Deep Fake Audio Detection

Abdullahil Kafi
DKE, Otto von Guericke University
Magdeburg, Germany
abdullahil.kafi@st.ovgu.de

Israt Nowshin
DKE, Otto von Guericke University
Magdeburg, Germany
israt.nowshin@st.ovgu.de

Manali Thakur
DE, Otto von Guericke University
Magdeburg, Germany
manali.thakur@st.ovgu.de

Satish Khadka
DKE, Otto von Guericke University
Magdeburg, Germany
satish.khadka@st.ovgu.de

Abstract

This paper mainly focuses on Deep Fake Audio: it's creation and detection. Recently there have been much research and advancement in this field but work related to Deep Fake Audio is less compared to Deep Fake Video. Thus, in this paper, we will be explaining how we generated deep fake audios and detected Deep Fake Audios..

Different generation methods were used to generate Fake Audios from some samples of audios and Convolutional Neural Network (CNN) has been used for the detection of the Fake Audios. It is evaluated in two ways: using Fake Audios that has been generated and adding some noise and corrupting the dataset. The proposed model has around 85 percent accuracy in detecting fake audios while it gives 83.3 percent accuracy in detecting fake audios while the data is corrupted.

Keywords: Deep Fake Audio, CNN, Auto-encoder, Variational Auto-encoder

1 Introduction

Advancement in technology is playing a major role in making our lives easier by enabling us to complete many tasks with ease nowadays. One of them is developing systems that can clone a person's voice. Deep Fake Audio means cloning a person's voice and being able to create a similar voice using Deep Learning. These advances have some advantages like designing assistive technologies, educational technologies, and games [1]. However, it has many disadvantages as this advanced technology can be used to harm people in many ways, like breaching someone's privacy and also spreading misinformation. They can even defeat the Speaker Verification system (ASV) and voice biometric system. Nowadays, improvement in Deep Fake Generation technology can make the detector unable to differentiate a fake voice and a real voice, which is a matter of concern.

The advancement in multimedia focuses on making authentication easier and more secure. Among the different authentication methods, the most common ones are fingerprint, face recognition, voice recognition, etc. These authentication methods are important to ensure more security of a user's profile and device. However, technological advancement did not only make security measures easier. There are many challenges to these multimedia authentication systems, for example, voice spoofing can be a threat to voice recognition authentication systems. Deep Fake Audios can generate almost a similar voice as a real person that becomes difficult for both human and machine to identify. Thus, being able to differentiate between real and fake audio has become a crucial part of ensuring the security of an individual.

2 Related Works

The approach used by authors in [2], for classification has been divided into four components: Speech Denoising using DNN (MLPs and CNNs) with adaptive filters, used NLP for speaker diarization, RNN for speaker labeling and classifying real and fake audios using CNNs; all achieving training accuracy over 90% and test accuracy to be 89%. The authors were not satisfied with their results and so performed ‘Transfer learning’, that is, added a new last layer to a pre-trained model, which increased the accuracy.

Similarly in [3], authors have used a real human speech database, generated using Google WaveNet and proposed a classification algorithm based on Convolutional Neural Networks (CNNs). The audio signals have been normalized and pre-processed into classic spectrograms and mel-spectorgrams. The dataset has been produced from four different cloud services for automated TTS conversions without sub-sampling, or channel mixing in order to avoid introducing any further trace that could bias the achieved results which have given greater than 90% accuracy and when used multiple bots for training, accuracy stayed over 70% for the test sets.

A new method for enhancing the generalization of detection method by a capsule network has been introduced in [4]. Also, the proposed method has shown capability to detect replay attacks. STFT and Linear frequency cepstral coefficients (LFCC) are extracted from the audio to form a feature map and the proposed capsule network consists of convolutional groups with funnel activation and then, a convolutional layer with leaky-ReLU is used. In this report, a similar approach has been considered, baselined with CNN approach.

Recent advances in technology led to the innovation of indistinguishable synthesized audio speeches. Run Wang et al. (2005), proposed a novel approach named DeepSonar in their paper which monitors the neuron behavior of a Deep neural network-based SR system with a simple binary classifier to crack down the AI synthesized voices. The authors focused on the layer-wise neuron activations for the deep network which can distinguish the different inputs in layers to detect the real and synthesized audios. Moreover, they researched three datasets and two languages for better detection rates and low false alarms. Furthermore, the authors worked on two types of fake audios generated by TTS and VC as they are more realistic and indistinguishable to human ears. Experimental results showed effectiveness in distinguishing fake voices and robustness against two manipulation attacks, voice conversions, and additive real-world noise. However, this paper also talks about the reason to use CNN which complies with the layer-wise neuron activation, and the evaluation methodology which talks about additive real-world noise [5].

Fake speech is created mostly using the artificial intelligence which is none other than deep learning techniques and signal processing techniques. Moreover, imitation can be a real deal in terms of creating fake audios as it is really natural and it is hard to distinguish. Dora M. Ballesteros et al. (2021) proposed a method of detecting fake speech using a Convolutional neural network that uses image augmentation and dropout. The authors trained an architecture with 2092 histograms of both original and fake voice recordings and cross-validated with 864 histograms. However, the authors focused on calculating precision and recall for the model which can be used to create a confusion matrix. The global accuracy was around 98% for the proposed model. The authors used both hand-crafted and automatic feature extraction methods for the model training. Additionally, the authors proved some hypotheses that are very useful for future researchers for their work and prove some vital information about fake and real audios [6]. This paper also talks about CNN which can be a good classifier for images.

However, on the other hand, Clara Borrelli et al. (2021) proposed short-term and long-term prediction traces for fake audio detection. The authors focused not only on the detection of the fake audios but also on the algorithms that produced the fake audio. Additionally, they considered two scenarios for the experiment, close-set and open set environment. Moreover, in a close-set scenario the detection of an algorithm is a must but in the open set environment, the architecture can detect and specify if the fake audio generation algorithm is something novel. Furthermore, the authors used multiple auto-regressive orders at once to create the feature set which is something novel. In the validation process, they used the ASVspoof 2019 data set which is publicly available [7]. In this research, ASVspoof [8] has been used for being the benchmark dataset in this field.

Ricardo Reimao (2019) in his thesis paper proposed a synthetic speech detection methodology using deep neural networks which achieved around 99% accuracy and better than human performance which is a very interesting fact [9]. The author also used the dataset of ASVSpoof, as it gives a wide variety of bonafide and spoof recordings. Furthermore, the author focused on using Neural networks, CNN, LSTM, GANs, and different classification algorithms. Moreover, the authors evaluated it in different scenarios with different experimental setups to see the drop in accuracy for each of the models which seems to be very useful in real world scenario.

In the survey of Deep Fakes Thanh Thi Nguyen et al. (2019) find out many algorithms which are used for the generation and detection of fake audios [10]. The authors presented an extensive discussion on the algorithms and methodologies as well as the challenges to using these algorithms. Furthermore, they discussed the challenges of feature selection and the proper feature selection for deep fake detection. Deep fake has become popular due to the quality of tampered videos and also the easy-to-use ability of their application. Furthermore, they have compared the CNN, GANs, LSTM, SVM methodology and the challenges these algorithms have. The survey gives a decisive idea that CNN, ResNet, and GANs can be good options for image classification which gave the confidence to this research to choose CNN as a model.

Aarti Karandikar et al. (2020) proposed a fake video detection method using convolutional neural networks [11]. Although this research's concern is with audio data however, the work seems interesting as they are proposing a method in which they have extracted the video features as still images. Automatic feature extraction method was used which worked fine for video data. Moreover, the authors used autoencoders to generate the fake videos which were later improved by the inclusion of GANs. However, they got around 70% accuracy for the CNN model which they applied on still images. On the other hand, Zhuxin Chen et al (2017) used ResNet and Model Fusion for Automatic Spoofing detection. The authors mentioned that according to ASVspoof 2017 it is clear that ResNet performs better than any other algorithm in spoof detection [12]. Different features and models for spoof detection has been used to gained more efficiency than any single-model system. Furthermore, the authors got good results while using ResNet with different filter sizes varying from 23 to 60.

Digital audio quality depends on the number of channels used to create the audio. Stereo audio consisting of two channels outperforms the mono audios. Tianyun Liu et al. (2021) proposed an SVM and CNN method to detect fake stereo audios. For the SVM architecture, the authors used Mel-frequency cepstral coefficient features while they used a five-layer CNN for the second methodology [13]. Furthermore, the research states that the waveform of deep fake audios and real audios can hardly be distinguished with human eyes while it is actually possible to find out the different trends with feature extraction approaches. Meanwhile, in this research, CNN has been used for the image input, not as an audio input signal cause the model hardly performs well with audio input.

According to Chen et al.(2020), Deep Fake which is also known as logical access voice spoofing is increasingly becoming a threat because of advancements in voice synthesis and voice conversion technologies. Thus to overcome this issue of generalization of audio Deep Fakes, the authors used large margin cosine loss function (LMCL) and online frequency masking augmentation to force the neural network to learn more robust feature embeddings [14]. In order to protect voice-based authentication systems from malicious attacks, the authors created the system to access logical access attacks. LMCL increases the variance between genuine and spoofed classes, and at the same time decreases the intraclass variance.

There has been a remarkable improvement in the Fake audio generation due to the improvement in the deep neural network models. Hira Dhamyal et al. (2021) states in the paper that the it is possible to detect fake audio in resources-constrained setting such as on edge devices and embedded controllers as well as with low-resource languages. In the paper, the authors analyzed two micro features: Voice Onset Timing (VOT) and coarticulation. VOT is defined as the length of time that passes between the release of a stop consonant and the onset of voicing, the vibration of the vocal folds. Coarticulation refers to the influences of phonetic segments on adjacent or near-adjacent segments that are observable in the acoustic or articulatory patterns of speech. The ideology here is to capture the difference of the voice producing mechanism in humans which includes the air pressure system, vibratory system, and resonating system which exist in the real audio but are not copied or replicated by the synthesized audio. Furthermore, the authors used the AutoVOT library which requires word aligned transcript and the audio signal as input. Audio is passed through the AutoVOT model which results in boundaries of predicted VOT. Finally, a conclusion was drawn that spoof speech can exceed the normal range of VOT and on average is higher than the VOT in bonafide speech [1]. This methodology inspired this research as it porved that with low resources fake audios can be detected.

Multiple datasets have been designed to hold back the development of fake audio detection. However, they exclude the situation when an attacker might hide some small fake clips in between the real audios. Jiangyan Yi et al. (2021) discusses the development of a dataset to tackle such scenarios.The dataset involves changing only a few words in an utterance using a synthesis technology. The HAD dataset is based on the AISHELL-3 corpus which is a multi-speaker Mandarin speech corpus for training text-to-speech (TTS) models. The dataset consists of two subsets: A partially fake audio set and a Fully real and fake audio set and the evaluation is carried out in terms of Equal Error Rate of the utterance level and Precision (P), recall (R), and F1-score(F1) for the overall segment-level fake audio detection performance. The authors finally concluded that it is more difficult to detect the partially fake audio than the fully fake audio [15]. Similarly, this research tried to follow the related methodology by tweaking the real audios by introducing some fake audios in between and seeing how well the model is able to distinguish the real ones from the modified ones.

There are a lot of detection approaches that perform really well in differentiating fake audios from real ones. Nonetheless, the problem persists when the system has to deal with unseen spoofing data. Fine-tuning and retraining might help mitigate the problem but this procedure, however, requires a lot of time and computational resources. Haoxin Ma et al. (2021) has defined a way to detect fake audio without forgetting. The authors proposed a continual learning method named detecting fake audio without forgetting (DFWF) to learn new spoofing attacks as they are introduced. The aim is that the system never forgets past knowledge and focuses more on the invariance of genuine speech. For this, the authors proposed three methodologies: (1) Regularization approaches, where hand-craft regularization term in loss function is added to constrain the learning process. (2) Replay approaches to record some examples in the buffer and replay experience when training new tasks. (3) Dynamic architecture approaches to dynamically change the structure of neural networks according to the new task [16]. Using this method not only preserves the model's detection capabilities on previous data but also saves time and computing resources and also mitigates the catastrophic forgetting problem.

Recent advances in the inverse mapping of image generators have led to the question of whether it is possible to map audio to a latent vector that can regenerate it. A lot of methods like Optimization-based methods and Auto-encoders based methods have been defined to address the problems. Nicky Bayat et al. (2019) discusses an auto-encoder-based method that trains a neural network encoder model to predict latent vectors of audio samples. The idea here was to show that the approaches used for projecting images to latent space can be used on audio as well. The authors were the first to employ a multi-level feature loss besides the MSE loss in training an audio inverse mapping model. The goal for the inverse mapping model was to take as input the spectrogram of generated and real audio and output the predicted latent vector for a pre-trained WaveGAN. To train the inverse mapping model two types of objective functions were used : (1) MSE between latent vectors and (2)

Multi-level feature loss[17]. The deep network encoders were found to be faster than optimization-based models and are more accurate in terms of both auditory quality and classification accuracy.

3 Methodology

In this section the authors discuss about their methodology and the workflow which is followed. This section is divided into two subsection. Firstly, generation of the fake audios and secondly, fake audio detection. Each subsection contains a detail discussion on the methods followed by the authors. Figure 1 demonstrates the workflow of the proposed methodology.

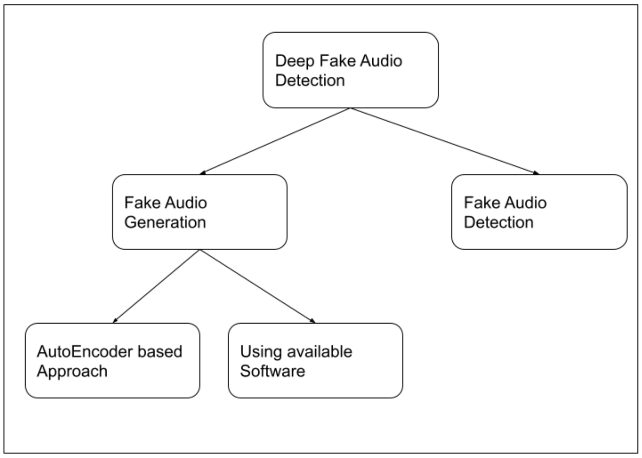

Figure 1. Methodology and Workflow

3.1 Generation

This section provides an in depth discussion about the fake audio generation using different deep learning techniques. A brief discussion is given for the methodologies that has been explored to generate fake audios for the generation mechanism. This section is divided into two subparts, Deep learning based audio generation and existing software based audio generation.

3.1.1 Deep learning Based Audio Generation.

In this subsection, the authors discussed mostly about the Auto-encoders which was used in both systems. However, Auto-encoders is also a part of the unsupervised machine learning which tends to work on unlabeled data mostly. As discussed in the previous section most of the fake audios are generated using different kind of auto-encoders.

Audio Generation Using Variational Autoencoders and CNN

For this methodology, the authors used Variational autoencoders with a Convolutional neural network (CNN). There are different methodologies to generate audios and the most common is to use the text-to-speech methods. However, there are also different audio methods that can be applied to generate audio. Any user can give audio as an input and the algorithm will imitate the voice to create new audios which will be generated randomly in the designated language with given text. For this experiment, English language has been used.

This architecture is based on three parts. First part is about audio pre-processing, second part is to create variational auto-encoders which analyze the existing audio and the third part is to generate audios using Variational Autoencoders.

At first, the audio is taken as an input and then it is pre-processed to remove noise. For this experiment, WAV format audio files were used as an input and first converted into waveform array then to a spectrogram. Before converting the audio into a spectrogram it was denoised using different methodologies. Then, for the same audio file, multiple spectrograms were created using different lengths, and later on, multiple spectrogram chunks consisting of a shorter time period were created but it was assured that these were of equal size. After this, for each audio, a folder was created which will consist all the equal-sized spectrograms. However, in this way the mostly used feature is extracted from an audio sample which is the voice.

In the next phase, the model was created for analyzing the input data. This consists of Variational Autoencoder. However, an Autoencoder consists of an Encoder and Decoder model. In this experimental setup, the authors created a deep convolutional autoencoder model consisting of a mirror encoder and decoder architecture. Multiple layers were chosen for the model, starting from the input which takes a grayscale image which was the pre-built spectrogram with the shape of 28 * 28 *1. It is followed by five convolution layers with dimensions 512, 256, 128, 64, and 32. After every convolution layer, the activation function was set as ‘relu and batch-normalization was applied. For the encoder part, the data has been downsampled from a large input size as the convolution size also decreases. After the five convolutional layers, we used to flatten and log variance to get the final output of the encoder model. Next, this output of the encoder model is fed to the decoder which has the decoder input followed by a dense layer and reshapes layer. After that five convolutional transpose layers were built and with the same relu activation and batch normalization to decode the encoded layers apart from the last layer where the sigmoid activation function was used. After running the model we got the following results, which are shown in Table 1,

Table 1. Parameters

Parameters	Number
Total params	2,372,673
Trainable params	2,369,729
Non-trainable params	2,944

However, next, the model was trained in 100 epochs using the pre-processed audio data to get an average loss of 0.08 and Kl loss around 120.344. Furthermore, doing the training with 200 epochs led to an average loss of around 0.076 which seems to be a little less than the 100 epochs loss. While trying to generate after training the model, it fails to generate any actual audio. With different sizes of text the model generated a fixed length audio with noise. It could not generate an actual audio signal imitating the voice that was used during training. For this model, the authors had to record 50 sample voice for a single person and generate spectrograms. However, it is understood that the pre-processing part was not correctly done, which resulted the failure.

Figure 2 shows the generated and input spectrogram after generating five audio samples. The input spectrogram is the audio of the provided text to the model in real voice and the generated spectrogram shows the genereted audio of that text by the model.

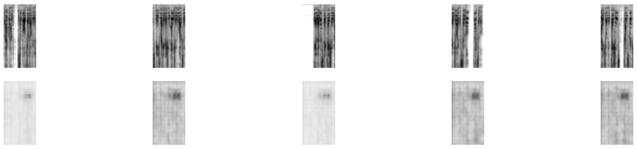

Figure 2. Real and Generated Audio Mel Spectrograms

Figure 3 and 4 shows the individual audio input and the generated audio spectrogram.

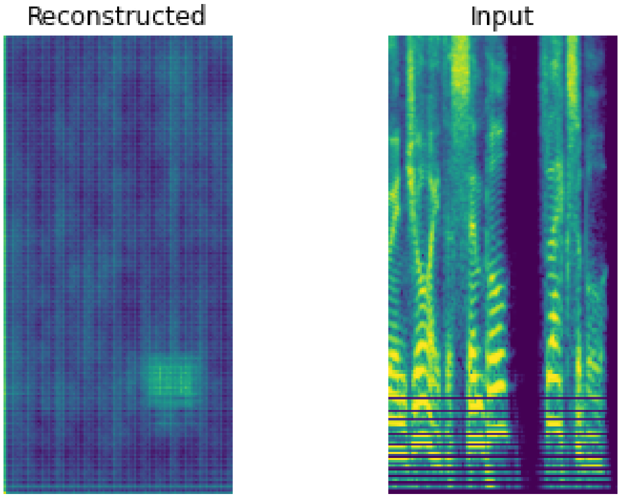

Figure 3. Audio Input and Regenerated Audio Mel-Spectrogram

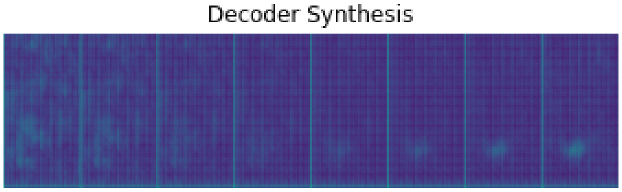

Figure 4. Generated Mel-spectrogram

Generation using variational autoencoders. In this subsection the authors discuss about the usage of variational autoencoders to generate fake audio and the challenges faced during the process. This idea was greatly inspired by Valerio Velardo, a music and AI researcher[1]. A small disussion about the terms is being followed by the actual method.

DimensionalityReduction : Dimensionality Reduction refers to the process of reducing the number of features that describe the data by transforming the data from High-dimensional space to Low-dimensional space [18].

Encoders : Encoders are networks (CNN in this model) that takes inputs and produces a low dimension latent vector which holds meaningful information or represents the input [19].

Decoders : Decoders work in the opposite manner to that of encoders. They are built using a similar network as that of encoders.Decoders takes in the feature vectors produced by the encoders and produces the output that is similar to the actual input [19].

VariationalAutoencoders : A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder which outputs a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute [19].

Kullback − Leibler − Divergence : The KL divergence between two probability distributions simply measures how much they diverge from each other. Minimizing the KL divergence means optimizing the probability distribution parameters (μ and σ) to closely resemble that of the target distribution [18].

The initial steps involved creation of an encoder and decoder model. For building the model we used Convolutional Neural Network(CNN). 5 layers were built with the filter size of 512, 256, 128, 64, 32 respectively and kernel size of 3*3. Adam was used as the optimizer for the CNN model and two loss functions were used, reconstruction loss(Root Mean Squared Error) And KL Divergence. The inputs were then compressed into a dimension of 128 which is also known as the bottleneck. Similarly, the decoder model was generated exactly the same way as the encoder model by reversing the procedure used for building encoder and feeding in the latent dimension as the input to the decoder model.

The next procedure involved pre-processing the audio files where the authors decided on the sample rate or the frame size and the duration in seconds for how the audios were to be extracted. For any audio not meeting the criteria, padding was used to make up for the missing values. Finally the audios were converted into a log spectrogram using the built in python library. Figure 5 illustrates the idea behind the model.

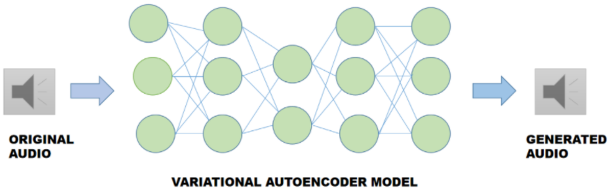

Figure 5. VAE Model for Generation[2]

The Final step involved generating the audios from the previously generated log spectrograms. For this method, The model was trained using various audio files using different learning rates and epoch size. The steps were repeated for a multiple number of times using different learning rates and epoch sizes and also tweaking the sample sizes in order to increase the accuracy and improve the generated results.

Furthermore, the major challenges faced using the above mentioned steps were that the model did really well for the generation of the spectrograms and also trained rigorously with accuracy around 80-90%. However, the audio that was generated sounded very distorted and was nowhere close to the original audio. We further worked on improving the model by modifying the parameters and also pre-processing the original audio even further but the attempt was a failure and the authors had to adopt a different procedure for generation of the audio.

[1] https://github.com/musikalkemist

[2] Idea adapted from https://github.com/musikalkemist/generating-sound-with-neural-networks

3.2 Audio Generation using Software

An open-source software named ‘Real Time Voice Cloning’ is used for generating audios using the voices we preferred cloning. It is a neural network based system implementation of “Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis (SV2TTS)"[3]. It is a three-stage deep learning framework that allows to create a numerical representation of a voice from a few seconds of audio, and to use it to condition a text-to-speech model trained to generalize to new voices [20].

The voice-cloning system is based on majorly three components as described by the author: 1) a speaker encoder network, trained using an independent dataset (noisy speech of thousand speakers without transcripts), thereby, generating a fixed-dimensional embedding vector from seconds of reference speech from a target speaker; 2) a sequence-to-sequence synthesis network, generating a mel spectrogram from text, conditioned on the speaker embedding; 3) an auto-regressive WaveNet-based vocoder that converts the mel spectrogram into a sequence of time domain waveform samples [20]. The software is able to synthesize both, seen as well as unseen data.

Installation: In order to work with this system, it is required to install Python (version 3.6 or 3.7) and ffmpeg. Ffmpeg [3] is another open-source and free software that deals with multimedia files, i.e., audio and video.

The default dataset that the software uses is the LibriSpeech datatset. The dataset doesnot contain punctuations and thus, sentence is to be written on a new line to indicate line-breaks. As it is a cross-platform software, any dataset can be added. Also, new audios recorded by the user can be used for the generation of synthetic audios. From the dataset, the speaker and the utterance is randomly selected after loading the speech in the software. The software shows the mel-spectogram and the embedding of the loaded utterance. Embedding is the numerical representation of the voice, generated by the software. The software also has a functionality to show the visualization of the embeddings if more than 3 utterances are loaded. A cluster can be seen in this visualization, thus embeddings from different speakers create distinct clusters. A new synthesized audio is generated from the embeddings and the text provided to the software.

As the synthesizer has been trained on the audio books, the tone of the synthesized audios is seen to be artificial. The synthesizer turns the mel-spectogram into speech and the vocoder converts the speech into cloned voice. The encoder, synthesizer and the vocoder can be selected from the options provided. In this report, the gensmelraw and the Griffin vocoder has been used. An audio of approximately 5 seconds needs to be provided to the software. The audio processing was seen to be faster for shorter inputs by the Griffin vocoder, however, the audio quality was distorted.

The dataset creation for the fake audio detection involves the synthetic audio from different speakers of differnt gender, voice tones as well as audios generated from user’s real audios. It also includes shorter sentences (6-10 words), longer sentences (15-30 words) as well as seen and unseen text, that is, the text that is not used for training the synthesizer.

3.3 Deep Fake Detection

Audio deep fakes are the curse of the modern technological era which leads to misinformation. The way to prevent audio deep fakes is ongoing research which is inspired by the video deep fakes. For video deep fakes, the video stream is manipulated and it could be detected using different classification algorithms. However, for audio detection, a simple classification algorithm can not be trusted, which means that the use of deep neural networks is a must. Furthermore, most of the research suggests the usage of CNN, GANs, and Autoencoders despite autoencoders being really naive in this situation.

In this subsection, the proposed method for audio deep fake detection will be discussed. Besides that, the dataset preparation and the workflow of the methodology will also be discussed. The dataset was intended to be built by using deep

[3] https://github.com/CorentinJ/Real-Time-Voice-Cloning

generation methodologies in the beginning. However, as it was not working as it was supposed to be, the plan was changed and it was shifted to the software-based generation process to build the dataset. The experiment started with 150 fake audios and it continued till 500 datasets. For generation of audios, short audio clips are used which are recorded randomly using different sound clips and they are later used to generate the fake audios. Apart from these generated fake audios, the dataset of ASVspoof [8] 2019 database is also used for evaluation which consists of 107 speakers (46 males and 61 female) voices. The best part of the dataset is that the data is already partitioned into three parts, namely training, development, and evaluation. However, for the evaluation part, only the evaluation data which consists of 48 voices (21 male and 27 female voices) are used. Figure 6 shows the diagram for the detection methodology.

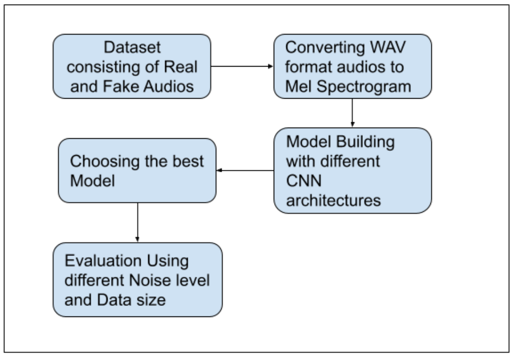
Dataset consisting of Real and Fake Audios
Converting WAV format audios to Mel Spectrogram
Model Building with different CNN architectures
Choosing the best Model
Evaluation Using different Noise level and Data size

Figure 6. Detection Framework

In the second step, the '.wav' format audios are converted into Mel-spectrograms using wav2mel[*] which creates a Mel spectrogram. Different frequency levels are set to see the Mel spectrogram. Mel spectrogram is mostly used in audio classification as it is not the standard spectrogram; rather it uses Mel scale instead of normal frequency in the y-axis. Moreover, it uses Decibel Scale instead of Amplitude to indicate colors. However, for this method, the libriosa library is used which has the visualization property for a Mel spectrogram from audio or NumPy array using different sample rates. Besides that, the whole dataset consisting of audios is converted into an image dataset consisting of real and fake spectrograms. Therefore, the images are labeled as real and fake, and is followed by a unique number to distinguish between them. Furthermore, Figure 1 shows a simple spectrogram generated from one of the real audio samples whereas Figures 2 and 3 show the Mel spectrogram generated by a real and fake audio sample from the dataset which has been used. For the Mel spectrogram, different hop sizes are also used. As the audio samples are more or less around 10 seconds, it is optimal to slice all the audios to the same size of 10 seconds which led to better Mel spectrograms. However, Mel spectrograms are created using different hop sizes and Mel scales. Later on, for creating the Mel spectrograms for the dataset, we used the same hop-size and Mel-scales. Figure 7 shows a simple spectrogram for a real audio.

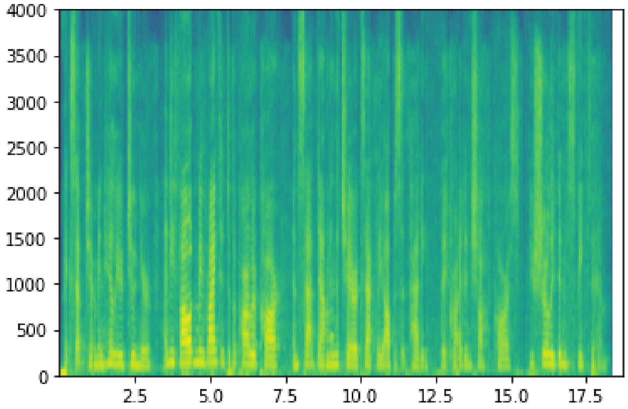

Figure 7. Simple Spectrogram For Real Audio

Furthermore, for the model, the best parameters needs to be selected to decide which parameter is giving the best training result. For that in the model selection process, the Mel spectrogram is taken with hop size 256 and Mel scale 10, which gave an optimal solution. Moreover, for the detection, the selected method is based on a Convolutional Neural Network with three layers consisting of different dimensions. Next, the dataset is created by using a NumPy array which stores the image and its label in an array index so that it can be accessed while training the data. However, it is also split into training and test datasets. Figure 8 and Figure 9 illustrates Mel Spectrogram of a real and a fake audio.

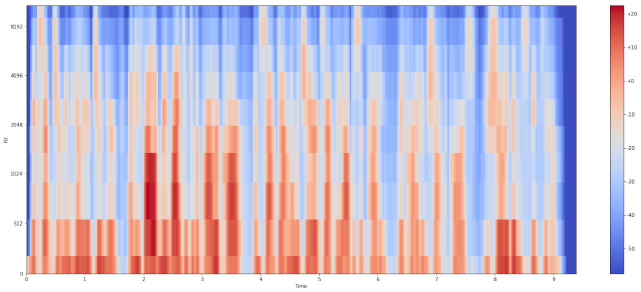

Figure 8. Mel spectrogram For Real Audio

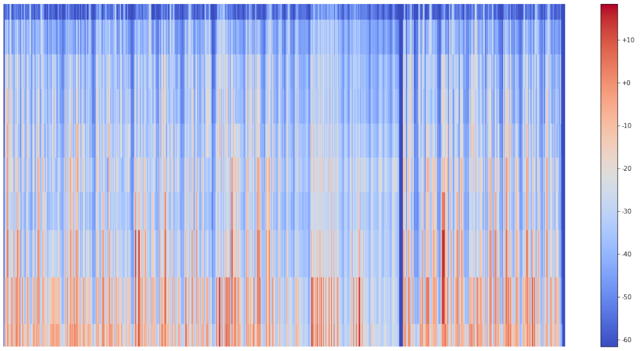

Figure 9. Mel spectrogram For Fake Audio

First of all, the first layer has the input dimension which is the same as the dimension of the Mel spectrogram. Convolutional 2D is used as the first layer with a dimension of 16 filters and a stride size of (3,3). Relu is used as an activation function for the first layer. After that, a Max Pooling layer is used which has a (2,2) dimensionality. This layer is followed by another Convolutional 2D layer with filter size 64 and stride size (3,3). A similar activation function is used for this layer as earlier. Again Max pooling is used for getting the most informative pixels from the images. This layer is followed by a similar Convolution 2D layer having dimensions of 64 and activation as relu. After the convolutional layers and Max Pooling, a Flatten layer is applied which is followed by a Dense layer consisting of 1024 dimensions which then is followed by another dense layer with 1 output and sigmoid activation function. Figure 10 illustrates the model and Table 2 shows the parameters of the model after compilation.

Table 2. Parameters of the Detection Model

Parameters	Number
Total params	287,921
Trainable params	287,921
Non-trainable params	201

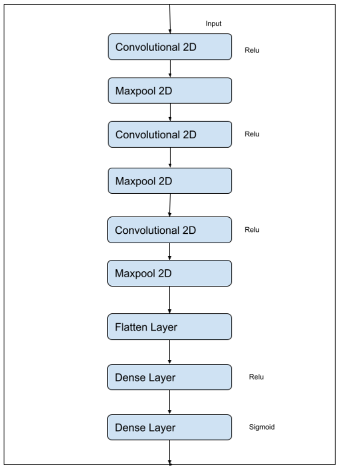

Figure 10. Detection Model Using CNN

The training is conducted in two different ways to determine if the model is the right choice. The cross-validation has been done using different model parameters and changing the input dimensions and also adding more hidden layers. For the initial phase of the training, only 50 sets of data have been used with 50% of data being fake and 50% being real. However, with only 15 epochs the model ran into convergence and it showed an accuracy over 99% which seemed inappropriate. So, the training set is increased to 150 and consists of 60% fake audios and 40% real audios which made the dataset a bit more biased. However, with 50 epochs now the model has an average loss of 2.975 which is binary cross-entropy. After testing it on 50 datasets the model scores around 81.76% accuracy to predict the outcome. Even with five convolutional layers, the model has approximately similar accuracy in testing although it has a slight performance increase in the training phase. However, as there is only a slight difference in the model's performance, it does not indicate the need for a more complex structure, the three-layer model is thought to be reliable.

4 Result and Evaluation

In this section, the results will be discussed along with different evaluation methods. The evaluation method is divided into two parts consisting of two different datasets for this. The first dataset consist of the generated audios from the software and the second dataset is from ASVspoof competition 2019. For the first part, the model was tested on the software-generated and ASVspoof evaluation data to get the accuracy, F1 Score.

Confusion Matrix: It is a matrix that summarizes the prediction results of a classification problem. Figure 11 shows a confusion matrix where we can see that vertical axis denotes the predicted class and the horizontal axis denotes the true class or label. [21]

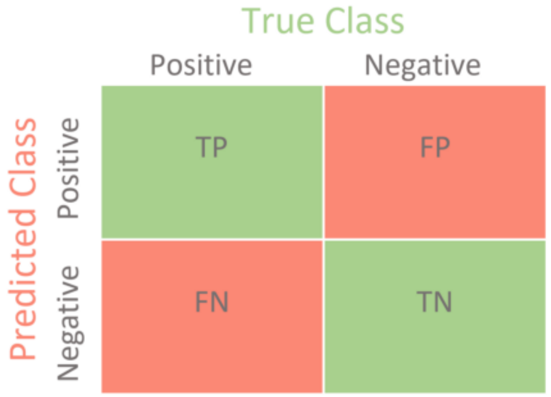

Figure 11. Confusion Matrix [4]

True Positive(TP) means the predicted and true class is positive.

False Positive(FP) means the true class is negative while it is predicted as a positive class.

False Negative(FN) means the true class is positive while the predicted class is negative.

True Negative(TN) means both true class and predicted class are negative.

Precision: It finds out how much percentage of positive predicted classes are actually positive. [22]

$Precision = \frac{TP}{TP+FP}$ [22]

Recall: Among the positive examples it calculated how much of the examples are actually from positive class. [22]

$Recall = \frac{TP}{TP+FN}$ [22]

F1 Score: It considers both precision and recall.

$F_1\ Score = \frac{(2\times Precision\times Recall)}{(Precision+Recall)}$

Accuracy: It calculates how much successful a model was in predicting everything in correct class, i.e., positive class is predicted to be positive and negative class is predicted to be negative.

$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$

4.1 Varying Dataset Size

In this section the authors focused on testing the model in different datasize and datasets.

4.1.1 Generated Dataset.

This dataset is the custom build dataset that the authors had worked on. The data was increased from 150 to 500 to see the prediction accuracy of the model. As the test data size increased the accuracy increased however after the evaluation of 500 data it decreased a bit. The average accuracy for the test was around 86% and the F1 score was 0.88. Table 3 demonstrates the Accuracy and F1 Scores,

4.1.2 ASVSpoof Dataset.

This dataset consists of Logical and Physical access datasets. The physical access dataset is taken from the sensory level and it contains the real-time spoof attacks on a sensory level. On the other hand, logical access data is generated from state-of-the-art technologies like the text to speech and voice conversion methodologies [8]. A similar model was applied to this dataset for calculating F1 Score and accuracy. The average accuracy for this data set has also been quite similar and it was around 83.3% and F1 score around 0.89. Table 4 shows the scores in detail.

[4] Image is taken from https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Table 3. Results Using Generated Audios

Data Size	TP	FN	FP	TN	Accuracy	F1-Score
150	70	10	10	60	0.86	0.87
200	100	20	5	75	0.87	0.88
250	120	5	15	100	0.88	0.90
300	177	30	22	71	0.82	0.86
400	234	13	36	117	0.87	0.90
500	300	30	42	128	0.85	0.88

Table 4. Results Using ASVSpoof Dataset

Data Size	TP	FN	FP	TN	Accuracy	F1-Score
200	140	24	18	18	0.79	0.86
500	349	30	42	79	0.85	0.90
800	601	30	76	93	0.86	0.91

4.2 Corrupted Dataset

However, another evaluation methodology was to see if the model can score similar accuracy when the real or bona fide data has some noise or fake audios in it. Therefore, in the audio generation method, the authors created generated audios and for the real audios, some noise was added. Also, some of the fake audios were included in the real audio dataset to measure the accuracy after training with this dataset. However, firstly with 150 data in hand around 5% of data was labeled incorrectly to check if the model scores similar accuracy. Surprisingly, the model got an accuracy of 82% with a F1 score of around 0.83. Furthermore, with 200, 300, and 500 data also maintaining the 5% ratio of incorrectly labeled data the model scored an accuracy of 79.4% on average with an average F1 score of 0.82. Table 5 shows an overview of the generated results.

Table 5. Result on Corrupted Dataset

Data Size	TP	FN	FP	TN	Accuracy	F1-Score
150	71	14	12	53	0.82	0.83
200	131	19	22	28	0.79	0.85
300	126	36	26	112	0.79	0.79

5 Conclusion

In this paper, the authors have proposed different ideas about the generation and detection of fake audios. Different experiments were conducted and not all ended up generating successful results. Finally, a decision was made to use “Real-Time Voice Cloning” [20] for the generation of audio. A deep learning framework (Convolutional Neural Network) was used for the effective detection of fake audios. The experiment result displayed accuracy of around 86 percent and F1 Score around 0.88 using the synthesized dataset. Furthermore, using ASVspoof 2019 dataset it provides an accuracy of 83 pecent. Overall, even a simple model like CNN can be used for the detection of fake audio but it is only suitable for very small scale detection.

6 Future Works

As this research shows that CNN can be used for Fake Audio detection however Generative adversarial networks (GAN) works really well for audio data. There are a lot of research going on on this field and the authors plan to build a model using GAN which can perform better.

References

[1] H. Dhamyal, A. Ali, I. A. Qazi, and A. A. Raza, "Fake audio detection in resource-constrained settings using microfeatures," *Proc. Interspeech 2021*, pp. 4149–4153, 2021.

[2] R. Wijethunga, D. Matheesha, A. A. Noman, K. D. Silva, M. Tissera, and L. Rupasinghe, "Deepfake audio detection: A deep learning based solution for group conversations." IEEE, 12 2020.

[3] A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, and S. Tubaro, ""hello? who am i talking to?" a shallow cnn approach for human vs. bot speech classification." IEEE, 5 2019.

[4] A. Luo, E. Li, Y. Liu, X. Kang, and Z. J. Wang, "A capsule network based approach for detection of audio spoofing attacks." IEEE, 6 2021.

[5] R. Wang, F. Juefei-Xu, Y. Huang, Q. Guo, X. Xie, L. Ma, and Y. Liu, "Deepsonar: Towards effective and robust detection of ai-synthesized fake voices," 5 2020.

[6] D. M. Ballesteros, Y. Rodriguez-Ortega, D. Renza, and G. Arce, "Deep4snet: deep learning for fake speech classification," *Expert Systems with Applications*, vol. 184, 12 2021.

[7] "Synthetic speech detection through short-term and long-term prediction traces," *EURASIP Journal on Information Security*, vol. 2021, 12 2021.

[8] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[9] R. A. M. Reimao, "Synthetic speech detection using deep neural networks," 2019.

[10] T. T. Nguyen, Q. V. H. Nguyen, C. M. Nguyen, D. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection: A survey," 9 2019.

[11] A. Karandikar, "Deepfake video detection using convolutional neural network," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, 4 2020.

[12] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatic spoofing detection." ISCA, 8 2017.

[13] T. Liu, D. Yan, R. Wang, N. Yan, and G. Chen, "Identification of fake stereo audio using svm and cnn," *Information*, vol. 12, 6 2021.

[14] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 132–137.

[15] J. Yi, Y. Bai, J. Tao, Z. Tian, C. Wang, T. Wang, and R. Fu, "Half-truth: A partially fake audio detection dataset," *arXiv preprint arXiv:2104.03617*, 2021.

[16] H. Ma, J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Wang, "Continual learning for fake audio detection," *arXiv preprint arXiv:2104.07286*, 2021.

[17] H. Malik and R. Changalvala, "Fighting ai with ai: Fake speech detection using deep learning," in *Audio Engineering Society Conference: 2019 AES International Conference on Audio Forensics.* Audio Engineering Society, 2019.

[18] I. Shafkat. (2018) Intuitively understanding variational autoencoders. [Online]. Available: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

[19] J. Jordan, Mar 2018. [Online]. Available: https://www.jeremyjordan.me/variational-autoencoders/

[20] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," 6 2018.

[21] J. Brownlee, "https://machinelearningmastery.com/confusion-matrix-machine-learning/," Nov 2016. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/

[22] J. Mohajon, "Confusion matrix for your multi-class machine learning model," May 2020. [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

A Contribution of Writing

Table 6. Contribution in Report

Team Members	Contribution in Report
Abdullahil Kafi	Literature Review, Audio Generation, Audio Detection, Result and Evaluation
Israt Nowshin	Abstract, Introduction, Literature Review, Audio Generation
Manali Thakur	Literature Review, Audio Generation, Future Work
Satish Khadka	Literature Review, Audio Generation, Conclusion