FAKE AUDIO SPEECH DETECTION

Shilpa Lunagaria[1], Mr. Chandresh Parekh[2]
[1]*Student at M. Tech, School of Information Technology & Cyber Security,*
[2]*Dean, School of Information Technology & Cyber Security,*
Raksha Shakti University, Lavad, Dahegam, Gandhinagar, Gujarat, India

Abstract- **Advantage of deep learning is very vast in these days. Training of images, videos, audio and test it according to requirement has become simple and user friendly. Still there are some disadvantages and risks. In this paper, we will discuss about audio that created by deep fakes it is very popular word in latest technology fake audio not only terrifying but actually beginning to happen. Fake audio can be used for malicious purposes which affect directly or indirectly human life. For an example, google map use deep learning based navigation; if it modifies then we will be misdirected. In this work, many papers on how to differentiate real or fake audio has been referred. Python and deep learning has been used and implemented to achieve the aim. Audio files or video file are being used as an input of this work then model has been trained for uniquely identify features for voice creation and voice detection. Deep learning technique is used to find accuracy between real and fake.**

Index terms- **voice creation; voice detection; deep learning; python.**

I. INTRODUCTION

Deep leaning uses as generate realistic images audio, videos in current days. News Channels are the real example of deep learning. It gives some real or fake news but people don't know whether it is original or fake. In this paper, two main things are required to achieve the aim that one is trained data set like fake audio clips (trained data using AI or Deep Learning) or ASV Proof Challenge[1] can help to provide different language with different voice tone and second thing is any deep learning and python technique that calculate function to
differentiate real or fake audio speech. There are two types of ASV dataset; text dependent and text independent. Language tone and content with audio characteristics are required in text dependent; it uses i-vector Probabilistic Linear Discriminate Analysis. Text independent works with different languages to extract feature using Mel Frequency coefficients which will extract features of audio. To identify real or fake audio clip using vulnerable system, we are able to get 80 to 85 percentage accuracy. Python library such as numpy, pandas or librosa has been used for audio analysis. The audio data has been trained using deep learning.

II. DEEP FAKE VOICE GENERATION

The state-of-the-art AI approaches proved to be successful in capturing the linguistic details and producing a smooth, natural human sound (e.g. DeepVoice3 [2] or Tacotron2 [3]). By the help of holy deep learning, it is possible to extract the language into a meta representation and synthesize the audio by using this representation.

Voice cloning technology is relatively accessible on the Internet today. Montreal-based AI startup Lyrebird provides an online platform that can mimic a person's mimic's speech when trained on 30 or more recordings [4]. Baidu last year introduced a new neural voice cloning system that synthesizes a person's voice from only a few audio samples [5].

A text-to-speech (TTS) synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module. There is not yet a consensus on the optimal neural network architecture for TTS. However, sequence-to-sequence models have shown promising results. An end-to-end generative text-to-speech model synthesizes speech directly from characters. Input format for single speaker TTS systems is <text, audio> pairs and <text, audio, speaker> pairs is for multi- speaker TTS systems. Text Preprocessing has been performed as uppercase all characters in the input text. Four different word separators, indicating slurred-together words, standard pronunciation and space characters, a short pause between words, and a long pause between words. For example, the sentence "Either way, you should shoot very slowly," with a long pause after "way" and a short pause after "shoot", would be written as "Either way%you should be obtained through either maual labeling or by shoot/very slowly%." with % representing a long pause and / representing a short pause for e ncoding convenience. The pause durations can estimated by a text-audio aligner such as Gentle

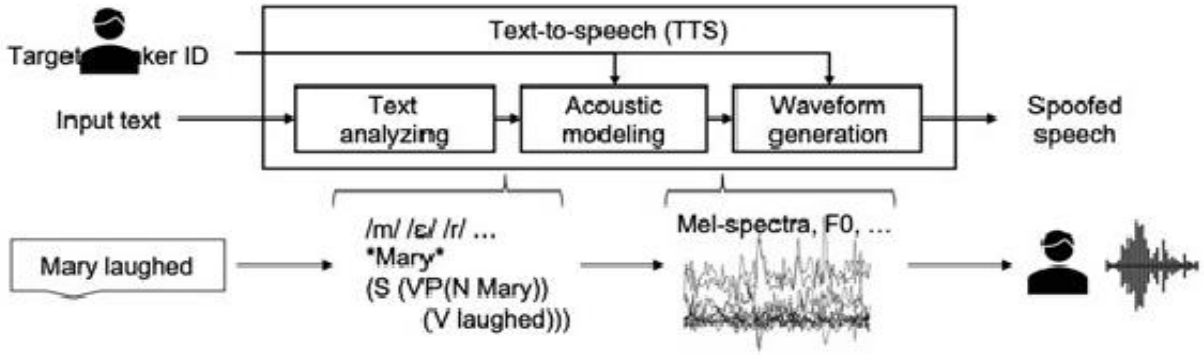

Figure 1. Voice Generation

Voice Conversation (VC) is the process of converting the source speaker's voice to a sound similar to the target speaker's voice. VC deals with the information that relates to the segmental and suprasegmental features and keep the language content similar. Earlier studies include statistical techniques, such as Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), unit selection, principal component analysis (PCA), and Non-negative matrix factorization (NMF) for VC task. Recently, DNN, Wavenet, and GAN represent a technology leap.

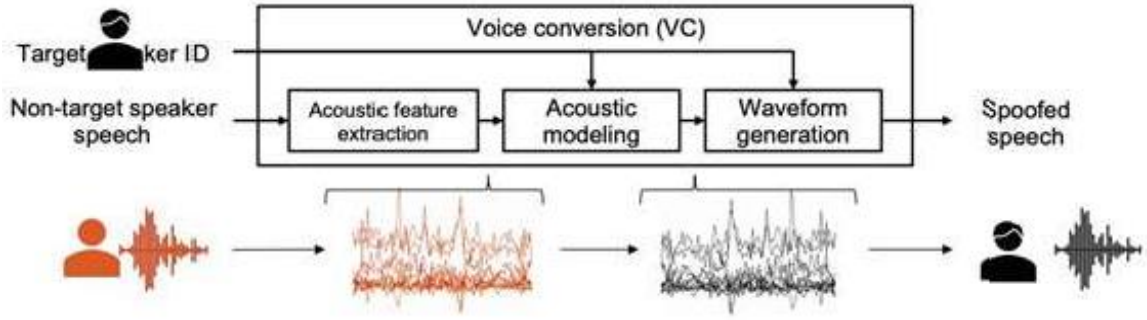

Figure 2. Voice Conversation way is take input Wave

There are several methods available for Neural Voice Cloning such as Whole Model Adaptation, Speaker Embedding Adaptation and voice conversion. Whole Model Adaptation do fine- tuning a multi-speaker generative model; where in Speaker Embedding Adaptation, a model has been independent and it gives output B wave see the (figure. 3). Also we can use Net1 is a classifier in which we perform process like wav-> spectrogram-> mfccs->phoneme dist. Net1 classifies spectrogram to phonemes that consists of 60 English phonemes trained to predict new speakers embedding which will apply to a multi-speaker generative model. at every timestep. For each, the input is log magnitude spectrogram and the target is phoneme dist. Objective function is cross entropy loss. Voice conversation can be done by different way like, non parallel requirement and second TIMIT dataset used that cont ins 630 speakers utterances and corresponding phones that speaks similar sentences. It gives 70% test accuracy.

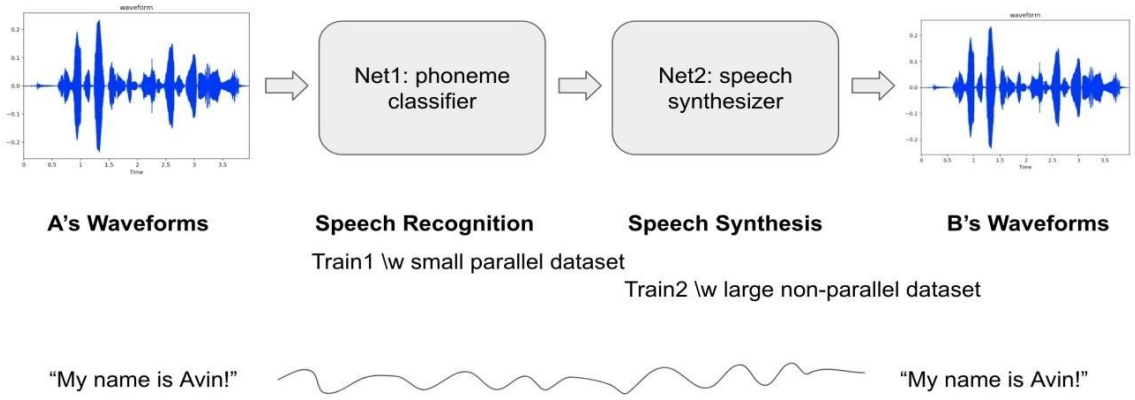

Figure 3. Voice Conversation(2nd method)

And Net2 is a synthesizer that contains net1 as a sub-network .In which process will be like wav-> spectrogram-> mfccs->phonemedist.-> spectrogram-> wav.Net2 synthesizes the target speaker's speeches .the input/target is a set of target speaker's utterances .Since Net1 is already trained in previous step. Loss is reconstruction error between input and target. Datasets will be target1(anonymous female)arctic dataset that is public and other one is target2(kate winslet)over 2 hours of audio book sentences read by her that is private. Griffin Lim reconstruction when reverting wav from spectrogram.

A generative model can be trained from scratch with a large amount of audio samples, here, focus is on voice cloning of a new speaker with a few minutes or even few second data. It is challenging as the model has to learn the speaker characteristics from very limited amount of data, and still generalize to unseen texts.

To create a fake voice of President Trump, Deep Learning based Text to Speech synthesis models like DeepVoice3 [2], Tacotron2 [3], VoiceLoop [7] and other similar systems [5-13] can be used. The (Text, Utterance) pairs of President Trump has been passed as input to these models. Acoustic model will learn to generate the spectrograph form character text and vocoder model will learn to generate audio wave from the given spectrograph.

For speaker adaptation to a portion of the model, selected layers of the pre-trained TTS model has been trained with the PT data. This method reduces the training time and also reduces the requirement of large size of training data. Speech embedding vector of PT from an encoder has been got in speaker adaptation only to the embeddings of PT. The TTS model is trained on the (Text, Utterance, Speaker Embedding) pairs. PT embeddings have been passed to the model and model synthesize the voice conditioned on PT embeddings.

We require some essential components like Synthesizer which is also known as acoustic model. Sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014) encode a variable- length input into hidden states, which are then processed by a decoder to produce a target sequence. An attention mechanism allows a decoder to adaptively select encoder hidden states to focus on while generating the target sequence (Bahdanau et al., 2015). Attention-based sequence-to-sequence models are widely applied in machine translation (Bahdanau et al., 2015), speech recognition (Chorowski et al., 2015), and text summarization (Rush et al., 2015). In speech synthesis, generative models can be conditioned on text and speaker identity [e.g., Arik et al., 2017b]. While text carries linguistic information and controls the content of the generated speech, speaker identity captures characteristics such as pitch, speech rate and accent. Then second component is vocoder, which is also known as waveform generator. Neural vocoder has the function of taking the lossy spectrograms (the ones produced by the first system) and overlaying an additional layer of naturalness. This process transforms the spectrograms from something utilitarian into something that's a work of art. In technical terms, the neural vocoder is tasked to invert these spectrograms back to audio while reconstructing the phase. e.g. World, Griffin – Lim, Wavenet, WaveRNN, WaveGlow. And third component is Encoder which is GE2E and most important is training datasets for single speaker and multi speaker for sing le speaker we used Speech dataset from Keith Ito require for training

24 hours and for multi speaker Libri speech require 1000 hours that is clean and noisy sets and 2484 speakers. A single speaker model can require ∼20

hours of training data [e.g. Arik et al., 2017a, Wang et al., 2017], while a multi-speaker model for 108 speakers [Arik et al., 2017b] requires about 20 minutes data per speaker. Evaluation can be done using mean opinion score or Equal error rate

1. DEEPFAKE VOICE DETECTOR

Speaker recognition usually refers to both speaker identification and speaker verification. A speaker identification system identifies who the speaker is, while automatic speaker verification (ASV) system decides if an identity claim is true or false. The ASV systems are vulnerable to various kinds of spoofing attacks, namely, synthetic speech (SS), voice conversion (VC), replay, twins, and impersonation [16].

A general ASV system is robust to zero-effort impostors; they are vulnerable to more sophisticated attacks. Such vulnerability represents one of the security concerns of ASV systems. Spoofing involves an adversary (attacker) who masquerades as the target speaker to gain the access to a system. Such spoofing attacks can happen to various biometric traits, such as fingerprints, iris, face, and voice patterns [14-16]. We are focusing only on the voice-based spoofing and anti-spoofing techniques for ASV system. The spoofed speech samples can be obtained through speech synthesis, voice conversion, or replay of recorded speech.

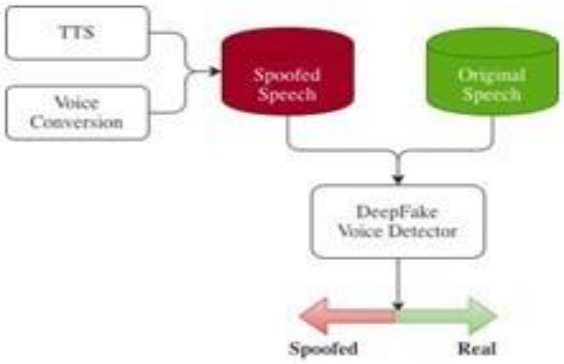

Figure 4. Framework

To discern between real and fake audio, the detector uses visual representations of audio clips called spectrograms, which are also used to train speech synthesis models.

Google's 2019 ASV Spoof dataset [17] contains over 25,000 clips of audio, featuring both real and fake clips of a variety of male and female speakers.

Raw audio is preprocessed and converted into a mel-frequency spectrogram — this is the input for the model. The model performs convolutions over the time dimension of the spectrogram, then uses masked pooling to prevent over fitting. Finally, the output is passed into a dense layer and a sigmoid activation function, which ultimately outputs a predicted probability between 0 (fake) and 1 (real).

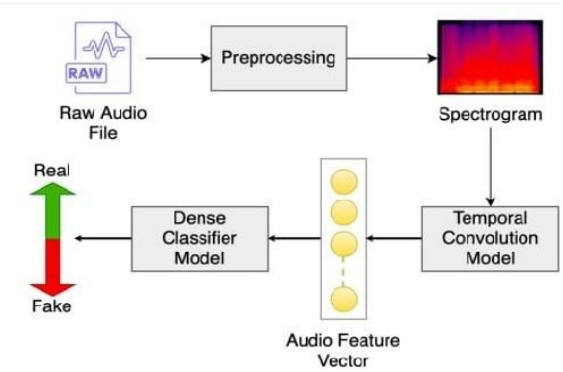

Figure 5. Model

The baseline model achieved 99%, 95%, and 85% accuracy on the train, validation, and test sets respectively. The differing performance is caused by differences between the three datasets. While all three datasets feature distinct and different speakers, the test set uses a different set of fake audio generating algorithms that were not present in the train or validation set.

2. Experiment Details in Tables with Source

This table explains sources from github that uses in my experiment or use as a source code .series of table describe step by step procedure to accomplish my task.

	Model	Synthesizer	Vocoder	Link
1	**TTS-M1**	Tacotron2	WaveGlow	https://github.com/NVIDIA/tacotron2/
2	**TTS-M2**	Tacotron	Griffin-Lim	https://github.com/keithito/tacotron
3	**TTS-M3**	Tacotron	WaveRNN	https://github.com/mozilla/TTS/
4	**TTS-M4**	Tacotron2	WaveRNN	https://github.com/mozilla/TTS/
5	**TTS-M5**	Tacotron2	PWGAN	https://github.com/mozilla/TTS/
6	**TTS-M6**	Tacotron2	MelGAN	https://github.com/mozilla/TTS/
7	**TTS-M7**	DeepVoice3	Griffin-Lim	https://github.com/r9y9/deepvoice3_pytorch
8	**TTS-M8**	DeepVoice3	WORLD	https://github.com/hash2430/dv3_world
9	**TTS-M9**	Tacotron2	WaveNet	https://nvidia.github.io/OpenSeq2Seq/html/index.html
10	**TTS-M10**	DC-TTS	Griffin-Lim	https://github.com/Kyubyong/dc_tts
11	**TTS-M11**	ESPNet	PWGAN	https://github.com/kan-bayashi/ParallelWaveGAN
12	**TTS-M12**	ESPNet	MelGAN	https://github.com/kan-bayashi/ParallelWaveGAN
13	**TTS-M13**	VoiceLoop	WORLD	https://github.com/facebookarchive/loop
14	**TTS-M14**	FastSpeech	MelGAN	https://github.com/espnet/espnet

Table 1: Text to speech

	Model	Architecture	Link
1	**AST-M1**	RandomCNN	https://github.com/mazzzystar/randomCNN-voice-transfer
2	**AST-M2**	VGG16-1DCNN	https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/
3	**AST-M3**	CBHG-DNN	https://github.com/andabi/deep-voice-conversion

4	**SV2TTS**	GE2E (encoder) + Tacotron (synthesizer) + WaveRNN (vocoder)	https://github.com/CorentinJ/Real-Time-Voice-Cloning
5	**AST-M4**	DeepVoice3 Adapter	https://sforaidl.github.io/Neural-Voice-Cloning-With-Few-Samples/

Table 2: Audio Style Transfer

	Model	**Architecture**	**Link**
1	**VC-M1**	MelGAN-VC	https://github.com/marcoppasini/MelGAN-VC
2	**VC-M2**	ASR-TTS	https://github.com/espnet/interspeech2019-tutorial
3	**VC-M3**	ESPNet Merlin	https://github.com/r9y9/icassp2020-espnet-tts-merlin-baseline

Table 3: Voice conversation

	Model	**Architecture**	**Link**
1	**AD-M1**	Temporal Convolution	https://github.com/dessa-oss/fake-voice-detection
2	**AD-M2**	Encoding Similarity Match	https://github.com/resemble-ai/Resemblyzer
3	**AD-M3**	CQCC GMM + MFCC ResNet + CQCC ResNet	https://github.com/BhusanChettri/ASVspoof2019
4	**ASSERT**	Squeeze-Excitation and Residual networks	https://github.com/jefflai108/ASSERT
5	**AD-M4**	ResNet34	https://github.com/rahul-t-p/ASVspoof-2019
6	**AD-M5**	CPC	https://github.com/jefflai108/Contrastive-Predictive-Coding-PyTorch
7	**AD-M6**	GMM-MFCC	https://github.com/elleros/spoofed-speech-detection
8	**AD-M7**	CycleGAN	https://github.com/kstoneriv3/Fake-Voice-Detection

Table 4:DeepFake Detection

III. CONCLUSION

Any audio clip or YouTube video id can be used as an input and put in module. It will give the accuracy between real and fake content. This has been achieved by deep learning technique and python library.

REFERENCES

[1] Massimiliano Todisco and Xin Wang and Ville Vestman and Md Sahidullah and Hector Delgado and Andreas Nautsch and Junichi Yamagishi and Nicholas Evans and Tomi Kinnunen and Kong Aik Lee, "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection", arXiv:1904.05441, 2019.

[2] Wei Ping and Kainan Peng and Andrew Gibiansky and Sercan O. Arik and Ajay Kannan and Sharan Narang and Jonathan Raiman and John Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning", arXiv:1710.07654, 2017.

[3] Jonathan Shen and Ruoming Pang and Ron J. Weiss and Mike Schuster and Navdeep Jaitly and Zongheng Yang and Zhifeng Chen and Yu Zhang and Yuxuan Wang and RJ Skerry- Ryan and Rif A. Saurous and Yannis Agiomyrgiannakis and Yonghui Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions", arXiv:1712.05884, 2017.

[4] Francesca Crist. WIRED Insider. (2018, October). How Lyrebird Uses AI to Find Its (Artificial) Voice. Retrieved from https://www.wired.com/brandlab/2018/10/lyrebird-uses-ai-find-artificial-voice.

[5] Sercan O. Arik and Jitong Chen and Kainan Peng and Wei Ping and Yanqi Zhou, "Neural Voice Cloning with a Few Samples", arXiv:1802.06006, 2018.

[6] McAuliffe, Michael & Socolof, Michaela & Mihuc, Sarah & Wagner, Michael & Sonderegger, Morgan. (2017). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. 498-502. 10.21437/Interspeech.2017-1386.

[7] Yaniv Taigman and Lior Wolf and Adam Polyak and Eliya Nachmani, "VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop", arXiv:1707.06588, 2017.

[8] Sercan O. Arik and Mike Chrzanowski and Adam Coates and Gregory Diamos and Andrew Gibiansky and Yongguo Kang and Xian Li and John Miller and Andrew Ng and Jonathan Raiman and Shubho Sengupta and Mohammad Shoeybi, "Deep Voice: Real- time Neural Text-to-Speech", arXiv:1702.07825, 2017.

[9] Yuxuan Wang and RJ Skerry-Ryan and Daisy Stanton and Yonghui Wu and Ron J. Weiss and Navdeep Jaitly and Zongheng Yang and Ying Xiao and Zhifeng Chen and Samy Bengio and Quoc Le and Yannis Agiomyrgiannakis and Rob Clark and Rif A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis", arXiv:1703.10135, 2017.

[10] Tomoki Hayashi and Ryuichi Yamamoto and Katsuki Inoue and Takenori Yoshimura and Shinji Watanabe and Tomoki Toda and Kazuya Takeda and Yu Zhang and Xu Tan, "ESPnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text- to-Speech Toolkit", arXiv:1910.10909, 2019.

[11] Jemine, Corentin, "Automatic Multispeaker Voice Cloning" (Master thesis), http://hdl.handle.net/2268.2/6801, 2019.

[12] Nal Kalchbrenner and Erich Elsen and Karen Simonyan and Seb Noury and Norman Casagrande and Edward Lockhart and Florian Stimberg and Aaron van den Oord and Sander Dieleman and Koray Kavukcuoglu, "Efficient Neural Audio Synthesis", arXiv:1802.08435, 2018.

[13] Ye Jia and Yu Zhang and Ron J. Weiss and Quan Wang and Jonathan Shen and Fei Ren and Zhifeng Chen and Patrick Nguyen and Ruoming Pang and Ignacio Lopez Moreno and Yonghui Wu, "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis", arXiv:1806.04558, 2018.

[14] Li Wan and Quan Wang and Alan Papir and Ignacio Lopez Moreno, "Generalized End-to-End Loss for Speaker Verification", arXiv:1710.10467, 2017.

[15] T. Kinnunen, K.-A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, D.-A. Reynolds, "t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification", Proc. Odyssey 2018 - The Speaker and Language Recognition Workshop.

[16] Kamble, Madhu & Patil, Hemant. (2018). A Survey on Replay Attack Detection for Automatic Speaker Verification (ASV) System. 10.23919/APSIPA.2018.8659666.

[17] Yamagishi, Junichi; Todisco, Massimiliano; Sahidullah, Md; Delgado, Héctor; Wang, Xin; Evans, Nicolas; Kinnunen, Tomi; Lee, Kong Aik; Vestman, Ville; Nautsch, Andreas. (2019). ASVspoof 2019: The 3rd Automatic Speaker Verification Spoofing and Countermeasures Challenge database, [sound]. University of Edinburgh. The Centre for Speech Technology Research (CSTR).
https://doi.org/10.7488/ds/2555