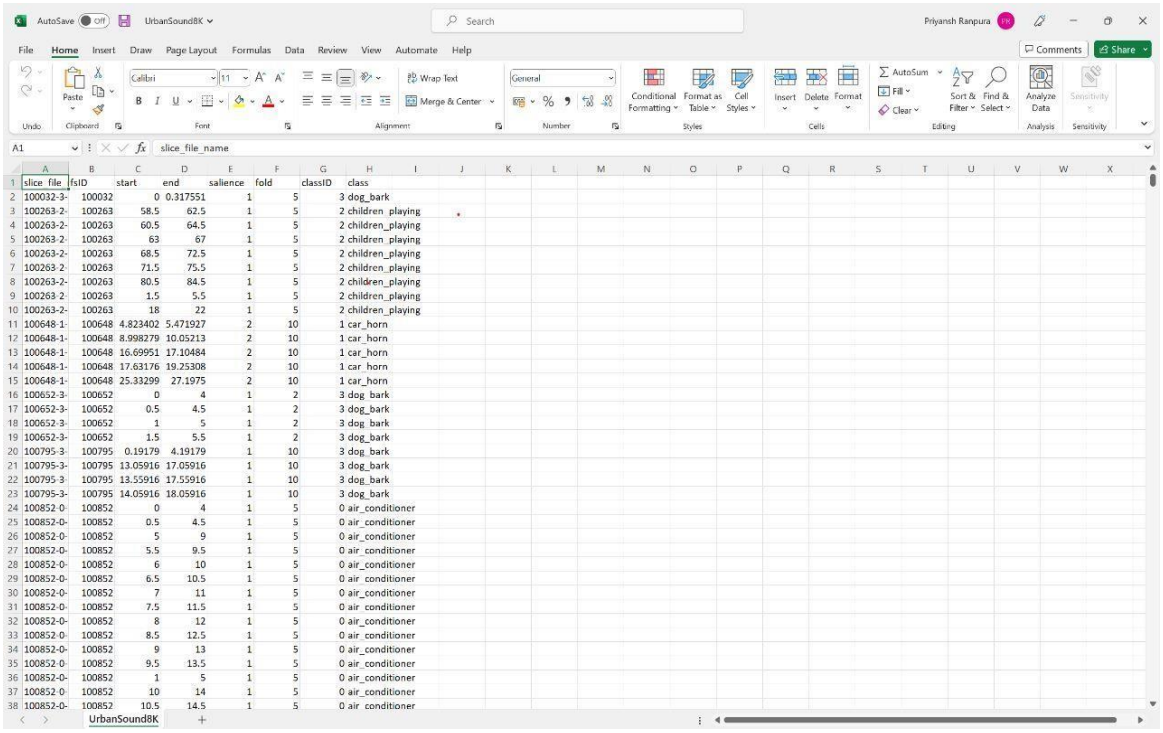


Figure 3.3: Directory Structure

Below figure is our metadata file.



slice_file	fsID	start	end	salience	fold	classID	class
100032-3-	100032	0	0.317551	1	5	3	dog_bark
100263-2-	100263	58.5	62.5	1	5	2	children_playing
100263-2-	100263	60.5	64.5	1	5	2	children_playing
100263-2-	100263	63	67	1	5	2	children_playing
100263-2-	100263	68.5	72.5	1	5	2	children_playing
100263-2-	100263	71.5	75.5	1	5	2	children_playing
100263-2-	100263	80.5	84.5	1	5	2	children_playing
100263-2-	100263	1.5	5.5	1	5	2	children_playing
100263-2-	100263	18	22	1	5	2	children_playing
100648-1-	100648	4.823402	5.471927	2	10	1	car_horn
100648-1-	100648	8.998279	10.05213	2	10	1	car_horn
100648-1-	100648	16.69951	17.10484	2	10	1	car_horn
100648-1-	100648	17.63176	19.25308	2	10	1	car_horn
100648-1-	100648	25.33299	27.1975	2	10	1	car_horn
100652-3-	100652	0	4	1	2	3	dog_bark
100652-3-	100652	0.5	4.5	1	2	3	dog_bark
100652-3-	100652	1	5	1	2	3	dog_bark
100652-3-	100652	1.5	5.5	1	2	3	dog_bark
100795-3-	100795	0.19179	4.19179	1	10	3	dog_bark
100795-3-	100795	13.05916	17.05916	1	10	3	dog_bark
100795-3-	100795	13.55916	17.55916	1	10	3	dog_bark
100795-3-	100795	14.05916	18.05916	1	10	3	dog_bark
100852-0-	100852	0	4	1	5	0	air_conditioner
100852-0-	100852	0.5	4.5	1	5	0	air_conditioner
100852-0-	100852	5	9	1	5	0	air_conditioner
100852-0-	100852	5.5	9.5	1	5	0	air_conditioner
100852-0-	100852	6	10	1	5	0	air_conditioner
100852-0-	100852	6.5	10.5	1	5	0	air_conditioner
100852-0-	100852	7	11	1	5	0	air_conditioner
100852-0-	100852	7.5	11.5	1	5	0	air_conditioner
100852-0-	100852	8	12	1	5	0	air_conditioner
100852-0-	100852	8.5	12.5	1	5	0	air_conditioner
100852-0-	100852	9	13	1	5	0	air_conditioner
100852-0-	100852	9.5	13.5	1	5	0	air_conditioner
100852-0-	100852	1	5	1	5	0	air_conditioner
100852-0-	100852	10	14	1	5	0	air_conditioner
100852-0-	100852	10.5	14.5	1	5	0	air_conditioner

Figure 3.4 : Metadata file(csv)

CHAPTER 4 - DEEP LEARNING ALGORITHM

This chapter deals with the domain of deep learning and some theoretical concept associated with it.

4.1 What is Deep Learning ?

Deep learning, which is fundamentally a neural network with three or more layers, is a subset of machine learning. These neural networks make an effort to mimic how the human brain functions, however they fall well short of being able to match it, allowing it to "learn" from vast amounts of data. While a neural network with a single layer can still make approximations, additional hidden layers can help to refine and optimize for accuracy.

4.2 Machine Learning VS Deep Learning

Structure, labelled data is used by machine learning algorithms to produce predictions, which means that the model's input data is used to identify certain features that are then arranged in tables. This doesn't necessarily imply that it doesn't employ unstructured data; rather, it just indicates that if it does, it typically goes through some pre-processing to put it in a structured manner.Some of the data pre-processing that is generally involved with machine learning is eliminated with deep learning. These algorithms can handle text and visual data that is unstructured and automate feature extraction, reducing the need for human specialists. For instance, we want to categorise a collection of images of various pets by "cat," "dog," "hamster," etc. Deep learning algorithms can decide which characteristics-like ears are most crucial for differentiating one species from another. This hierarchy of features is created manually by a human specialist in machine learning. The deep learning algorithm then fine-tunes and adapts itself for accuracy through the processes of gradient descent and backpropagation, enabling it to make predictions about a fresh animal shot with greater accuracy.Along with being capable of supervised learning, unsupervised learning, and reinforcement learning, machine learning and deep learning models can also learn in other ways. To categorise or make predictions, supervised learning uses labelled datasets; this involves some sort of human interaction to accurately classify input data. Unsupervised learning, in

contrast, does not require labelled datasets; instead, it analyses the data for patterns and groups them according to any identifying traits. A model learns through the process of reinforcement learning to perform an activity in an environment more accurately in order to maximise the reward.

4.3 Artificial Neural Network (ANN)

The phrase “artificial neural network" refers to a branch of artificial intelligence that was inspired by biology and is based on the brain. A computational network based on biological neural networks, which create the structure of the human brain, is typically referred to as an artificial neural network. Artificial neural networks also feature neurons that are linked to each other in different layers of the networks, just as neurons in a real brain. Nodes are the name for these neurons.Artificial neural networks are used in artificial intelligence to simulate the network of neurons that make up the human brain, giving computers the ability to comprehend information and make decisions in a manner similar to that of a person. Computers are programmed to function exactly like a network of interconnected brain cells to create an artificial neural network.The human brain contains about 1000 billion neurons. Between 1,000 to 100,000 association points are present in each neuron. Data is distributedly stored in the human brain, allowing us to simultaneously access many pieces of information from memory as needed. The human brain is said to contain a staggering number of incredible parallel processors.

Table 4.1: Comparing features of Biological Neural Network with Artificial Neural Network

Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

Below figure shows the structure of a biological neuron.

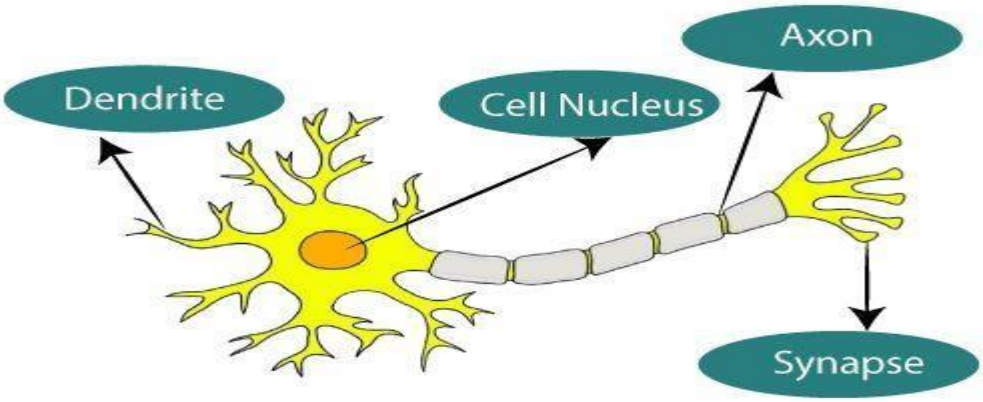


Figure 4.1: Biological Neurons

Below figure shows the structure of an artificial neural network.

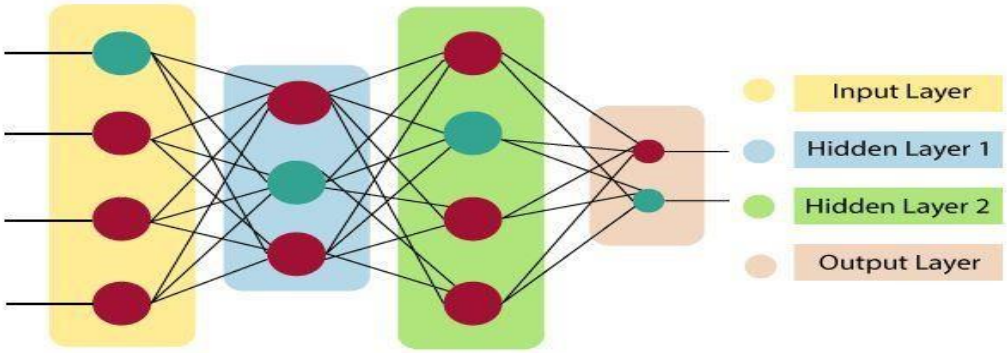


Figure 4.2: Neural Network

(i)Input Layer: As the name suggests, it accepts inputs in several different formats provided by the programmer.

(ii)Hidden Layer: The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

(iii)Output Layer: The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

Below is the formula for computing the weighted sum being discussed.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

4.3.1 How ANN Work

The ideal way to visualize an artificial neural network is as a weighted directed graph, where the nodes are the artificial neurons. The directed edges with weights represent the relationship between the neuron inputs and outputs. The input signal for the artificial neural network comes from an external source as a pattern and an image as a vector. Then, for each n-th input, these inputs are mathematically assigned using the notation x(n).Each input is then multiplied by the weights that correspond to it (these weights are the information that the artificial neural networks use to solve a particular problem). In the artificial neural network, these weights often indicate how well neurons are connected to one another. Inside the computing unit, a summary of each weighted input is created.The output is made non-zero by adding bias if the weighted total is equal to zero, or else something else is added to scale up the output to the system's reaction. The input for bias is the same, and the weight is 1. The sum of the weighted inputs in this case can range from 0 to positive infinity. Here, a specific maximum value is benchmarked to keep the response within the bounds of the desired value, and the sum of the weighted inputs is fed through the activation function.The set of transfer functions utilised to produce the desired output is referred to as the activation function. A variety of activation functions exist, although they are mainly either linear or non-linear sets of functions. The Binary, linear, and Tan hyperbolic sigmoidal activation function sets are a few of the often employed sets of activation functions.Binary: The output of a binary activation function is either a one or a zero. Here, a threshold value has been established in order to achieve this. The final output of the activation function is returned as one or 0 depending on whether the net weighted input of neurons is

greater than 1.Sigmoidal Hyperbolic: The Sigmoidal Hyperbola function is generally seen as an "S" shaped curve. Here the tan hyperbolic function is used to approximate output from the actual net input. The function is defined as: $F(x) = (1/1 + \exp(-x))$

4.4 Convolutional Neural Network (CNN)

The overall design of a typical CNN is made up of a number of different layers of various types. Each input is sent through several Convolutional layers that include filters, pooling layers, fully connected layers, and an output layer that provides a probabilistic value to support the classification. Decisions must be made regarding both architectural patterns, such as the number of convolution and pooling layers, input data format, filter dimension, etc., and hyper parameters, such as learning rate, dropout probability, number of epochs, batch size, etc., during the training of a CNN.Because of its extraction and classification components, CNN typically creates good classifiers and performs well with classification tasks. Our suggested model is a sequential model with two Conv2D layers, three dense layers, and the output layer as the last layer. By moving a filter's window over the shape of the input, performing a matrix multiplication, and then supplying the stored result into a feature map, our suggested model operates.

4.4.1 Convolution Layer

For feature detection, both convolutional layers are employed. The input shape is sent to the first Conv2D layer, which has 64 filters, a kernel size of 5, and a stride of 1. The filter parameter indicates the node numbers in each layer. In this scenario, the size of each layer in our model will rise from 64 to 128. The option controls the size of the kernel window, which in our case is 5. This size produces a filter matrix of 5x5. Stride monitors the filter's operation throughout the input range. The filter moves one unit at a time as it converges around the input volume in the first layer, where stride is set to 1. The output of this convolutional layer with "same padding" has the same height and weight as the input. ReLU is the name of the activation function we used for this layer; it has a number of advantages over conventional units, including effective gradient propagation and quicker computation than sigmoid units, maintaining, despite their simplicity, adequate discriminatory characteristics [15]. The MaxPooling2D layer, which is used to reduce the dimension of the input shape supplied through the convolutional layer, comes after the initial Conv2D Layer.

4.4.2 Flatten Layer

The flatten layer converts the output of the convolutional layers into a onedimensional array to be inputted into the next hidden Layer.

4.4.3 Fully Connected Layer

The model has been further processed using two dense layers (Fully Connected Layers) and an output layer. The first two dense levels of the following three layers will have 256 and 512 nodes, respectively. Dense layers refer to a linear process that uses a weight to connect each output to each input as it passes through the layers. The first two thick layers employ ReLU, a nonlinear function that merely resets all of the negative activations to zero. This has been shown to increase the model's nonlinear features while having no influence on the fields of the overall network.

4.4.4 Output Layer

The output layer having activation function SoftMax will consist of 10 nodes in our model that refers to the possible classification numbers [43]. This added constraint allows converging quicker than it would otherwise. The model then predicts the choice with the highest probability. Below figure shows the graphical representation of the CNN just described.

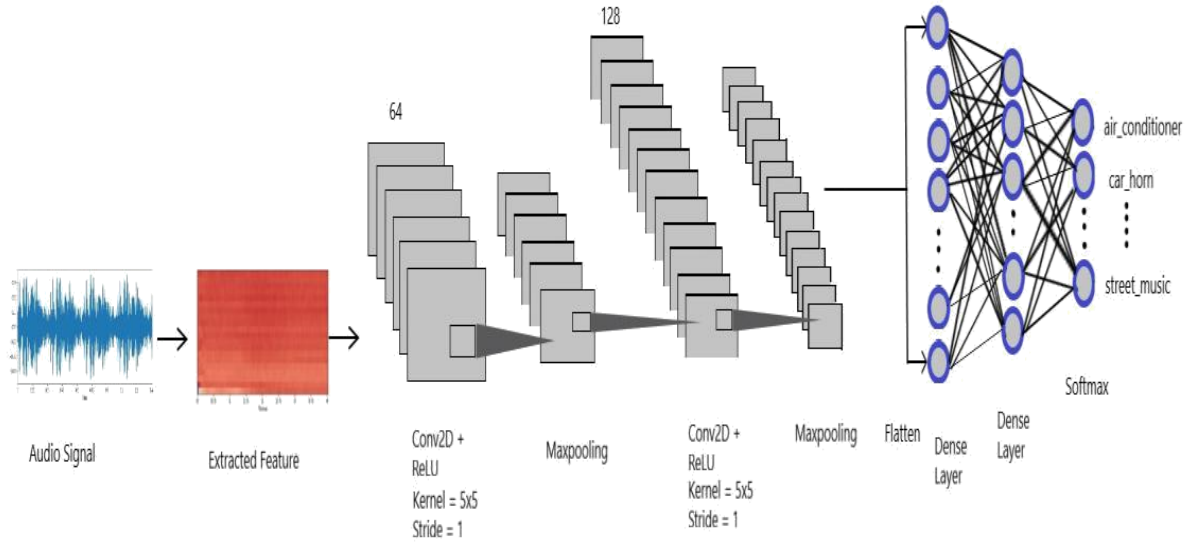


Figure 4.3: Architecture of our Convolutional Neural Network

CHAPTER 5 - FEATURE EXTRACTION

This chapter describes the techniques used for feature extraction in our problem statement.

5.1 MFCC

A representation of a sound's short-term power spectrum used in sound processing is called a mel-frequency cepstrum (MFC), which is based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.An MFC is made up of a number of coefficients known as mel-frequency cepstral coefficients (MFCCs). They are derived from a nonlinear "spectrum-of-a-spectrum" cepstral representation of the audio sample. The mel-frequency cepstrum (MFC) differs from the cepstrum in that the frequency bands are evenly spaced on the mel scale, which more closely resembles the response of the human auditory system than the linearly spaced frequency bands used in the conventional spectrum. When used in audio compression, for instance, this frequency warping can improve the representation of sound and potentially lower the transmission bandwidth and storage needs of audio signals. Below figure shows the steps involved in computing the MFCCs for a certain audio signal.

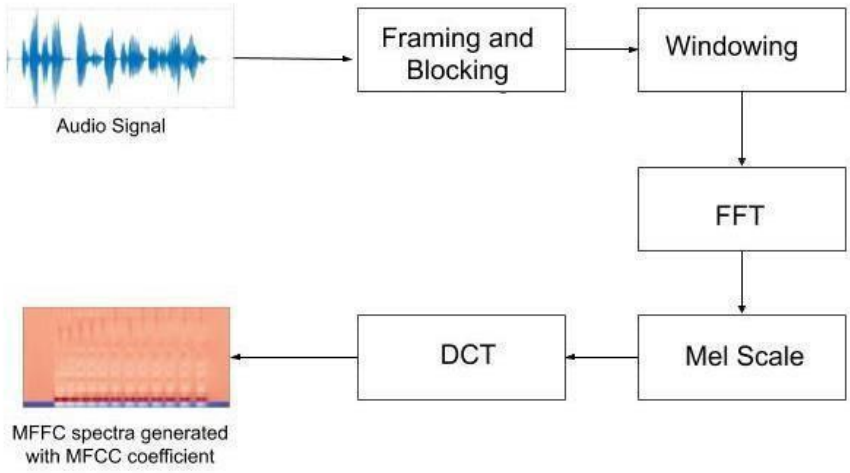


Figure 5.1: Steps to extract MFCCs from an audio signal

Mel-frequency bands are equally dispersed in MFCC and are quite comparable to the human voice system; as a result, MFCC can be effectively utilised to characterise speakers, for example, by identifying the speaker's mobile phone type and other Data. The manufacture of electronic components in a phone has tolerances since various electronic circuit realisations do not have exactly the same transfer functions, which is relevant when discussing voice recognition to identify mobile phones. If the task performing circuits come from various manufacturers, the differences in the transfer function from one realisation to another become more obvious. As a result, each mobile phone adds a convolutional distortion to the speech that is input, which has a distinct effect on the recordings made using the cell phone. As a result, by multiplying the original frequency spectrum with a further multiplication of the transfer function unique to each phone, followed by signal processing techniques, a single phone can be recognized from the recorded voice. So, by characterising cell phone recordings, one can determine the brand and model of the phone.

5.2 Mel-Spectrogram

Mel scale and a spectrogram are combined to create a Mel Spectrogram. where the frequency scale's non-linear transformation is represented by the mel scale. As previously mentioned, this is accomplished using overlapping triangular filters. The procedures needed to produce Mel specs and MFCC coefficients are somewhat similar. The audio signal is first divided into smaller frames, and each frame is then given a hamming window. Following that, DFT is used to transition from the time domain to the frequency domain. Triangular filters are used to extract the frequency bands in the Mel filter bank step. In order to create the spectrum, which includes the spectral envelope and spectral features, log() is used at the very end. This aids in creating the mel-spectrogram. Below figure shows the steps involved in computing the Mel Spectrogram for a certain audio signal.

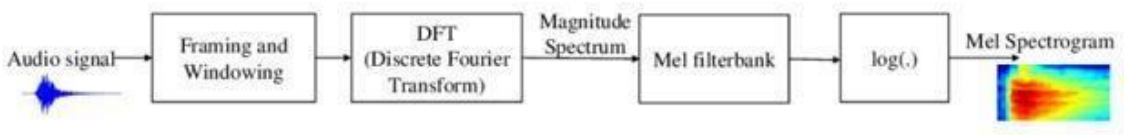


Figure 5.2: Steps required to extract Mel spectrogram.

CHAPTER 6 - MODEL ARCHITECTURE

This chapter deals with the main part of the code-creating the models.

6.1 Using ANN

It starts by importing the necessary libraries such as os, numpy, pandas, and matplotlib. Then, it imports IPython.display, librosa, and tqdm libraries to display audio signals and progress bars respectively.The next step involves loading the UrbanSound8K dataset and setting the audio dataset path. The feature_extractor() function is then defined and used to extract features from the audio files in the dataset. These extracted features are then stored in a pandas dataframe for further processing.

The features and class labels are then split into training and testing datasets using the train_test_split() function from the sklearn library. A label encoder is also used to transform the class labels into categorical data. The neural network model is then defined using the Sequential model from the tensorflow.keras.models library. This model has four layers, each with a different number of neurons and dropout rate.The model is then compiled using categorical cross-entropy loss, accuracy metrics, and the Adam optimizer. The fit() function is used to train the model with the training dataset, and the ModelCheckpoint function from the tensorflow.keras.callbacks library is used to save the best model. After the model is trained, the testing dataset is evaluated for accuracy using the evaluate() function. A sample audio file is then loaded, and its features are extracted and used to predict the class label using the trained model. Finally, the predicted class label is displayed using the label encoder object.

6.2 Using CNN with TensorFlow

This is a Python code that trains a Convolutional Neural Network (CNN) model to classify audio files into different sound categories using the UrbanSound8K Dataset. Here's a breakdown of the code:

a) The first few lines import the required libraries including NumPy, pandas, matplotlib, IPython.display, librosa, sklearn, and tensorflow.
b) The metadata DataFrame is created by loading the UrbanSound8K.csv file. This file contains metadata for each audio file, including the file path, class label, fold number, and so on.
c) The audio_dataset_path variable contains the path to the folder that contains all the audio files.
d) The extract_features() function takes an audio file, its sample rate, and its class ID as input, and returns the Mel Frequency Cepstral Coefficients (MFCCs) of the audio file. The MFCCs are a commonly used feature representation for audio signals.
e) The features list is initialized, and a loop is run through each row of the metadata DataFrame. For each row, the file path is constructed, and the corresponding audio file is loaded using librosa.load(). The extract_features() function is then called to extract the MFCCs of the audio file, and the resulting feature and class label are appended to the features list.
f) The labels and features arrays are created from the features list. The LabelEncoder from the sklearn.preprocessing module is used to encode the class labels as integers, and the to_categorical() function from tensorflow.keras.utils is used to one-hot encode the class labels.
g) The train_test_split() function from sklearn.model_selection is used to split the data into training and testing sets.
h) The Conv1D model from tensorflow.keras.layers is used to define the CNN model architecture. This model consists of three convolutional layers, a max-pooling layer, two dropout layers, and two dense layers. The model is compiled using the Adam optimizer and the categorical_crossentropy loss function.
i) The fit() method is called to train the model using the training data. The batch_size and epochs hyperparameters are set to 32 and 50 respectively.
j) After the model has been trained, the evaluate() method is called to evaluate the model on the testing data. The test loss and accuracy are printed.

CHAPTER 7 - RESULT OVERVIEW

This chapter deals with comparing and analyzing the results obtained during the course of our model creation.

7.1 Result Overview

Due to its size and reduced number of classes, the UrbanSound8K dataset facilitates the classification of audio. We have thoroughly tested our dataset on the given model to demonstrate the value of feature stacking, augmentation, and model size. Our strategies for the research experiment included locating features that yield high accuracy for both models and operate well. The MFCC was the top feature that demonstrated the most accuracy for both models in such settings when compared to other Features. Our second method of conducting research experiments was stacking several features to see how the models' categorization accuracy changed as a result. On the other hand, stacking features raises the cost of computation, but some of the stacking strategy hasn't actually had the impact on performance that was initially anticipated. This is due to the fact that there is a significant disparity in the array values and that the spectrograms produced after stacking do not exhibit a consistent pattern for each class. The best performance to date was 97.52%, however some of the stacking greatly improved the performance of our model, enabling us to achieve a validation accuracy of 98.8%.We also show that CNN work well with the PyTorch as compared to TensorFlow. While using ANN we could not get required accuracy using TensorFlow due to the computational power and resources available.

7.2 Similarity and Difference Between CNN and ANN

We also show that CNN work well with the PyTorch as compare to TensorFlow. While using ANN we could not get required accuracy using TensorFlow due to the computational power and resources available. For the majority of its computations, ANN relies on weights and an activation function. The best way to explain how ANN functions is to say that it recreates the neural network of the brain artificially. As a human would, it goes back and "changes" its thinking after making a mistake. Rows of data points housed by neurons that are all connected by the same neural network are referred to as "layers" in an ANN. Weights are used by ANN to learn. After each iteration through the neuron in an ANN, weights are modified. ANN

adjusts the weights in the past based on the accuracy determined by a "cost function". In contrast, CNN does not use weights or neurons. Instead, CNN uses filtration to examine image inputs and applies multiple layers to images. The arithmetic layer, corrected linear unit layer, and fully connected layer are these layers. These layers' tasks include processing data inputs, comprehending patterns that the network can "see," and producing an n-dimensional vector. This n-dimensional output is used to identify distinctive features and link them to the supplied image input. The user can then receive the output of the classification. Despite their differences, both approaches produce epochs to evaluate the efficacy of the models created and use measures of the error to improve learning.

Finally, there are specific situations in which ANN might be preferred over CNN, and vice versa. Since their mathematical processes differ from one another, they are each more adept at solving problems. CNN is typically a more effective and precise method of handling categorization issues. For issues with small datasets and no requirement for image inputs, ANN remains dominating. Below figure represents ANN model with TensorFlow module. It provides 74% accuracy in test data.

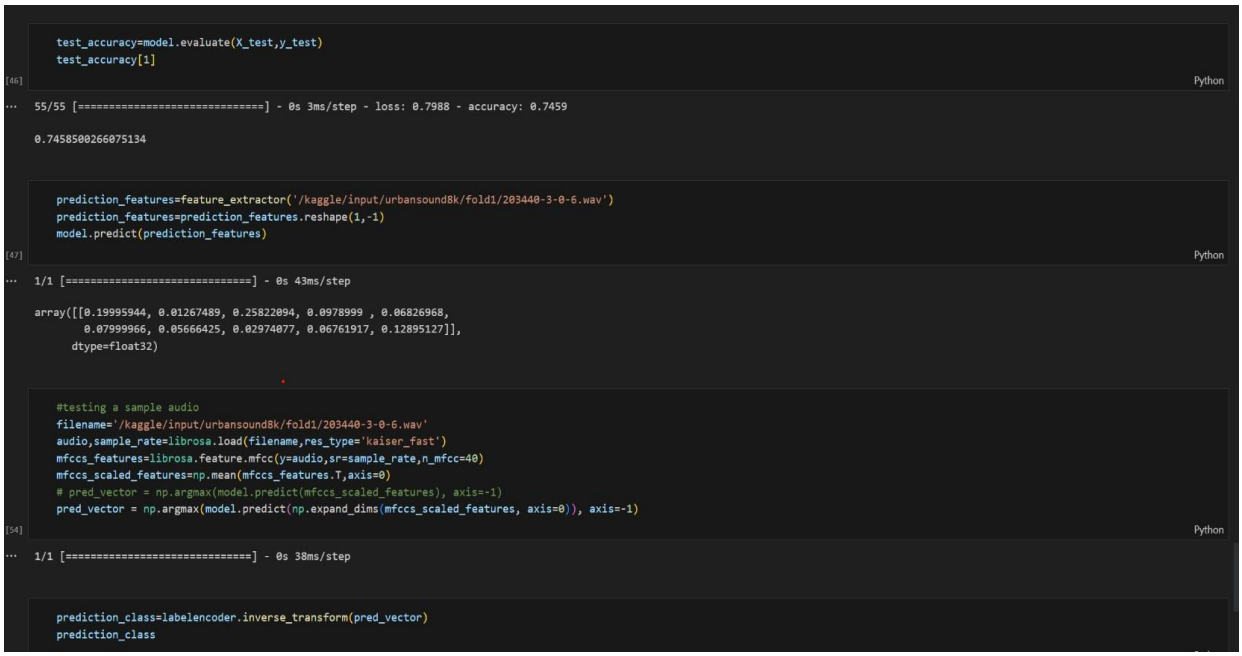


Figure 7.1: ANN Accuracy

Below given figure represents CNN model with Tensorflow module. It provides 91% accuracy in test data.

Figure 7.2: CNN Accuracy

CHAPTER 8 - CONCLUSION

This chapter details about the conclusion of our project and its future scope.

8.1 CONCLUSION

After working on ANN and CNN using TensorFlow, we came to the conclusion that CNN using TensorFlow had the highest accuracy we could get. Also, we tried PyTorch as an alternative. We found PyTorch as not suitable for audio classification as it could not provide us the required accuracy.

8.2 FUTURE SCOPE

Our project though still has room for improvement, as the code in pytorch was just a basic implementation of how pytorch can be used, we can still optimize it to a much higher extent, this is because pytorch provides the code which has good degree of reusability due to the implementation of object-class structures, and thus can prove to be an effective and modular code if we take care of optimization through improving the accuracy of the model.

REFERENCES

- Basic Python. (2023). [Online]. Available: https://www.w3schools.com/python
- Introduction to ML. (2023). [Online]. Available: https://www.geeksforgeeks.org/introduction-machine-learning/
- Supervised Machine Learning. (2023). [Online]. Available: https://www.javatpoint.com/supervised-machine-learning
- Unsupervised Machine Learning. (2023). [Online]. Available: https://www.javatpoint.com/unsupervised-machine-learning
- Classification. (2023). [Online]. Available: https://www.javatpoint.com/classification-algorithm-in-machine-learning
- Regression. (2023). [Online]. Available: https://www.javatpoint.com/regression-analysis-in-machine-learning
- Linear Regression. (2023). [Online]. Available: https://www.javatpoint.com/linear-regression-in-machine-learning
- Logistic Regression. (2023). [Online]. Available: https://www.javatpoint.com/logistic-regression-in-machine-learning
- Naive-bayes-classifier. (2023). [Online]. Available: https://www.javatpoint.com/machine-learning-naive-bayes-classifier
- Krish Naik's Complete Deep Learning Playlist. (2023). [Online]. Available: https://www.youtube.com/playlist?list=PLZoTAELRMXVPGU70ZGsckrMdr0FteeRUi
- Chris Filo, "UrbanSound8K."(2020),Hosted on Kaggle. https://www.kaggle.com/datasets/chrisfilo/urbansound8k?datasetId=500970
- Gradient Descent. (2023). [Online]. Available: https://towardsdatascience.com/implementing-gradient-descent-in-python-fromscratch-760a8556c31f
- ChatGPT. (2023). [Online]. Available: https://openai.com/blog/chatgpt

- MLP Neural Network. (2023). [Online]. Available: https://machinelearninggeek.com/multi-layer-perceptron-neural-network-usingpython

- Artificial Neural Network. (2023). [Online]. Available: https://www.javatpoint.com/artificial-neural-network

- Overview of hyperparameter tuning. (2023). [Online]. Available: https://cloud.google.com/ai-platform/training/docs/hyperparameter-tuningoverview

- Easy Hyperparameter Tuning with Keras Tuner and TensorFlow. (2023). [Online]. Available: https://pyimagesearch.com/2021/06/07/easy-hyperparameter-tuning-with-kerastuner-and-tensorflow/

- Correlational Research Design [Examples, Types, Advantages, Disadvantages, Characteristics]. (2023). [Online]. Available: https://t4tutorials.com/correlational-research-design-examples-types-advantagesdisadvantages-characteristics/

- RNN guide. (2023). [Online]. Available: https://www.tensorflow.org/guide/keras/rnn

- CNN algorithm for MNIST dataset. (2023). [Online]. Available: https://github.com/Crisp3333/cnn-algorithm

- CNN in machine learning. (2023). [Online]. Available: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machinelearning/

- Image classification using CNN (CIFAR10 dataset) | Deep Learning Tutorial 24 (Tensorflow & Python). (2023). [Online]. Available: https://youtu.be/7HPwo4wnJeA

- How to make Jupyter Notebook run on GPU. (2023). [Online]. Available: https://stackoverflow.com/questions/51002045/how-to-make-jupyter-notebook-torun-on-gpu

- Google Colab (Online Jupyter Notebook with free GPU). (2023). [Online]. Available: https://colab.research.google.com/

- Gradient Descent algorithm and its variants. (2023). [Online]. Available: https://geeksforgeeks.org/gradient-descent-algorithm-and-its-variants/

- Radial Basis Function Neural Network Simplified. (2023). [Online]. Available: https://towardsdatascience.com/radial-basis-function-neural-network-simplified6f26e3d5e04d

- Librosa(Python Library). (2023). [Online]. Available: https://librosa.org/

- Innovation in Augmented Listening Technology. (2023). [Online]. Available: https://publish.illinois.edu/augmentedlistening/tutorials/music-processing/tutorial1-introduction-to-audio-processing-in-python/

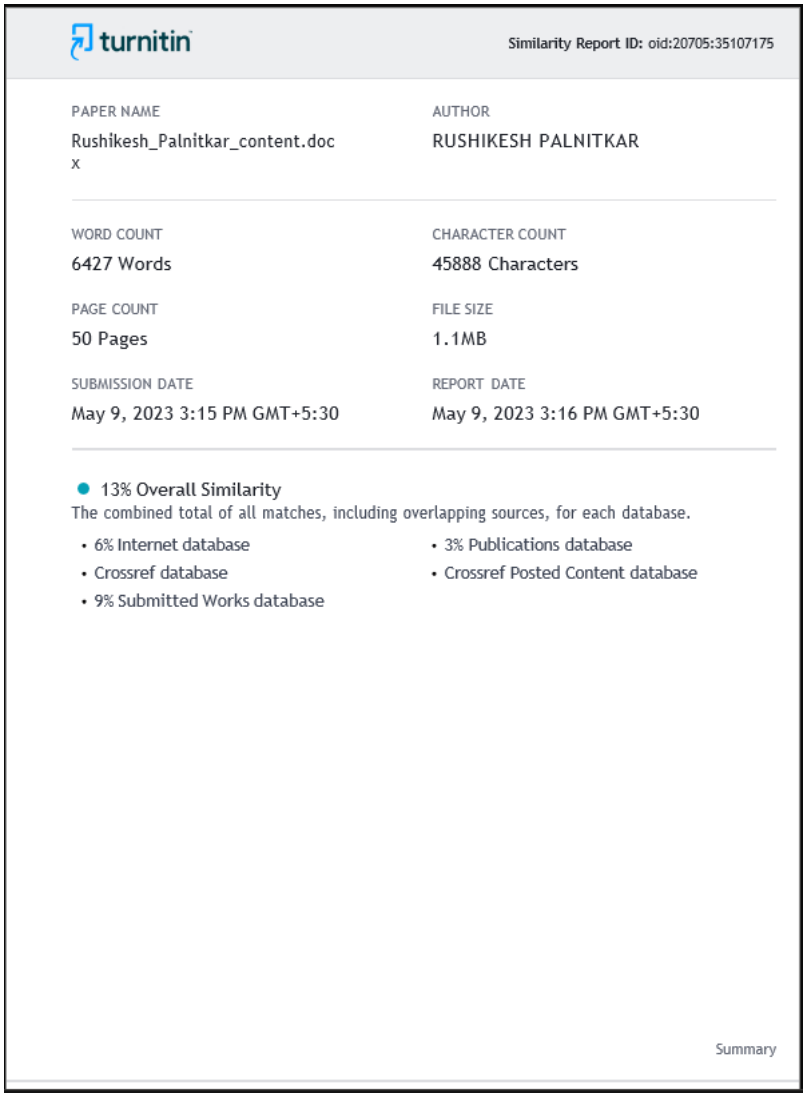
- Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. (2017). [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7829341

- Valerio Velardo's Channel. (2023). [Online]. Available: https://www.youtube.com/playlist?list=PLwATfeyAMNoirN4idjev6aRu8ISZYVWm

- BISAG-N Website. (2023). [Online]. Available: https://apps.bisag.co.in/ or https://bisag-n.gov.in/

- Wikipedia. (2023). [Online]. Available: https://en.wikipedia.org/wiki/Bhaskaracharya_Institute_For_Space_Applications_and_Geo-Informatics

PLAGIARISM CERTIFICATE



turnitin

Similarity Report ID: oid:20705:35107175

PAPER NAME
Rushikesh_Palnitkar_content.docx

AUTHOR
RUSHIKESH PALNITKAR

WORD COUNT
6427 Words

CHARACTER COUNT
45888 Characters

PAGE COUNT
50 Pages

FILE SIZE
1.1MB

SUBMISSION DATE
May 9, 2023 3:15 PM GMT+5:30

REPORT DATE
May 9, 2023 3:16 PM GMT+5:30

13% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- Crossref database
- 9% Submitted Works database
- 3% Publications database
- Crossref Posted Content database

Summary

Plagiarism Report