**A**

**Minor Project Report**

on

**Fake Audio Detection using Machine Learning**

Submitted for partial fulfillment for the degree of

**Bachelor of Technology**

(Information Technology)

in

Department of Information Technology

by

Aditya Jain

189302088

Unnati Jain

189302013

Under the Guidance of

Dr. Gaurav Aggarwal

(July-2021)

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**MANIPAL UNIVERSITY JAIPUR**



**CERTIFICATE**

Date

This is to certify that the project titled **Fake Audio Detection using Machine Learning** record of the bonafide work done by **ADITYA JAIN** (189302088) and **UNNATI JAIN** (189302013) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **(Information Technology)** of Manipal University Jaipur, during the academic year 2020-21.

**Dr. Gaurav Aggarwal**

*Project Guide, Dept of Information Technology*

*Manipal University Jaipur*

**Dr. Pankaj Vyas**

*HOD, Dept of Information Technology*

*Manipal University Jaipur*

**ABSTRACT**

In the modern world, audio morphing is widespread, and detecting such malpractices is foremost important for law enforcement agencies as well as for the protection of any individual's integrity. Advancements in creation of such fake audio clips has overtaken the possible ways of expediting them. We were aiming for a way to detect audio forgery using various machine learning algorithms. [Deepfakes](https://en.wikipedia.org/wiki/Deepfake) are a synthetic means in which a person is replaced with someone else’s affinity. Deepfake audio is created using AI and machine learning algorithms, which is a discrete creation method from the traditional audio manipulation using audio editing software like Lyrebird AI, Overdub etc. Deepfake audio can be used for various purposes, spanning from entertainment to criminal activities. Most deepfake audio apps are meant for entertainment purposes but have the potential to be used for more harmful intents and purposes for wide spreading rumours and for other criminal purposes. This work is an attempt to classify such audio clips as Fake or Authentic. Data set consists of various Narendra Modi speeches from by various rallies across India from different timelines. Data set is pre-processed using a software called Goldwave which helped us reduce the noise in the samples. Data sets will then be used to train a machine learning models using TensorFlow and Keras. We achieved our goal of classifying the audios successfully with an accuracy rate of 97.69%. The accuracy achieved is so high due to the constraints of the dataset.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Title** | **Page No** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Contents** |  |
| Abstract | |  | i |
| List of Figures | |  | ii |
| List Of Tables | |  | ii |
|  | | | |
| **Chapter 1** | | **INTRODUCTION** |  |
|  | 1.1 | Introduction to work done/ Motivation (*Overview, Applications & Advantages*) | 1 |
|  | 1.2 | Project Statement / Objectives of the Project | 1 |
| **Chapter 2** | | **BACKGROUND OVERVIEW** |  |
|  | 2.1 | Conceptual Overview (*Concepts/ Theory used*) | 2 |
|  | 2.2 | Technologies Involved | 3 |
|  | 2.3 | System Architecture | 3 |
| **Chapter 3** | | **METHODOLOGY** |  |
|  | 3.1 | Detailed Methodology Adopted |  |
|  | 3.2 | Architecture of CNN |  |
|  | 3.3 | Properties of Convolution Layer |  |
| **Chapter 4** | | **IMPLEMENTATION AND RESULTS** |  |
|  | 4.1 | Results |  |
| **Chapter 5** | | **FUTURE WORK AND CONCLUSION** |  |
|  | 5.1 | Progress Chart/Timeline Chart |  |
|  | 5.2 | Future Work and Conclusion |  |
|  | | | |
| References | |  |  |

**Chapter 1. Introduction**

Deepfake audio can be used for various purposes, spanning from entertainment to criminal activities. Data set consists of various Narendra Modi speeches from by various rallies all across India from different timelines. We achieved our goal of classifying the audios successfully with an accuracy rate of 97.69%. The accuracy achieved is so high due to the constraints of the dataset. The data set is pre-processed using a software called Goldwave which helped us reduce the noise in the samples. Data sets will then be used to train a machine learning models using TensorFlow and Keras. We hope this will help law enforcement agencies detect such malpractices.

*1.1 Introduction to work done/ Motivation*

Over the past few years, there’s been an expansion in new research using neural networks to simulate a human voice. These models, mainly developed at tech giants like Google, can generate increasingly realistic, human-like speech. While the progress is overwhelming, we are aware of the risks this technology can pose if used with the immoral intent. It may be used to synthesize speech to fool voice authentication systems, or they might be used to create forged audio recordings to defame public figures and spread rumors. Perhaps equally concerning, public awareness of "deep fakes" (audio or video clips generated by deep learning models) can be exploited to manipulate faith in media: as it becomes harder to differentiate real from tampered content. If we perfect the art of fake audio detection, we can help identify fake audios preventing spreading of rumors and hate speech and make better informed decisions. Future application of such a system can make voice recognition and authentication systems more robust and secure.

*1.2 Project Statement / Objectives of the Project*

The inaccuracy with which we can today deal with audio deep fakes and misconception & rumors in the society through social media which could have been prevented, had there been such systems in place. The aim is to devise such a system to avoid the mentioned situations. Our project’s main aim is to predict, with accuracy, the speech features of various public figures like PM Narendra Modi using signal processing algorithms. Then further apply neural network to classify them as a Fake or Real Audio.

**Chapter 2. Background Overview**

*2.1*  *Conceptual Overview (Concepts/ Theory used)*

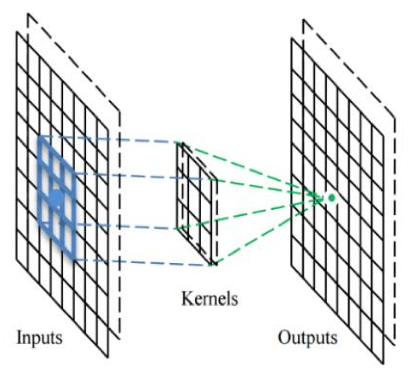
CNN is one of the most common types of neural networks used to recognize and classify pictures. CNNs are commonly utilized in domains such as object detection, face recognition, and so on. CNN image classifications takes an input image, processes it, and categorizes it into several groups. An input image is seen by computers as an array of pixels, with the number of pixels varying depending on the image resolution. The code for the Convolution neural network is written in Python [1] using Keras [2] Package.

Figure 1 Working of Kernel in a Convolution Layer

From the time domain, audio splits are translated to the frequency domain. The amplitude spectrum is the result of this process. The MFCC feature is one of the most essential methods for extracting a feature from an audio signal, and it is frequently utilized when working with audio signals. A signal's Mel frequency cepstral coefficients (MFCCs) are a short group of characteristics (often 10–20) that succinctly define the overall shape of a spectral envelope.

*C(x(t)) = F -1[log(F(x(t))]*

x(t): Time domain Signal

Diagram

Description automatically generatedF -1[log(F(x(t))]: Cepstrum i.e., the result of computing the inverse Fourier transform (IFT) of the logarithm of the estimated signal spectrum.

Figure 2 Working of Kernel in a Convolution Layer

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

This curve plots two parameters:

* True Positive Rate
* False Positive Rate

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

TPR=TP/(TP+FN)

False Positive Rate (FPR) is defined as follows:

FPR=NP/(NP+TN)

*2.2 Technologies Involved*

Google Colab : Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. Google Colab [3] is used as an environment due to the free availability of GPU to accelerate model performance.

Librosa : Librosa is a robust Python package for working with and analyzing audio. It's the first step toward dealing with audio data at scale for a variety of applications, from identifying a person's voice to extracting personal features from audio.[7]

TensorFlow : TensorFlow is an open-source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open-source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API.

*2.3 System Architecture*

Our primary working system(s) are:

Apple MacBookPro13

2.4 GHz Quad-Core Intel Core i5

8 GB LPDDR3

Intel Iris Plus Graphics 655

512GB SSD

HP Pavilion x360 Convertible

Intel Core i3 -8130U

4 GB DDR3

NVIDIA GeForce MX130 Graphics

1TB HDD

**Chapter 3. Methodology**

*3.1 Detailed Methodology Adopted*

The main ambition of the project is to segregate and identify various sound modules using techniques of machine learning in python. First, We proceeded with collection and preprocessing of a dataset that has some data that can be used for both visual and theoretical presentation.

Since the samples in consideration are from live events and speeches there was a lot of noise and other issues with samples. Thus, data set is preprocessed using a software called Goldwave which helped us reduce the noise in the samples, treating mic bleeding and various other issues.

The data set consists of various public figures speeches from different timelines along with mimicries done by voice artists that were posted on various social media platforms.

The dataset is then divided into categories mainly training (70-80%) and testing Dataset(20-30%).As the name suggests, the former dataset is used to train the model while the latter is used for model testing. The audio is cut into frames of fixed size. The next frame usually overlaps the previous frame to capture maximum information. The process is repeated until the end of the file and in case of presence of a shorter frame, zero paddings are added to it. These audio splits are converted to frequency domain from the time domain. The spectrum obtained is called the amplitude spectrum.

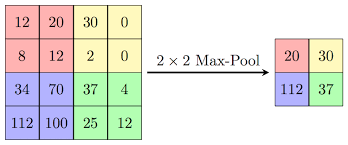
Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map. This helps in down sampling of images and retaining the most important information.[5][6]

Figure 3 Working of Max Pooling

The MFCC feature is one of the most essential methods for extracting a feature from an audio signal, and it is frequently utilized when working with audio signals. A signal's Mel frequency cepstral coefficients (MFCCs) are a short group of characteristics (often 10–20) that succinctly define the overall shape of a spectral envelope.

*C(x(t)) = F -1[log(F(x(t))]*

MFCC defines the power spectral of one frame only but over time these spectral coefficient change and hence it was observed that appending these trajectories of features to the power spectral significantly improved the performance of ASR. These are known as delta coefficients.

𝑑t=𝛴Nn=1( c t+n - c t-n) / 2𝛴Nn=1 n2

Here 𝑑𝑡is a delta coefficient, from frame t computed in terms of the static coefficients 𝑐𝑡+𝑛 to 𝑐𝑡−𝑛. A typical value for N is 2 (derivative order).

CNN is one of the most common types of neural networks used to recognize and classify pictures. CNNs are commonly utilized in domains such as object detection, face recognition, and so on. CNN image classifications takes an input image, processes it, and categorizes it into several groups. An input image is seen by computers as an array of pixels, with the number of pixels varying depending on the image resolution. The code for the Convolution neural network is written in Python [1] using Keras [2] Package. As we add more convolutional layers, the accuracy tends to saturate and then degrades quickly. Features learned from one of the previous layers are stacked with a new layer. So as the layers increase, almost certainly new features can be learned due to residual features which are extending from layers before.

*3.2 Architecture of CNN*

Diagram

Description automatically generated

Figure 4 Architecture of CNN

*3.3 Properties of Convolution Layer*

As observed after every few stacked layers input of the primary layer gets added to the last layer. This enables features to be learned efficiently by deeper layers. A variety of operations can be performed on the two layers like addition, average, concatenation etc. Taking inspiration from this architecture, I used a block structure as shown in Figure 4. Additional details of the convolution layers are given in Table 1.

**Table 1.** Properties of each convolution layer in the proposed architecture

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer Number** | **Properties Kernel Size** | **Filter** | **Activation** |
| **Conv\_2D\_1** | (3,5) | 16 | Tanh |
| **Conv\_2D\_2** | (3,5) | 32 | Tanh |
| **Max Pool** | Pool size = (2, 5) |  |  |
| **Conv\_2D\_3** | (5,5) | 64 | Tanh |
| **Conv\_2D\_4** | (5,5) | 126 | Tanh |
| **Max Pool** | Pool size = (2, 5) |  |  |

To standardize our results, we used the following architecture (Table 2) to use each of the models.

**Table 2.** Complete Architecture of Network Used

|  |  |  |
| --- | --- | --- |
| **Layer Number** | **Name** | **Specifications** |
| 1 | Input Layer | Resize according to requirements of architecture |
| 2 | Existing Convolutional Layers (from Table 1) | - |
| 3 | Flatten | - |
| 4 | Dense | Activation = Tanh Shape = 128 |
| 5 | Dense | Activation = Tanh Shape = 32 |
| 6 | Dense | Activation = Sigmoid Shape = 1 |

**Chapter 4. Implementation and Result**

*4.1 Result*

Using the proposed complete Architecture as given in Table 1 and Table 2, the Accuracy achieved was 97.69%. The dataset in consideration is an amalgam of various PM Narendra Modi speeches and that of various people []. Precision i.e. Accuracy of positive predictions was observed to be 0.97 while the percent of positive predictions were correct i.e. F1- Score was observed to be 0.98.

Shape

Description automatically generatedFigure 5 shows the performance of a classification model at all classification thresholds using ROC Curve.

Figure 5 ROC Curve

**Chapter 5. Future Work and Conclusion**

*5.1 Progress Chart/Timeline Chart*

GANTT CHART:

1. Research and analysis
2. Synopsis abstract and literature review
3. Study and collection of data
4. Designing blueprint
5. Training, testing, and preparation of the dataset
6. Model implementation
7. Documentation
8. **A screenshot of a computer

   Description automatically generated with low confidence**Final evaluation and completion of a project

*5.2 Future Work and Conclusion*

The proposed architecture works efficiently on the applied dataset. The audios are converted to MFCCs along with its derivative to attain a two-dimensional input for the model. Using this architecture, the model learns quickly, achieving comparable accuracy on the datasets. The model achieves 97.69 % accuracy which shows the potential of the architecture to perform better with further works. However, testing with more data is required. A diversified testing data shall be created in future work to check it with real case scenarios and more noise. Furthermore more datasets of other Public Figures shall be tested to generalise the architecture. This also open the future capability of the network to be used for tasks like speaker identification and verification, and study of other similar wave forms.

**REFERENCES**

[1] Chollet, F., &others.(2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[2] "G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrumvoor Wiskundeen Informatica (CWI), Amsterdam, May 1995."

[3] Bisong E.(2019) Google Colaboratory .In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. A press, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8\_7

[4] Christopher T. (2019), “An introduction to Convolutional Neural Networks”,

https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7 (Mar. 1 2021).

[5] ComputerScienceWiki,(2018),“Max-pooling/Pooling”, https://computersciencewiki.org/index.php/Max-pooling\_/\_Pooling, (Mar. 1 2021).

[6] MahajanP.(2020),“MaxPooling”,https://poojamahajan5131.medium.com/max-pooling-210fc94c4f11, (Mar. 1 2021).

[7] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg,

and Oriol Nieto. “librosa: Audio and music signal analysis in python.” In Proceedings of the

14th python in science conference, pp. 18-25. 2015