Final Project Report

Digital Audio Forensics Using Artificial Intelligence

Project Supervisor

Dr. Sonia Salman

Submitted By

F230218F24

Ahmad Aman BC200401244

Software Projects & Research Section,
Department of Computer Sciences,
Virtual University of Pakistan

CERTIFICATE

This is to certify that Ahmad Aman (BC200401244) has worked on and completed their Software Project at Software & Research Projects Section, Department of Computer Sciences, Virtual University of Pakistan in partial fulfillment of the requirement for the degree of BS in Computer Sciences under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfills the requirements of BS in Computer Sciences.

Supervisor / Internal Examiner

Dr. Sonia Salman

Supervisor,

Software Projects & Research Section,

Department of Computer Sciences

Virtual University of Pakistan

_____

(Signature)

External Examiner/Subject Specialist

<<External Supervisor Name>>

_____

(Signature)

Accepted By:

_____

(For office use)

EXORDIUM

In the name of Allah, the Compassionate, the Merciful.

Praise be to Allah, Lord of Creation,
The Compassionate, the Merciful,
King of Judgment-day!

You alone we worship, and to You alone we pray for help,
Guide us to the straight path

The path of those who You have favored,

Not of those who have incurred Your wrath,
Nor of those who have gone astray.

DEDICATION

"To my parents, who have always believed in me."

ACKNOWLEDGEMENT

I am very grateful to my supervisor **Dr. Sonia Salman** for her guidance and technical support. She always encouraged me to discuss freely any problem during the whole process of project development. I pay special tributes to her very kindness by giving me a chance to improve upon my work in the light of her valuable feedback.

Further I want to thank my elder sister Ameema Ilyas for connecting me with industry professionals whose advice steered me in the right direction. I am thankful to my friend Zain Ateeq who provided me with good educational resources. His professional experience in the field of machine learning (ML) enabled me to select the most suitable neural network model. Insightful conversations with my friend Muhammad Abubakar helped me to see the project from different perspectives. Furthermore, I want to thank my sister Fareeha Aman and brother Abdullah Aman for assisting me in creating the dataset used for training. I also acknowledge the previous work done in this domain which provided a solid foundation for my own work.

In the end, I am thankful to my father and mother for their unending encouragement, support, and prayers.

PREFACE

In recent years, the rapid advancement of artificial intelligence and machine learning has brought about significant changes in various fields, including digital audio forensics. The spread of AI-generated speech has posed new challenges for the detection and prevention of audio fraud, which has profound implications for security, legal, politics, and media industries.

This project, titled "Digital Audio Forensics Using AI/ML," is the culmination of my final year efforts at Virtual University, Pakistan. The primary objective of this project is to develop and train a neural network model capable of detecting fake AI-generated speech with high accuracy. Through extensive research, data collection, and model training, I have endeavored to create a robust solution that can differentiate between authentic and synthetic audio.

The journey of developing this project has been both challenging and rewarding. I was inspired to explore this topic due to my interest in the field of deep learning and audio engineering. Audio signals or any time series signals have always been fascinating to me because they are a marvelous example of complexity arising from simplicity. This project is my attempt to tackle both the world of deep learning and signal processing to make something useful.

Table of Contents

CHAPTER NO. 1: Gathering and Analyzing Info.

1.1 Introduction

The first phase of any machine learning or any software engineering project in general is information gathering and analyzing that information to figure out the functional and non-functional requirements of the software. In the case of this specific project, this task entails gathering information about the current state of research in the domain of detecting deepfake audios and figuring out the fidelity of the current deepfake audio generation models. The better the model and consequently the quality of the generated audio, the harder it is to detect it. This can be a bit of a cat and mouse chase where one thing is always trying to beat the other.

The other most important thing during information gathering is deciding which dataset to use for the model training. Afterall, the model cannot infer anything if it is not fed the right amount and right quality of data.

1.2 Purpose

The main purpose of this phase is to understand the scope and hence chart out the requirements of the project. What it can and can't do and what are the technical specifications that are essential for the project to work as required. This whole phenomenon is the process of gathering software requirements.

1.3 Scope

1.3.1 Project Title:

"Digital Audio Forensics Using Artificial Intelligence"

1.3.2 Project Justification:

The project aims to leverage the intersection of audio engineering, signal processing, and artificial intelligence to develop a digital audio forensics application. This application will analyze and verify the authenticity of audio recordings by applying AI approaches to differentiate between authentic and modified/fake audio samples.

1.3.3 Project Objectives/Features:

- Implement a neural network model for processing audio clips using feature extraction methods and machine learning algorithms.
- Enable users to initiate the detection process and receive clear results indicating the authenticity of the audio clip.
- Provide users with the option to offer feedback for continuous system improvement.

1.3.4 Constraints:

- The system should process audio clips within a reasonable time frame.

1.3.5 Deliverables:

- Neural network model for audio analysis.
- Integration with specified tools i.e. different ML models.
- Detection results for users.

1.4 Definitions, acronyms, and abbreviations

The following definitions and acronyms are important regarding the domain knowledge of this project:

- Machine Learning (ML): This is the process of statistical learning where some inputs along with their outputs are given to a system and the system then tries to predict any unknown outputs given new inputs.
- Artificial intelligence (AI): This is a broader field of computer science where the goal is to make machines that can reason like human beings.
- Deep Learning (DL): This is a subset of machine learning where deep neural networks are used to act as the function estimators.
- Neural network (NN): A neural network is a collection of neurons that take an input vector of certain length and output another vector.
- Artificial Neuron (AN): This is basically a function that takes a certain number of inputs and their respective weights and then outputs the sum of the product of all the inputs with their respective weights. The weighted sum is calculated as:

$$z = \sum_{i=1}^{n} (w_i \times x_i) + b$$
$$where, x_i = i^{th}\ input, w_i = it^{th}\ weight, and\ b = bias\ of\ the\ nueron$$

- Activation Function (AF): Every neuron has an activation function whose output is the final output of that neuron. These are introduced so that the neuron can even fit a curve that is non-linear. The most used activation function is the ReLu.
- Rectified Linear unit (ReLu): This is a neuron activation function. It just clamps the input that is less than a certain value to be zero and after that it acts as linear function. (Kinsley & Kukieła)

$$f(x) = \max(0, x)$$

1.5 Project requirements

1.5.1 Functional Requirements

Upload Audio Clips:

Users can upload audio clips to the application interface.

Initiate Detection:

Users can trigger the neural network model to initiate the detection process for uploaded audio clips.

Receive Audio Clips:

The neural network model receives audio clips uploaded by users for analysis.

Preprocessing:

Implement preprocessing modules to clean and standardize audio data, including noise reduction, normalization, and feature extraction. The model processes audio clips using feature extraction methods and machine learning algorithms.

Generate Detection Results:

The model generates accurate detection results, indicating whether the provided audio clip is real or fake.

1.5.2 Non-Functional Requirements

Collecting and cleaning data for ML model training:

Collect audio data for the training and tuning of the machine learning model. This includes cleaning the data and splitting it into training (70%) and testing (30%) set.

Training and tuning ML model:

Choosing and training a machine learning model that best suits the needs of the project.

Accuracy and Precision:

Define acceptable levels of accuracy and precision for detecting manipulated audio to meet the project's objectives.

Scalability:

Ensure the system can scale to handle large datasets and real-time processing demands.

Usability:

Provide a standard function to just run the model inference for testing individual files.

Performance:

The system gives result in as short amount of time as possible.

Reliability:

Define reliability requirements to ensure the system performs consistently and accurately over time.

Ethical Considerations:

Incorporate ethical guidelines in the development and use of the system, addressing issues such as privacy, bias, and transparency.

1.6 Use cases and usage scenarios

1.6.1 Use Case Diagram

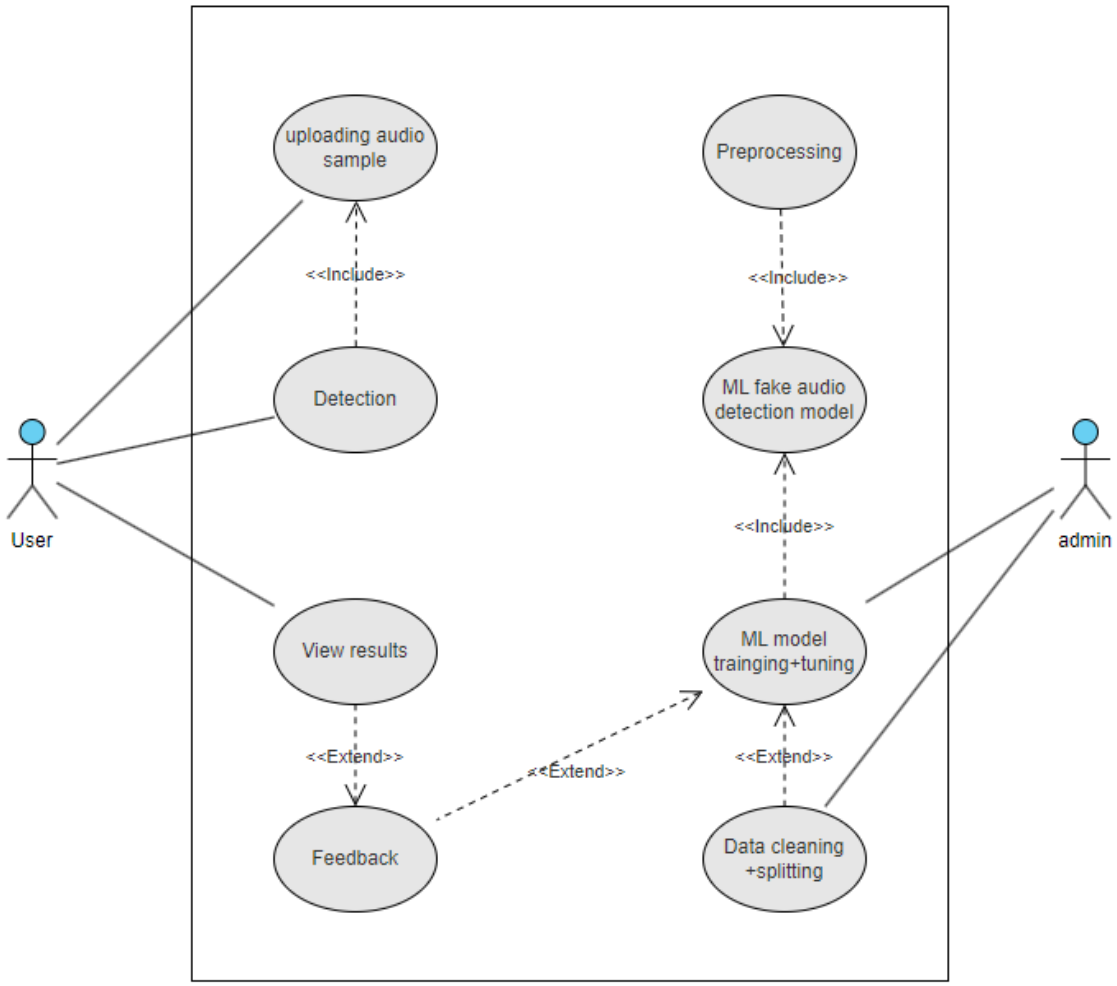

Figure 1 Use Case Diagram

1.6.2 Usage Scenarios

Use Case title	Use Case ID	Actions	Description	Alternative Paths	Preconditions	Post Conditions	Author	Exceptions

uploading audio sample	01	The user uploads the audio sample	The system receives the audio sample provided by the user	None	File is in valid format	Audio clip is successfully uploaded	BC200401244	File format is invalid
Detection	02	The user initiates the detection process	The system starts the detection of audio clip	None	Audio clip is successfully uploaded	Detection process is initiated	BC200401244	No clip is uploaded
View results	03	User clicks the result	The system displays the result of detection	None	Detection process is initiated	User gives feedback	BC200401244	Detection process has not started yet.
Feedback	04	User gives feedback	The system stores user feedback and tunes models weight accordingly	None	User gives feedback	Feedback is moved to ML model	BC200401244	No feedback given
Preprocessing	05	System processes the uploaded clip	System preprocesses the audio clip before sending for detection	None	Clip is uploaded and is valid	The clip is cleaned of any noise	BC200401244	Clip is invalid
ML fake audio detection model	06	The ML model makes prediction	The ML model makes prediction based on the training data	None	Clip has been preprocessed	Clip is sent for result display	BC200401244	Clip has too much noise
Data cleaning + splitting	07	Data is collected	Data of different audio samples is compiled and cleaned	None	Sample data is bias less	Data is ready for training and testing	BC200401244	Data has too much bias and noise

ML model training + tuning	08	ML model is trained	ML model is trained on the 70% of the data set and then tested against the rest 30%	None	Data is clean and organized	Model is ready for deployment	BC200401244	Data is not properly organized

1.7 Development Methodology

1.7.1 Chosen Methodology

For this project the adopted model of software development is called **VU Process Model**. This model is a combination of the Waterfall Method and the Spiral model. Here is the detailed explanation of these models:

Waterfall Method:

The Waterfall model is a linear and sequential approach to software development. It consists of distinct phases that are completed one after the other without revisiting the previous stages. The typical phases are:

Requirements:

Gather and document all requirements from stakeholders which includes the users and anyone else that has any stake in the project. The requirements are both functional and non-functional.

Design:

Create a detailed system design based on the requirements.

Implementation:

Develop the actual system or software based on the design. This also includes the development of a prototype that can be iterated upon to further improve the overall product.

Testing:

Conduct various testing phases (unit testing, integration testing, system testing) to ensure the product meets the specified requirements.

Deployment:

Deploy the product to the target environment.

Maintenance:

Provide ongoing maintenance and support for the system.

Spiral Method:

The Spiral model is an iterative and risk-driven model that combines the idea of iterative development with the systematic aspects of the Waterfall model. It involves repeating cycles, called spirals, each of which represents a phase in the software development life cycle:

Planning:

Identify objectives, alternatives, and constraints.

Risk Analysis:

Evaluate risks and develop strategies to manage them.

Development:

Develop the product incrementally.

Evaluation:

Review the results of each iteration and decide whether to proceed to the next iteration and the cycle repeats.

VU Process Model:

This model is a combination of both the waterfall method and the spiral method and hence includes the step-by-step approach to planning and requirement gathering and an iterative cyclical approach for the development of the project.

1.7.2 Reasons for choosing the methodology.

This process model is adopted for this project because the project is divided into four phases and the development phase needs a spiral like approach to compensate for any changing scenarios or constraints faced during the development phase. Any machine learning project is more akin to a research project wherein as you progress towards the goal, the path keeps changing and sometimes even the goal can be changed due to certain limitations. This is why such a methodology has been selected for this project.

1.7.3 Work Plan (Gantt Chart)

Following is the Gannt chart that lays out the schedule of this project:

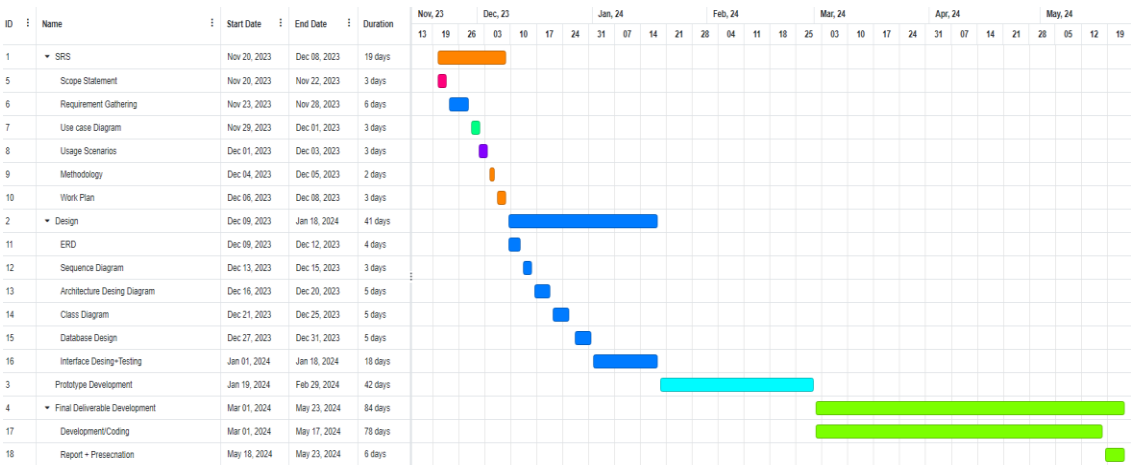

Figure 2 Gantt Chart

1.7.4 Project Schedule (Submission Calendar)

Deliverable	Start Date	End Date
SRS Document	Mon 20 Nov, 2023	Fri 08 Dec, 2023
Design Document	Sat 09 Dec, 2023	Thu 18 Jan, 2024
Prototype Phase	Fri 19 Jan, 2024	Thu 29 Feb, 2024
Final Deliverable	Fri 01 Mar, 2024	Thu 23 May, 2024

CHAPTER NO. 2: Designing the project.

2.1 Introduction

During the design phase of the project different technical aspects of the project were decided upon including but not limited to the neural network model of choice, the choice of the dataset and different tools that will be used during the development of the project.

2.2 Purpose

The main purpose of going through the design before the active development of the project is just planning so that the development process can be streamlined, and any incoming hiccups and bugs can be foreseen before they occur.

2.3 Architecture Diagram (Neural Network Model):

The neural network model that has been chosen for this project is a convolution neural network (CNN) based model. The model has four 2D convolution layers each followed by a pooling layer. In the end it has two fully connected layers. This model is thanks to the research of (Huang, Zhang, Tiovalen, & Balaji, n.d.) Here's a high-level architectural diagram of the model:

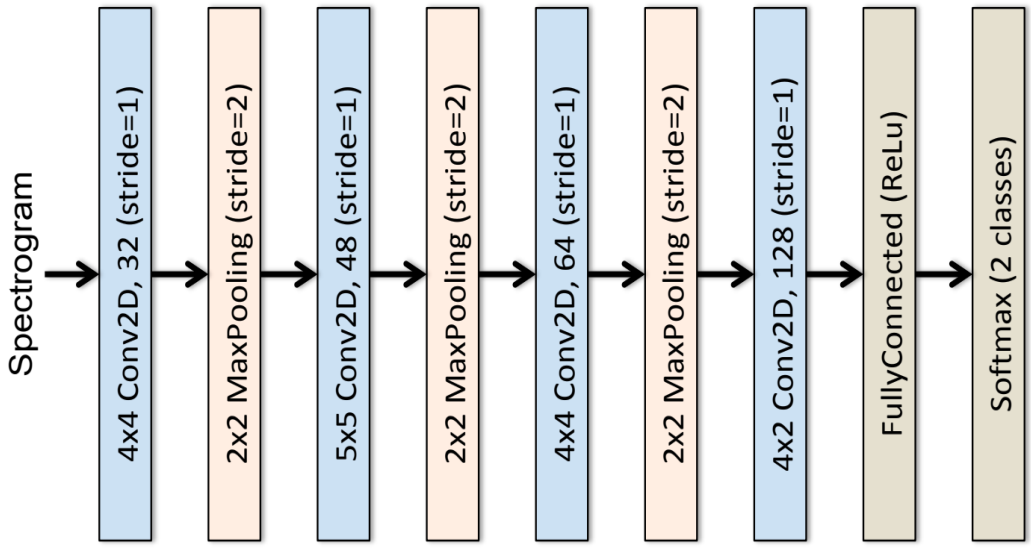

Figure 3 Neural Network High level overview. (Lieto, et al., 2019)

2.4 Class Diagram

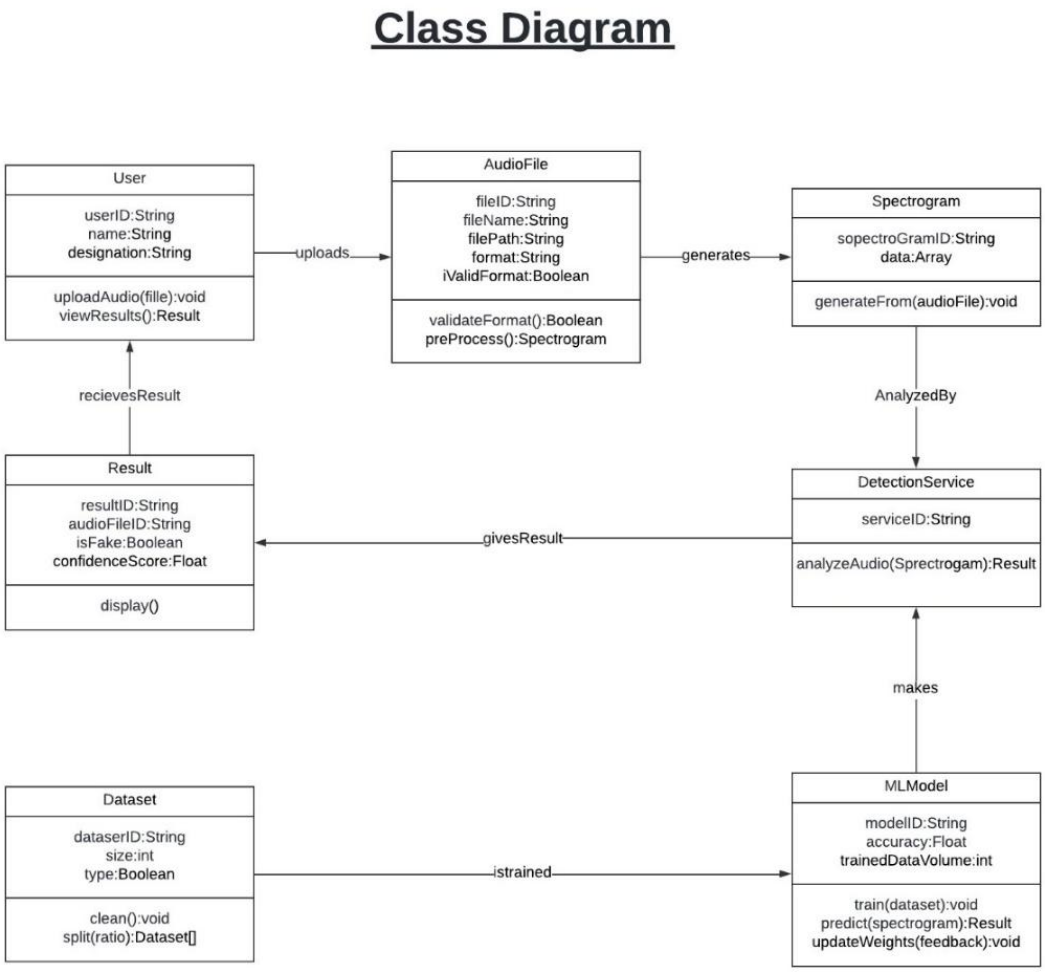

Figure 4 Class Diagram

2.5 Entity Relationship Diagram (ERD):

Following is the entity relationship diagram for this project. As this is a machine learning project and there is no database design involved here. Hence, the essence of an ERD has been adopted wherein entities are the basic logical unit of our machine learning project for instance audio sample or feature-set etc. Attributes are the different data we must deal with within these entities and lastly their relationship to each other has been shown using standard notation.

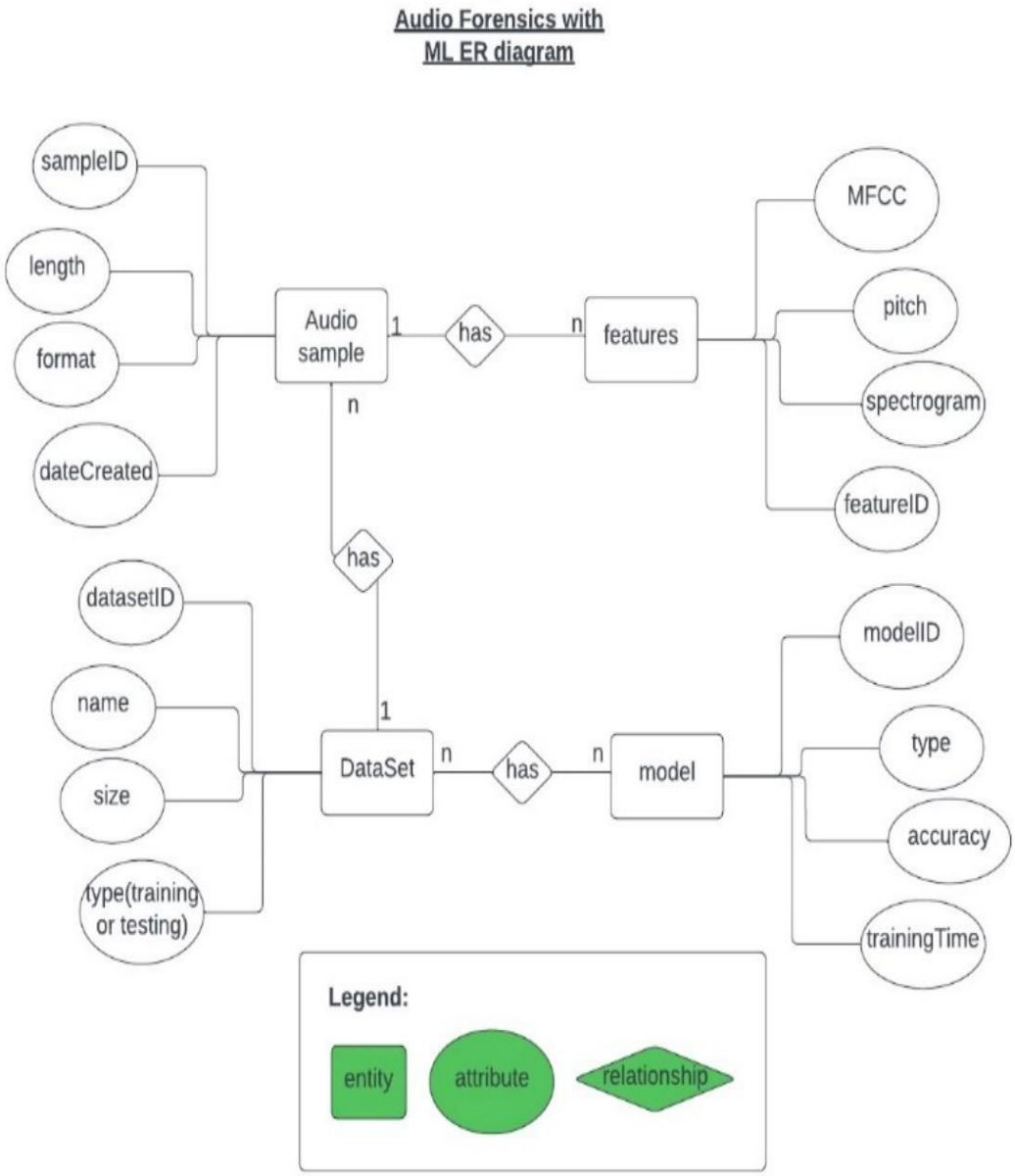

Figure 5 ERD Diagram

2.6 Sequence Diagrams:

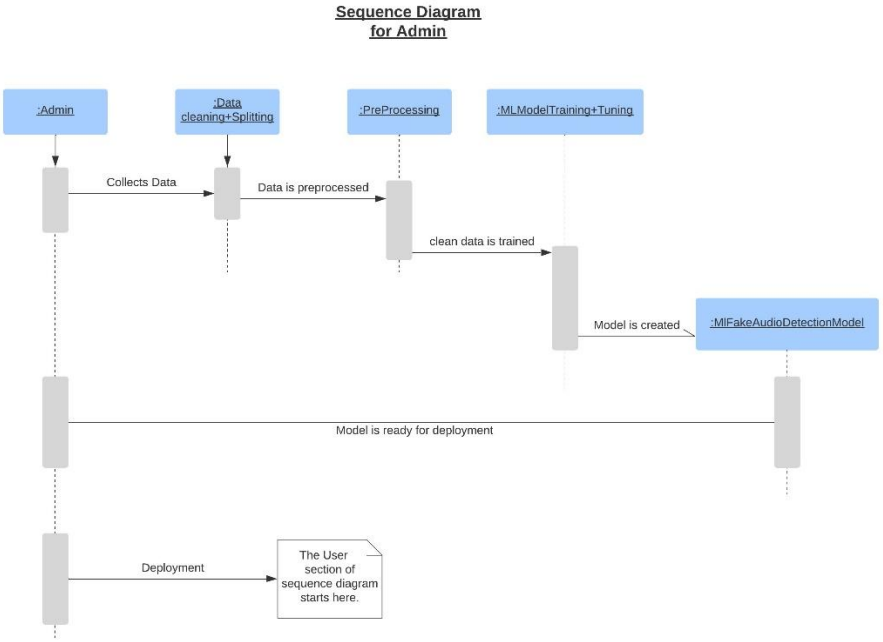

Figure 6 Sequence Diagram 01

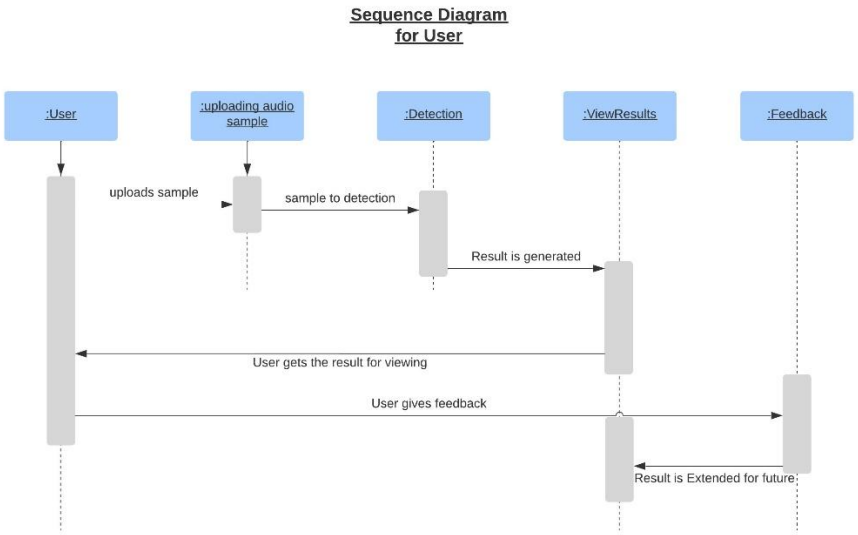

Figure 7 Sequence Diagram 02

2.7 Dataset Used:

Following are the datasets that will be used to train the machine learning for audio forensics. We need real human speech data plus synthetic AI generated data. For this purpose, a mixture of different datasets will be used. These include:

1. The Fake-or-Real (FoR) Dataset (deepfake audio) (Reimao & Tzerpos, n.d.)
2. A private Dataset that is made with the help of ElevenLabs.io

2.8 Interface Design:

As this is a machine learning model that will be tweaked and changed quite a lot during its development and even after a good accuracy rate has been achieved therefore, the main way of interacting with the model will be a Jupyter Notebook. Jupyter Notebooks are easily the best thing that can be used for a machine learning model. Here is a look at such a Notebook:

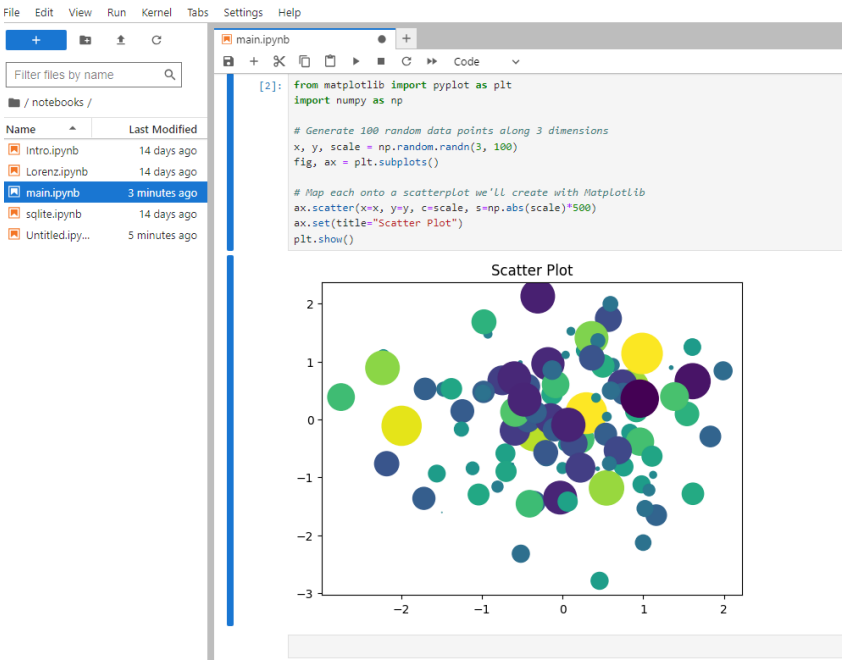

Figure 8 Jupyter Notebook

CHAPTER NO. 3: Development

3.1 Introduction:

The development cycle of the project entailed choosing the neural network model, choosing the best audio features for speech detection, and choosing the best dataset to use for training. The next part is implementing the whole development cycle using the tools and libraries provided in python.

3.2 Tools Used:

There are mainly two choices when it comes to writing neural network architecture in python. Namely, Pytorch and TensorFlow. For this project Pytorch was used for its better hardware compatibility (It can work on AMD GPUs as well). Furthermore the details of the libraries and their usage is as follows:

3.2.1 Torch.nn:

This library is used to make neural network class. The class inherits from this nn Super class. All the functionalities needed for a neural netwok i.e. different type of layers, backpropagation algorithm, various optimizers, and activation functions are pre-implemented in this class, which makes the iteration process very easy.

3.2.2 TorchAudio:

This library is used for audio processing. To convert the given audio file to its waveform, to then converting the waveform into the frequency domain applying Fast Fourier Transform and making spectrogram of the audio file, this library is very handy.

3.2.3 Dataset:

This library comes with Torch as well. This helps in creating a data-pipeline from the dataset to facilitate the training process.

3.3 Main Approach:

There are two main approaches to dealing with audio classification problems such as this one. These include a feature-based approach and an image-based approach. In a feature-based approach different audio features are extracted from the audio data and then a machine learning model is trained to detect those features and classify them as either fake or original. For this purpose, different sequential models are used. The other method is an image-based method whereby the audio data is converted to a Me-lSpectrogram, or MFCC which are basically image representation of the audio data. It has dark and light

spots depending on the different features and frequencies present in an audio file. Then a Convolutional Neural Network (CNN) (Lieto, et al., 2019) is trained on these images to classify them into fake or original. The main image feature used in this project is detailed in the next section. Below is a detailed overview of the neural network:

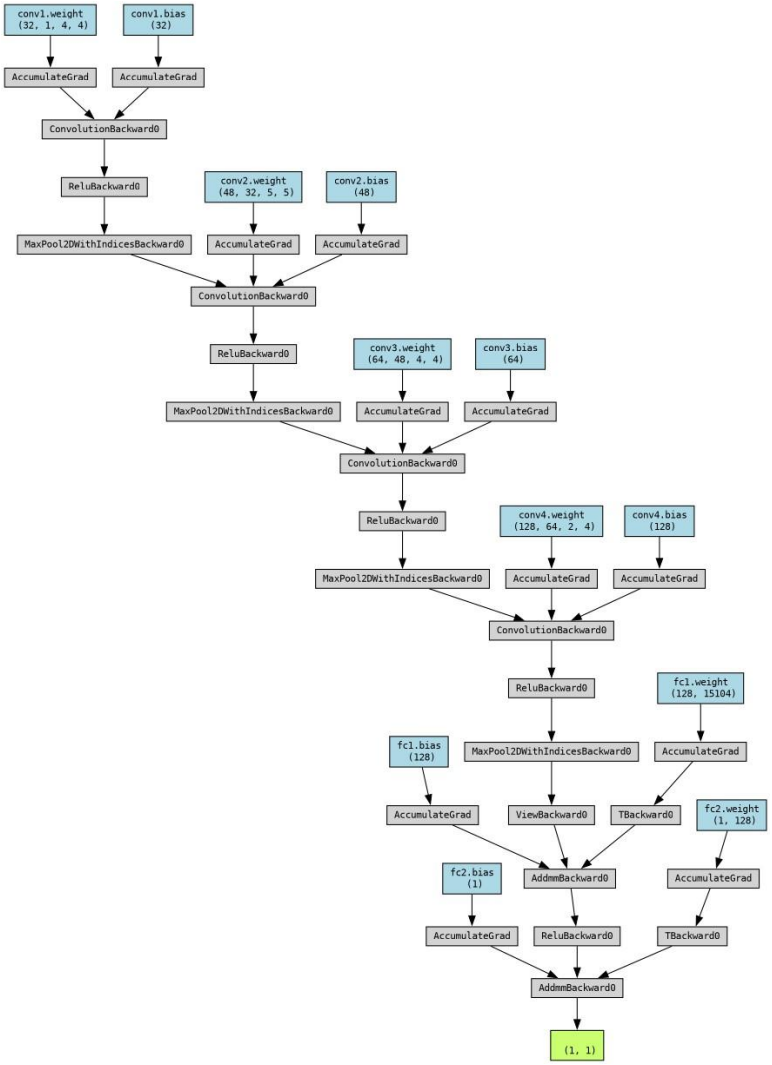

Figure 9 Detailed overview of ShallowCNN

3.4 Audio Features used:

The following main image-based audio features can be extracted from any audio file to train the model on:

3.4.1 Mel-Spectrogram:

A Mel-spectrogram is a representation of an audio signal's frequency content over time, but with frequencies scaled according to the human auditory system's perception. It is a spectrogram with the mel-scale applied to it.

Spectrogram:

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. It is essentially a time-frequency representation of the signal. In a spectrogram, time is represented on the x-axis, frequency on the y-axis, and the intensity of a particular frequency (it's amplitude) at a specific time is represented by the color or brightness of the corresponding point in the plot. Hence, the visual part of such feature comes into play.

Mel Scale:

The mel-scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. It is based on the perception of pitches by the human ear rather than the physical frequencies. The mel-scale is nonlinear and is designed to mimic the nonlinear response of the human ear to different frequencies. It is particularly useful for representing speech. This makes it perfect for use in this project.

Mel-spectrogram:

A Mel-spectrogram is created by applying a mel-filterbank to the frequency spectrum of the audio signal. Here's a Mel-Spectrogram from one of the audio files created during the prototype phase:

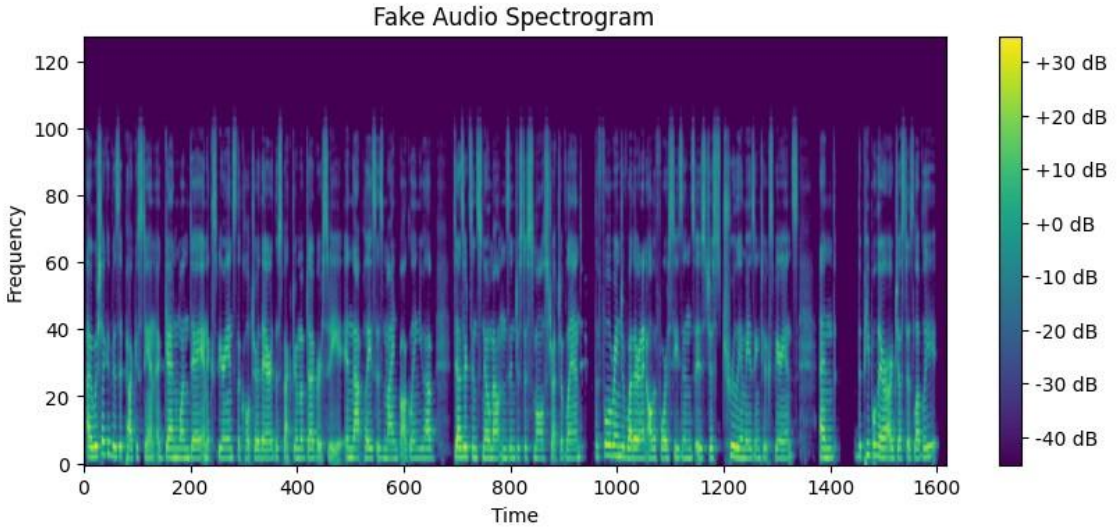

Figure 10 Mel-Spectrogram

3.4.2 Mel-Frequency Cepstral Coefficients (MFCC):

MFCCs are typically represented as a sequence of vectors, with each vector containing a set of coefficients representing the spectral characteristics of the audio signal within a particular frame. The number of MFCCs computed for each frame is a parameter that can be adjusted based on the specific application and requirements. **MFCC is the feature of choice for this project.**

For this project 40 such parameters i.e. MFCC coefficients have been used. Here's an MFCC from one of the audio files created during the prototype phase:

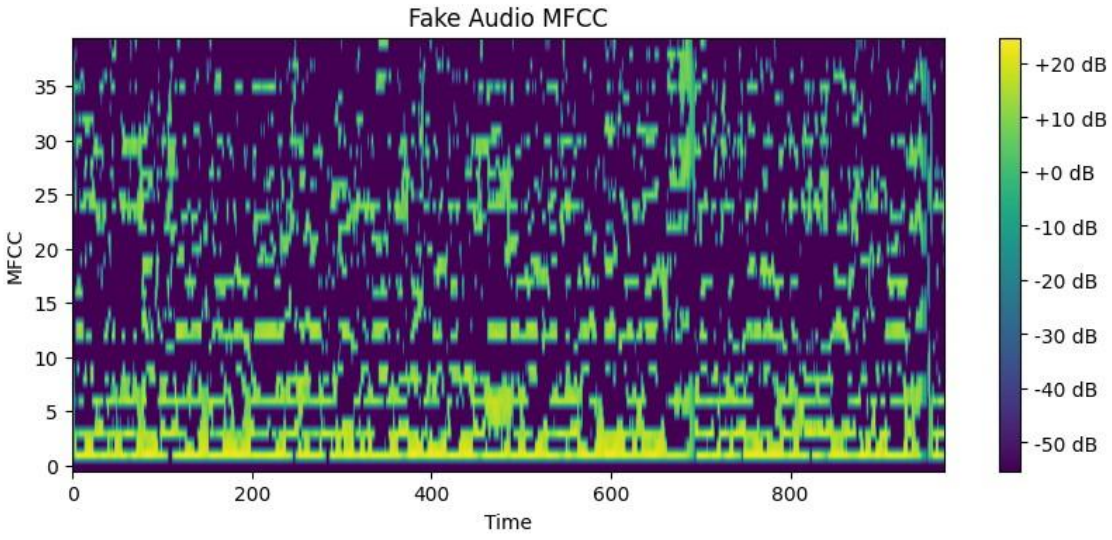

Figure 11 MFCC

3.5 Hyper-parameters used for training:

The training process involves setting up the data pipeline and deciding on the hyper parameters for training. There are two main hyper-parameters of choice here:

Learning Rate:

This is the amount of change that is added to the weights while the gradient descent step of the backpropagation algorithm occurs. For this project the best learning rate was **0.0001.**

Epochs:

This is the number of times the whole dataset is fed to the neural network. For this project the best epoch size was **5.**

Results:

Using the above defined hyper-parameters, the chosen CNN was able to give the **accuracy of 96%** on the test set from the dataset. The test data is the data that the model has not encountered while training.

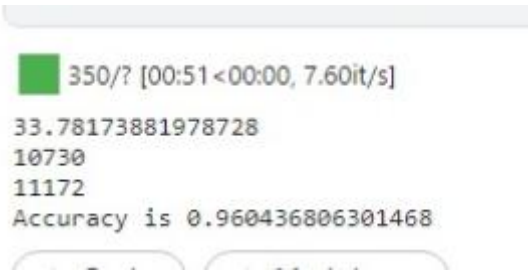

Figure 12 Screen Grab of the accuracy during testing.

Furthermore, the accuracy achieved on the prototype phase data is 90%. Following is a screengrab of that test:

Figure 13 Test accuracy on prototype phase data.

Conclusive Remarks:

While the results obtained from this project are above and around 90 percent accuracy, the model still is far from a generalized model for the task of fake audio detection. The quality of deepfakes has increased far and wide and the datasets used for forensics exercises such as this one have aged badly. The need of the hour is to have better data to train the model on. Furthermore, to really tackle the problem of deepfakes, strong media policies and scrutinizing processes need to be enacted.

References

Huang, M. H., Zhang, P., Tiovalen, J. R., & Balaji, M. (n.d.). *GitHub/Audio Deep Fake Detection*. Retrieved from Github: https://github.com/MarkHershey/AudioDeepFakeDetection/blob/master/models/cnn.py

Kinsley, H., & Kukieła, D. (2020). *Neural Networks from Scratch in Python.*

Lieto, A., Moro, D., Devoti, F., Parera, C., Lipari, V., Bestagini, P., & Tubaro, S. (2019). Hello? Who Am I Talking to?" A Shallow CNN Approach for Human vs. Bot Speech Classification. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/ICASSP.2019.8682743

Reimao, R., & Tzerpos, V. (n.d.). *The Fake-or-Real (FoR) Dataset (deepfake audio)*. Retrieved from Kaggle.com: https://www.kaggle.com/datasets/mohammedabdeldayem/the-fake-or-real-dataset