Podcasts Deepfake Audio with Multilingual Representation

Sai Teja Gudipati
Electrical and Computer Engineering
University of waterloo, Canada
stgudipa@uwaterloo.ca

Shadman Raihan
Electrical and Computer Engineering
University of waterloo, Canada
s2raihan@uwaterloo.ca

Somesh Kumar Gupta
Electrical and Computer Engineering
University of waterloo, Canada
sk2gupta@uwaterloo.ca

Abstract - In this realm of media content consumption, podcasts play a prominent role in its exponential growth. So, we thought that the same podcasts content should be reachable to all diversities of people with their favorite "artistic voice and language", irrespective of which voice the original content is delivered. To achieve this with robust and reliability, Convolutional Neural Networks, NLP and GAN's are key building blocks of our model. There are three key stages involved, first the raw audio to text conversion, second language modelling and third inversion back to audio. As this case is speech, mel-spectogram and aligned linguistic features are the intermediate representations with phoneme boundaries. Evaluation of the model is done using Mean Opinion score metric(MOS).

Index terms : CNN, NLP, GAN

I. INTRODUCTION

Deepfake (stemmed from deep learning and fake) is a powerful artificial intelligence and machine learning technique which is used to generate or manipulate audio and video content and has a very good potential to deceive people. This process is based on involving Deep learning techniques to train the generative neural networks which are Generative adversarial networks (GANs) and autoencoders. Deepfakes[1]-[4]. method usually need lot of data samples to train the model which creates a realistic content experience. Most common Public figures data is largely available online, so these people are the initial targets of these Deepfakes[5]. Sometimes these can cause friction or political tensions which effect the election campaigning results by creating chaos in the financial marketplace or generate fake news. If the image or a video footage is considered, this technique can generate the unusual satellite imagery which can even mislead the military troops which has been developed in the late 19th century and has been applied to motion pictures in recent years.

Using this domain knowledge, we wanted to explore the new possibilities which can impact the society and people lives in a positive way. Our motivation behind this project is to make podcasts content reach to all diversities of people and listen them with their favourite artistic voice and their own language irrespective of which voice the original content is delivered. From the Q1 2020 statistics on podcasts from statista.com, they reported the podcasts growth has been increasing rapidly in the countries like South Korea(53%), Spain(39%), USA(35%), Canada (29 %) based on the survey conducted in major cities. But people in nonmetropolitan cities and technologically advancing countries are still below 3% in podcasts consumption and production. If these can be changed and same content is delivered and consumed in their own way, that we consider our project has successfully achieved its goal.

With the use of traditional machine learning techniques in speech recognition[6] we can feed the sound recordings simply into the neural network and train them to produce the text output[7], but here there is a problem of varying speed. As we know the sound waves are one dimensional and considering the particular moment of time, we will have a single value based on frequency measure of the wave. So, if we turn these sound waves into numbers, we can record the height of these waves by spacing the points equally. This method is called as sampling the wave which is an uncompressed wave of audio file. As the podcasts are of human speech the sampling rate of 16000hz is sufficient to cover the entire frequency range of the human speech. We have trained our model to detect English language with an approximate vocabulary of 50000 words. These contain small vocabulary which of tens of words, medium vocabulary which is hundreds of words and large vocabulary which is ten thousand of words[8]. Once the model converts the speech from the audio waveform to text the next process is conversion in to multi language modelling[9]-[10].

In the recent years we see that many unsupervised learning algorithm representations have been significantly improved and they correlate with the phenome structure there by improving the performance of recognition in speech models. While performing this experiment we have got some of the representations which are derived from the largest pretrained dataset dataset model from LibriSpeech, Librivox datasets. We train this model on 3 different languages to convert from English audio to Hindi, Telugu and Tamil. Once the translation is done the next step is to convert the language model into a celebrity voice using text to speech conversion model. The speech conversion is done using Fast Fourier transform which can analyse the frequency content of the signal and get the segments of signal called as spectrogram.

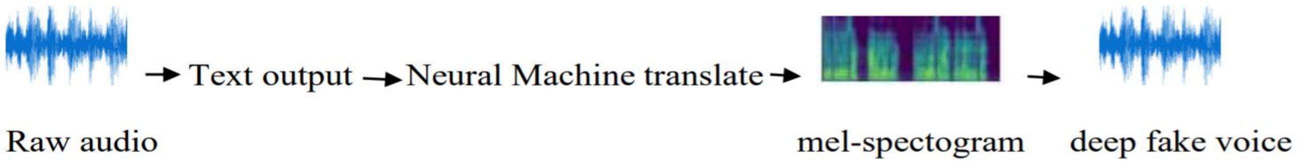


Figure 1 : Block diagram of the system

II. LITERATURE REVIEW

With the advancements in machine learning and deep learning the speech recognition models have been vastly outspread, So we have taken the results from 2012 where actually the era of deep learning has been elevated. For an overall view of the topic we have presented a graph showing the results published on speech recognition and translation models from various journals and google scholar databases.

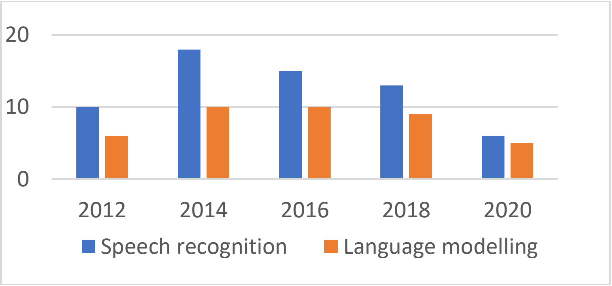


Figure 2 : Survey on no. of papers published on speech recognition and language modelling algorithms from 2012 to 2020.

Google wavenet [11] was designed with autoregressive and fully probabilistic model which has predictive distribution of each sampled audio and has been rated very close to native human performance. The model was built using causal convolutional architecture without any recurrent connections. This wavenet architecture was able to model the speech from multiple speakers by the use of probabilistic conditioning on one hot encoding of the speaker. The designed components can be used for conditional as well as unconditional audio generation. This Text to Speech algorithms was able to produce a Mean opinion score of 0.53 with English and 0.46 with Mandarin.

Deepvoice2 [12] an TTS standalone algorithm designed using deep neural networks which is an end to end neural synthesis architecture. There are 5 building blocks in this model where grapheme to phenome model and the audio synthesis model are key components. This has been implemented using very high-speed inference kernels for both GPU as well as CPU. TitanX Maxwell GPU is used to train over 20000 iterations to produce and phoneme error rate of 6.2% and the word error rate of 26.4%. While the performance od audio synthesis is optimized a Mean opinion score of 4.75 is resulted using ground truth 48KHz recordings.

Multilingual speech recognition model [13] is developed over the traditional architectures where the sub-word units are word lexicons are challenging on multiple language models as they are language specific. This is a sequence to sequence model consisting of encoder, decoder and attention network. The network configuration includes a 5-layer encoder model with 750 BiLSTM cells and a 2 layered Decoder model with 1024 LSTM cells in each of the layers. This has been implemented on tensorflow GPU and been trained using asynchronous gradient descent method with a learning rate of 1e-6 for the linguistic models. However, this model fails to handle code switching capability and results an incorrect grapheme in the output.

MelGAN [14] architecture was designed to train Generative networks reliably for generating a high-quality coherent waveforms with minimal training techniques. This model was trained on GTX 1080Ti Gpu using pytorch implementation. This has two modules a generator and a discriminator and a fully convolutional non autoregressive model which has 4.26M parameters. This achieved a MOS of 1.33 with spectral normalization and 3.04 with weight normalization while training with 500 epoch each.

Robust Multilingual speech representation[9] an unsupervised algorithm was trained with 8000 hours of noisy and diverse speech dialects and obtained some consistent results over 25 different phonetic languages and some of the low resource languages.

Deepfakes[1] with a variant of deep neural networks combined with autoencoder has been applied widely for the image compression and dimensionality reduction which can create a output of 128x128 and 512x512 image resolutions. Some of the recent tools are Faceswap, Fewshot, Deepfacelab. Dfaker etc. These can create an hurdle for forensic department because the professionals also cannot properly testify these algorithms as many of these are blackbox models.

Tacotron2 and waveglow [15] a sequence to sequence models which can synthesis the speech without any patterns or rhythms of the speech. These are implemented using the dropout layers instead of Zoneout for regularization of LSTM layers. This uses 512 residual channels to squeeze the vector operations.

SpecAugment ASR [16] an augmentation based speech recognition system. This is been inspired by cutout **algorithm in CV** and frequency masking techniques. The model when trained on LibreSpeech was able to achieve word error rate of 2.75 % and tested on other set and achieved 6.9% as the rate. The performance is been improved when trained on the much larger networks.

III. PROPOSED SYSTEM DESIGN

Our system design constitutes of three blocks, first the conversion of raw audio waveform to text, second natural language modelling and third inversion back to audio waveform. We will discuss each block with their inputs and outputs.

3.1. Raw Audio waveform as the input and text as the output. For this stage we are taking the publicly available audio podcasts and extracting the text from it using the below structure.

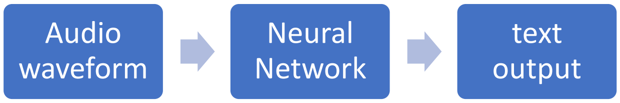


Figure 3: Block diagram of the speech to text model

As at the output of the whole system is audio, so the text generated at this stage should be with high precision without any occurrence of loss. And the challenge here is the speed of generation of generation text. From the phenomics point of view the same word can be spelled with too fast pace and also too slow pace. The system understands difference between two sounds and generates the output based on context of the word used. For example, the word “good morning” can be spelled very slowly so the model picks it and generates it as “gooood moorning” which is helpful for us when the inversion steps takes place. To tackle this, we need to design some deep neural network architecture which reduces the word error rate.

From the literature review we have seen that there are many acoustics models, with this knowledge we designed a sequence to sequence model. This consists of an encoder and a decoder network. This basic encoder module takes input and passes it on to the neural network to map the feature f^{et} ,and extract the text from it. Once the text is extracted it goes to a spell checker module to correct all the misspelled words.

This speech recognition can be categorized into two types, they are connected words which are spelled together with the minimum pause in between them and the isolated words which have pauses on both the sides of the sample. If the model detects the pause in between the words spelled the text outputs, it as a space between the word. This requires the model to be trained during which it can learn the patterns in the words of different speech. Once all these patterns are learnt these are stored in the memory cells as the templates. Once the patterns are learned, they should be recognized when a reference pattern is from the trained set.

The underlying principle involved here is analysing the speech patterns and generate the required features from it. . There are 3 key stages involved here:-

a) Speech analysis

b) Feature extraction

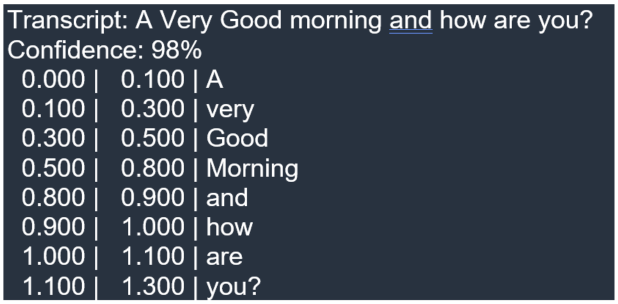
c) Text modelling.

We have modelled the system to recognize the time stamps of the word used. With this time offsets we get an exact time period of the word used in the sentence and this gives the starting and ending times of the word spelled in the audio. These will also be helpful in long duration audio files to search for a specific set of words. When there is no audio sample at some particular point the system takes the default offset of 300ms.

During this process there is also a challenge of recognizing the words with similar frequency such as dear and deer, which most often misspelled in the output text. For this we have used a technique of context recognition and the model is trained with whole vocabulary of 50000 english words. For example, if the context is regarding some animal behaviour the system takes “deer” as the audio input and outputs it. So, the above mentioned two features are key aspects of our speech to text model.

From testing the model with different kinds of audio waveforms, it is able to deliver the text output with latency of less than100ms

This particular system has many applications in recent times some of them are for visually challenged people can input the speech to the system so the system automatically converts the particular speech into a system command with very low latency of 30ms. These can also be implemented in educational institutions where the instructor lecture can be transcribed into text for students can directly sent to their emails. In the telecommunication sector this system model helps to store all the voice data in the form of text and will be helpful for analysing future queries. Now a days all the social media platforms are embedding this model into their system.



```
Transcript: A Very Good morning and how are you?
Confidence: 98%
 0.000 |  0.100 | A
 0.100 |  0.300 | very
 0.300 |  0.500 | Good
 0.500 |  0.800 | Morning
 0.800 |  0.900 | and
 0.900 |  1.000 | how
 1.000 |  1.100 | are
 1.100 |  1.300 | you?
```

Figure 4 : Time offsets of the words transcribed

Once after all the modelling and output text is generated then it is fed into natural language modelling block where the process of text translation and language processing takes place.

3.2 Second stage of the process is neural machine translate .

With the recent advancement in neural machine translation, there are some challenges associated with the speech recognition supporting multiple challenges. Some of them are sub-word unit, word lexicons and inventories which are specific to the particular language. This Language acoustic model takes lot of learnings from specific language pronunciation models in which these models already know the language of speech. This model has been trained on 3 different language models, they are Hindi, Telugu and Tamil. But the systems performs in single way translation i.e. only from English to Hindi or English to Telugu or English to Tamil. As the speech model is trained on one particular language.

This is a sequence to sequence model consisting of 3 blocks encoder network, attention network and a decoder network for the prediction of the acoustic phonemes. This encoder network is designed using Layered Bidirectional Recurrent neural networks which reads the feature ($i = i_1, i_2, i_3,\ldots i_k$) and those of the output features are ($j = j_1, j_2, j_3, \ldots j_k ,$).

The same architecture is designed for the decoder network with layered single directional Recurrent neural networks where the probability of the output sequence is calculated as k values as:

$$P(k \mid i) = P(k \mid j) = \prod_{t=1}^{T} P(\, i_t \mid j,\ i < t\,)$$

A little overview of what is RNN – These networks are particulary structured to take the input as a sequence of text and generate the same sequence of text as the output. The recurrence nature of these networks are due to the hidden layers in the network have a loop system where the output of one cell and the state of each cell from every time stamp becomes the input to the next cell. Where it is completely different from conventional CNNs.

There are 4 key stages in this model pipeline. They are preprocessing the data, modelling the data, prediction od the data and iteration of the architecture. In the preprocessing stage the data is loaded into the model and it is examined. The n the data cleaning along with padding and tokenization of the data takes place here. Once this is done, in the modelling stage building, training and testing the model takes place. After this in the prediction stage generates the translation of the text output from the previous stage to specific English to Hindi or English to Telugu and these outputs are then compared with ground truth translation schemes. Once the output is generated it passes onto iteration phases where the model is iterated with different architectural changes.

For implementing this we use Tensorflow for the backend processing and keras for frontend processing. As the keras library is very widespread and simpler to use, thereby the model can be designed with more ease. But there are some minor drawbacks to this keras models while considering the customization, but these drawbacks would not affect much in building the design.

a) Preprocessing the data

In the preprocessing the data the sampled inputs sequences are in English and outputs are language specific. As the data obtained is free of HTML tags, stop words, so there is no need to clean the data and all the data is already in lowercase. This data needs to tokenized, that means assigning the numerical vector values to the text which allows the deep neural network to perform the operations. Once the tokenization is done every word in the text output is assigned to a unique ID.

It creates index of words where each sentence is converted into a vector value. Then these sequence of unique word ID are fed into the model. The padding is done to convert every sentence into a similar length.

b) Modelling the data

The Recurrent neural architecture has some key components in its architecture. They are when each of the sequence of input word vectors are given to the model these are one hot encoded and are mapped to the raw trained english vocabulary datasets.. Then the Embedding layers in the RNN convert every word in the sequence to a vector and this context from the previous time stamps is then forwarded to the present word vector. In the decoder network there are fully connected dense layers which are used to decode the input to the right sequence translation. And then the outputs are generated as the one hot encoded vectors which are then mapped to the Hindi, Telugu, Tamil Language datasets.

c) Embeddings and Training Details

Embeddings in Natural language processing will be helpful to find some of the most precise syntactic relationship between the words. Where each word is projected into the dimensional space and finding the nearest neighbours in that space in which the words occupy same regions in the space. As the training of these embeddings require a lot of computational power and data, so we have utilized a pretrained package word2vec GloVe in our model architecture. The embeddings are trained using keras with the words vocabulary of 50000.

d) Decoder and Encoder network

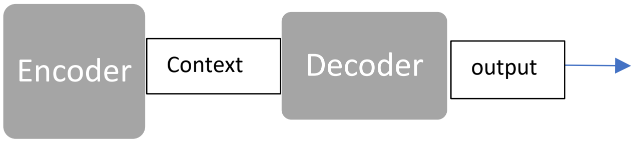


Figure 5 : Encoder decoder network of RNN

The two key components in this sequence to sequence model are encoder and decoder where the encoder job is to summarize the data of the words into a particular context state variable and this is decoded with generating the output sequence.

In this architecture, each of encoder and decoder have the recurrent loops at every part of the time stamps. For example, when we consider five timestamps for encoding the input sequence of words. The encoder performs the read operation at every time stamp and stores it in a hidden state which will then be passed to the next time stamp sequentially. This will be helpful to represent the context of the sequence network

If we consider each time stamp there are two inputs for each sequence of words which are the sequence words from the input and the hidden state. The encoder takes these as inputs to the next state and the decoder takes output from the previous state. Where each word is represented as a vector from embedding layers.

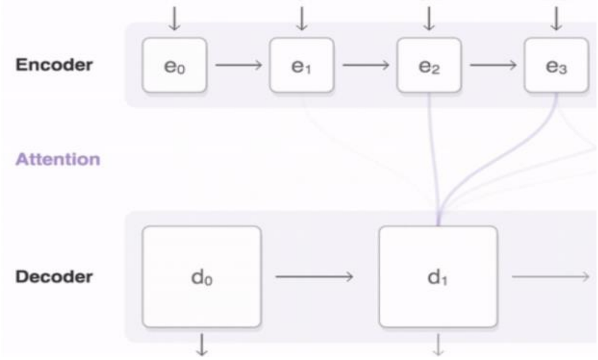


Figure 6 :Implementation of the network with each states and their outputs.

e) Grated Recurrent units with Bidirectional Layers.

To make these RNN more precise decision makers all the data flowing through the previous state to the next state is limited and only the relevant information is allowed to the next state to make the prediction process more precise this is done using the grated recurrent unit architecture where the unwanted information is removed from the previous state.

This is typically done using bidirectional layers while training the network for getting the better performance from the model. These two RNNs are trained simultaneously and the GRU has two gates in its architectural design where one gate decides what relevant information to be transferred to the next step and other gate decides the scope of the information which is to be forgetted. GRU is designed with 128 dense layers, with dropout of 0.5, categorical entropy as its loss function and adam as the optimizer. The model is able to achieve the validation accuracy of 96% when trained with 300 epochs.

Language	**Training units**	**Testing units**
Hindi	73400	18500
Telugu	64532	17800
Tamil	61600	15433
Total	199532	51733

Table 1:Language dataset statistical information

The above table represents the data of words from three different Indian languages and these are trained on Nvidia GTX 1660Ti Gpu. In these languages there are some overlaps between telugu and tamil because they are originated from same Dravidian scripting’s.

Language	**Word error rate %**
Hindi	18.7%
Telugu	25.3%
Tamil	12.8%
Wt. Average	18.93%

Table 2: Word error rate percentage with the weighted average

f) Performance improvements of the network

With the recent advancements in language processing and language modelling techniques some of the embedding techniques like BERT can perform even better. If we can increase the training vocabulary of the languages and inculcating more number of LSTM layers in Recurrent neural network can increase the performance of the model with some architectural changes.

BERT which is a Bidirectional Encoder Representations from transformer which can pretrain the neural network of bidirectional representation from the text by use of condition sequencing on the unlabled data on both right and left context of all the layers.

In the pretraining stage of BERT there are two important tasks performed. Masked language modelling and Next sentence prediction. In the masked models the some of the input tokens are masked at random and they are predicted afterwards. With the input data the bert algorithm masks around 15% of the tokens at random for prediction. In the NSP if two sentences are chosen at random , let’s say P and Q, Half of the time of Q is the actual time of the sentence that follows the first one P. This model has achieved and accuracy of 98% without any fine tuning.

Some of the results of Robust Multilingual representation techniques on low resource languages like Fongbe, Swahili, Wolof could get the WER of 54, 45, 39% respectively when trained on DeepSeeech2 models. When these are trained with the librispeech models the word error rates 73, 67, 65% respectively.

3.3 After finishing natural language modelling stage the next step is audio style transfer where the text output f rom the modelling stage is fed to the Generative adversarial network (GAN)[17] to generate the audio profile using spectrogram.

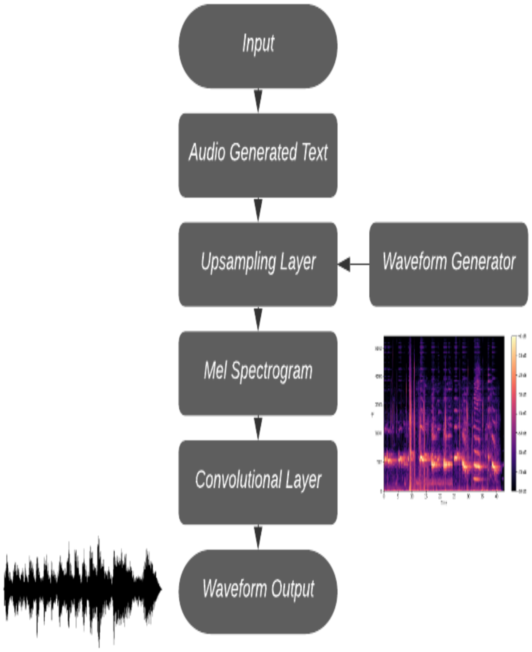


Figure 7: Block diagram of the GAN architecture

Using deep learning techniques, we can generate different images using style transfer algorithms by taking features from one image and learning those patterns and apply it to a different image. In the say way if we want to represent an audio waveform spectrograms are way to represent these audio waveforms in a 2 dimensional structure. When a time domain signal is given and we want to convert it into a frequency domain signal we use fourier transform techniques.

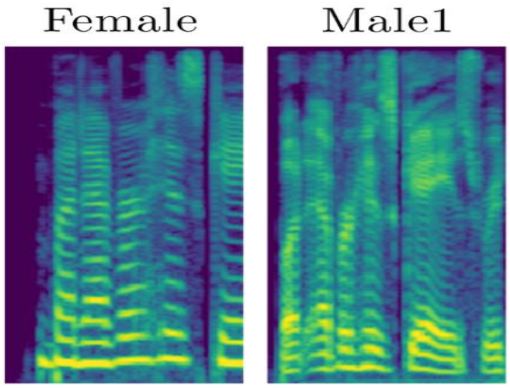


Figure 8:Mel spectrograms with logarithmic amplitude

So to use these spectrograms[18] efficiently, we consider every pixel in image in the decibel scale and take the log value of it and then converting to the mel scale with the application of the mel filter, which ultimately results in mel-spectrograms.

The generation of audio style transfer is inspired from the image style translation using GAN's where the main difference is applying the learnt techniques on the audio data. Our architecture is fully convolutional feed forward network. We have taken the reference from google tacotron 2[15] architecture and modified the convolutional layers for our use case.

This architecture consists of encoder network , decoder network and an attention network. The aim of the encoder is to take the characters as input in the form of sequence and convert these into feature representation's. These feature will be serving as the input to the decoder network to generate a spectrogram[19]. The input text is in the form of character embedding which is given to the convolution layers. These are fully connected layers with 512 hidden ReLU units in it. Once the sequence passes through these convolutional layers the output vector is concatenated and then it is passes through a Bidirectional LSTM layers. These will be helpful to generate and predict the target frame of the spectrogram.

The two LSTM layers are designed with 512 units and the predicted mel spectrogram will again pass through the full connected feed forward convolutional layers to predict some of the residual layers which add to the originally predicted spec and finally both of these spectrogram superimpose and produce an audio waveform with improved reconstruction. The layers after the mels input have 256 units and with 3x3 batch normalization with tanh activation function.

a) Preprocessing text

Out model is designed to take the character sequence as the input. There are different kind of pronunciation in the input sequence such as dates, domain names, numbers etc. As we have taken a limited set a dataset our model was able to pronounce text properly but when the words like homonyms are misspelled sometimes. The normalization process is done beforehand to check how these phonemes are pronounced. And then, we train the model with these phenome sequences.

b) Training phase

In the training phase of the GAN architecture we have used least squares techniques **(Mao et al., 2017) which is formulated as below.**

$$\text{Min } E_{x,z} \left[\sum_{K=1}^{n} -A_k(D(x,z)) \right] \qquad (1)$$

Here x represents the waveform and z represents the conditioning waveform.

We have trained the model with publicly available audio waveforms of two different persons. One male and other female to differentiate the pitch and dialects clearly in the output waveforms. But we have observed that the female profile waveforms are clearer than the male profile. As in the male profile some words with high pitch have the high frequency range. These words overlapped with concurrent words and causing noise when it is converted back to audio waveform.

c) *Conversion of Spectrogram to audio waveform*

The final step is inversion back to audio waveform from these spectrograms in which the frequency domain signals are converted to time domain audio waveforms with trained voice outputs. This results in a generation of deep fake voice outputs.

IV. EXPERIMENT RESULTS

In this section we will discuss the experiment results form the entire model as well as phase wise. The main metric here is Mean Opinion Score (MOS) which is metascore calculated by the average from number of individual single valued components. In recent times the audio and video outputs are not calculated manually by group of professionals rather the algorithms itself produce the results which are close to human evaluation schemes. These are generally given between 1 to 5.

The other metrics calculated are accuracy of the model and word error rate which tells us the efficiency of the model and latency rate.

When the target spectrogram features are computed from the algorithm the following results are obtained. The final transcribed waveform mel-spectrograms are listed below:

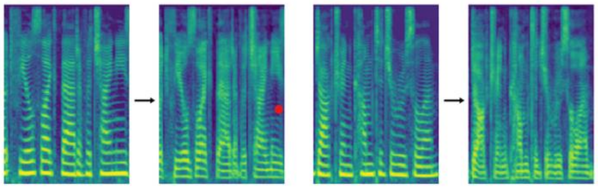


Figure 9 : male voice as the Input and male voice as output

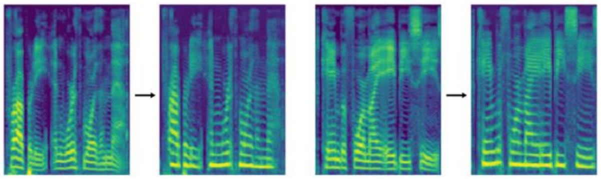


Figure 10 : male voice as the Input and Female voice as output

Table 3 : Summary of some of the english datasets is presented below.

Dataset name	**Hours trained**	**Language**
AV Speech	3200	Multilingual
Librispeech	~ 900	English
Wall street journal	~ 70	English
TIMIT data	~ 5	English

In the embeddings space, the comparison of visualizing the male and female voices with some quantitative results. Below shown are principle component analysis of the synthesized voice outputs and these when compared to the human speech.

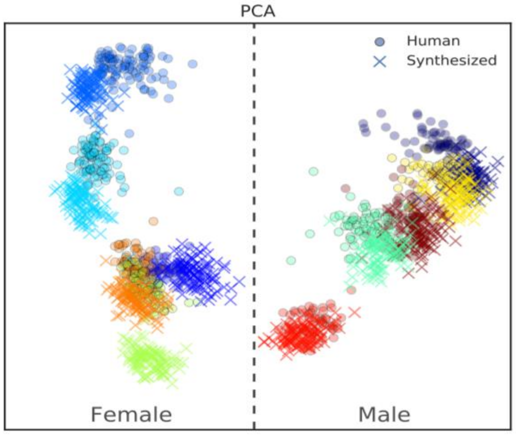


Figure 11: PCA on synthesized voice and human voice of both male and female.

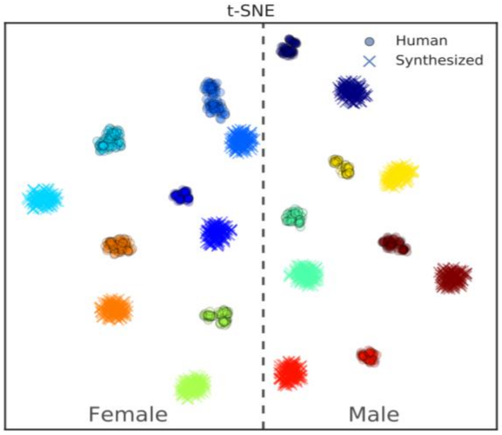


Figure 12: t-SNE on synthesized voice and human voice of both male and female.

These visualizations are generated to distinguish between the real and synthesized voices, where circles represent human generated voice and triangles represent synthesized voice and we can see that upon testing on both of these algorithms they are 92% accurate on female voice and 89.45% accurate on the male voices.

	Mean opinion Score on scale of 1-5 in naturalness			
Speech model	**English**	**Hindi**	**Telugu**	**Tamil**
Tacotron 2	4.30 ± 0.04	4.26 ± 0.16	3.87 ± 0.23	3.47 ± 0.47
wavenet	4.26 ± 0.081	4.08 ± 0.24	3.82 ± 0.12	3.15 ± 0.64
LSTM-RNN (our model)	3.23 ± 0.02	2.81 ± 0.54	2.56 ± 0.37	2.47 ± 0.07

Table 4 : Mean opinion scores of our model in comparison with state of the art algorithms.

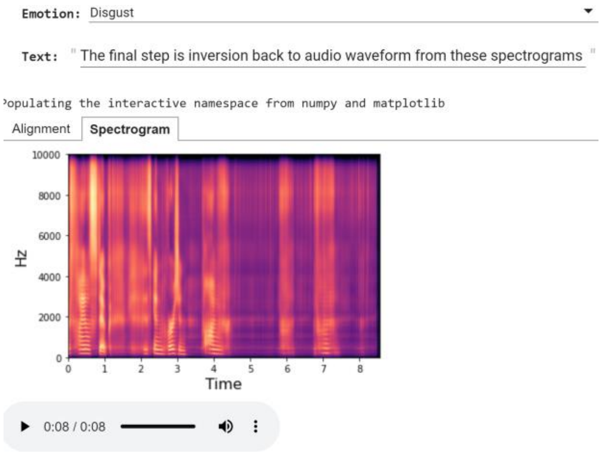
From the results obtained we can say that the MOS of english language is high in our model when compared to other languages. The maximum MOS score obtained is 3.25% and the minimum score obtained is 2.54% for tamil language.

The designed model during the training phase sometimes it under-fit the loss rate and there by increasing the word error rate. But this happens when the size of the dataset is limited. When a large dataset is considered, due to high volumes of vocabulary in language specific sets there may be chance of decrease in the word error rate.

Model	WER
Word piece models	9.2
Grapheme	12.0
Our model	18.93
LAS model	8.1

Table 5 : Word Error rate comparison with state of art models

We have also performed the pitch synthesis on the female voice . We have considered 4 different emotional pitches. They are Neutral, Angry, Sleepiness, and disgust. We have generated the spectrograms and audio waveforms of the voice of a scientist where the audio datasets are publicly available to train. Each waveform took approximately 40sec to generate based on the text input given to the model. These are trained on Nvidia GTX 1660Ti GPU and with the training time of 2 hrs 40mns .

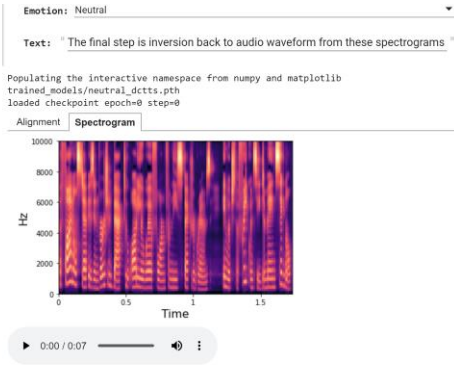


Emotion: Disgust

Text: "The final step is inversion back to audio waveform from these spectrograms"

```
Populating the interactive namespace from numpy and matplotlib
```

Figure 13: Disgust emotion and audio waveform and Spectrogram generation.

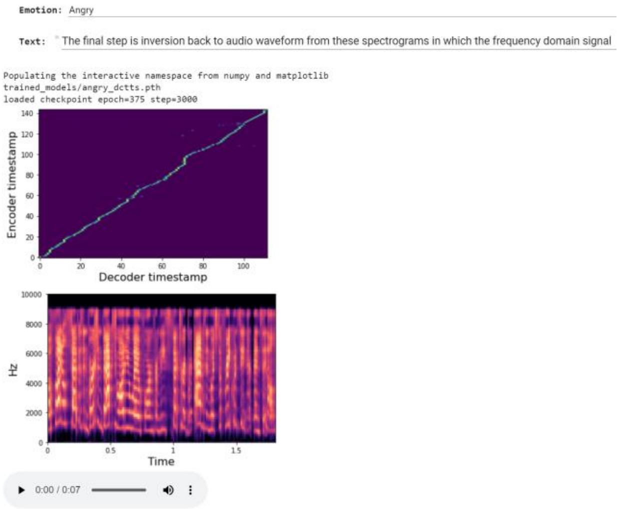


Emotion: Neutral

Text: "The final step is inversion back to audio waveform from these spectrograms"

```
Populating the interactive namespace from numpy and matplotlib
trained_models/neutral_dctts.pth
loaded checkpoint epoch=0 step=0
```

Figure 14: Neutral emotion and audio waveform and Spectrogram generation.

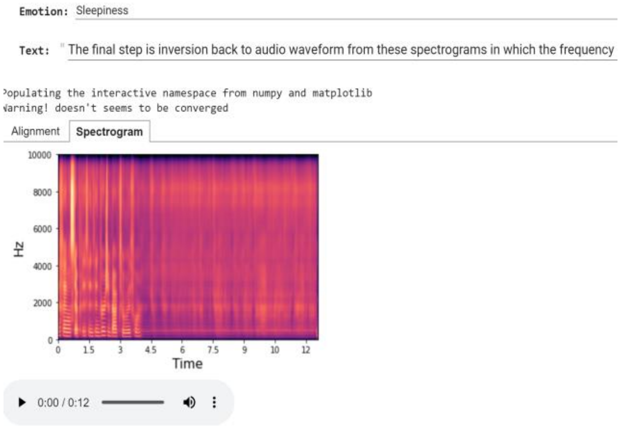


Emotion: Angry

Text: "The final step is inversion back to audio waveform from these spectrograms in which the frequency domain signal

```
Populating the interactive namespace from numpy and matplotlib
trained_models/angry_dctts.pth
loaded checkpoint epoch=375 step=3000
```

Figure 15: Angry emotion and audio waveform and Spectrogram generation



Emotion: Sleepiness

Text: "The final step is inversion back to audio waveform from these spectrograms in which the frequency

```
Populating the interactive namespace from numpy and matplotlib
Warning! doesn't seems to be converged
```

Figure 16: Sleepiness emotion and audio waveform and Spectrogram generation

From the spectrograms data, we see that while the pitch is sleepiness the results not good and there is a concurrent overlap between the words.

V. CONCLUSION AND FUTURE WORK

In this paper we have presented a deep learning system, that takes the podcasts audio as an input and can be converts to a multilingual model with the users favourite voice as output. We observed that our model overall accuracy in reconstruction of the original audio profile with different voice is approximately 89.45% . There are 3 key stages involved during this operation. First the conversion of speech/ audio waveform into text with the least possible error rate. Where we also designed the time offsets for each sampled word to get the exact time stamp of the spoken word. This process helps us to minimize the search time as well as reduction in concurrent overlap of the words. Second we have modelled the text output received from the first stage and translated to 3 different languages with the consideration phoneme boundaries because we needed to generate the same audio profile at the later stages of the model. Here we have achieved a word error rate of 18.93% which is slightly greater than the state of the art models because of the computational complexity and the data used for training the model is very limited when compared to classical models. In the third stage, the specific language model generated is fed to the GAN network where the voice model conversion takes place. We have trained our model with publicly available voice data of a person to check the efficiency of reconstruction. To model this we have taken reference from google tacotron 2 model architecture and modified the bidirectional it accordingly to our use cases. We are able to generate a MOS score of 3.25% as maximum for english and 2.54% as minimum for tamil language. We have also performed pitch synthesis using 4 different emotions such as angry, disgust, neutral and sleepiness. We have generated the mel spectrograms for each case. This whole pipeline of deeplearning system can make a greater impact in the society when used in right context. Notably our approach does not address low resource languages and is limited to 3 languages. We believe that this data driven modelling can prevail a large set of opportunities and use cases in the future.

REFERENCES

[1]. Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Saeid Nahavandi. (Jul 2020) Deep Learning for Deepfakes Creation and Detection: A Survey

[2]. Yang, W., Hui, C., Chen, Z., Xue, J. H., and Liao, Q. (2019). FV-GAN:Finger vein representation using generative adversarial networks. IEEE Transactions on Information Forensics and Security, 14(9), 2512-2524.

[3]. Guo, Y., Jiao, L., Wang, S., Wang, S., and Liu, F. (2018). Fuzzy sparse autoencoder framework for single image per person face recognition. IEEE Transactions on Cybernetics, 48(8), 2402-2415.

[4]. Cao, J., Hu, Y., Yu, B., He, R., and Sun, Z. (2019). 3D aided duet GANs for multi-view face image synthesis. IEEE Transactions on Information Forensics and Security, 14(8), 2028-2042

[5]. de Lima, O., Franklin, S., Basu, S., Karwoski, B., and George, A. (2020). Deepfake detection using spatiotemporal convolutional networks. arXiv preprint arXiv:2006.14749.

[6]. Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen. State-of-the-art speech recognition with sequence-to-sequence models arXiv preprint arXiv:1712.01769v6

[7]. Miss.Prachi Khilari, Prof. Bhope V. P. A review on speech to text conversion methods International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 7, July 2015

[8]. Sanjib Das, “Speech Recognition Technique: A Review”,International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 3, May-Jun 2012.

[9]. Kazuya Kawakami, Luyu WangChris, Dyer Phil Blunsom, Aaron van den Oord DeepMind, London, UK. Learning Robust and Multilingual Speech Representations arXiv preprint arXiv:2001.11128V1

[10]. Fréjus A. A Laleye, Laurent Besacier, Eugène C. Ezin, and Cina Motamed. 2016. First automatic fongbe continuous speech recognition system: Development of acoustic models and language models. In Proc. FedCSIS.

[11]. Aaron van den Oord Sander Dieleman Heiga Zen ¨ † Karen Simonyan Oriol Vinyals Alex Graves Nal Kalchbrenner Andrew Senior Koray Kavukcuoglu Wavenet: a generative model for raw audio, arXiv preprint arXiv:1609.03499v2.

[12]. Sercan Ö. Arık, Gregory Diamos, Jonathan Raiman, Andrew Gibiansky, Yanqi Zhou, Wei Ping, Deep Voice 2: Multi-Speaker Neural Text-to-Speech. arXiv preprint arXiv:1705.08947V2

[13]. Shubham Toshniwal Tara N. Sainath, Ron J. Weiss, Bo Li, Pero Moreno, Eugene Weinstein, Kanishka Rao. Multilingual speech recognition with a single end-to-end model arXiv preprint arXiv:1711.0164v2

[14]. Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Alexandre de Brebisson, Yoshua Bengio . Aaron Courville . MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. arXiv preprint arXiv:19110.06711v3.

[15]. Jonathan Shen , Ruoming Pang , Ron J. Weiss , Mike Schuster , Navdeep Jaitly , Zongheng Yang , Zhifeng Chen1 , Yu Zhan , Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous , Yannis Agiomyrgiannakis. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. arXiv preprint arXiv:1712.05884v2.

[16]. Daniel S. Park∗ , William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. arXiv preprint arXiv:1904.08779v3.

[17]. Jeff Donahue , Sander Dieleman , Mikołaj Binkowski, Erich Elsen, Karen Simonyan . End-to-End Adversarial Text-to-Speech. . arXiv preprint arXiv:2006.03657v1.

[18]. Ye Jia∗ Yu Zhang∗ Ron J. Weiss∗ Quan Wang Jonathan Shen Fei Ren Zhifeng Chen Patrick Nguyen Ruoming Pang Ignacio Lopez Moreno Yonghui Wu. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. arXiv preprint arXiv:1806.045584v4.

[19]. Naihan Li, Shujie Liu, Yanqing Liu , Sheng Zhao , Ming Liu, Ming Zhou. Neural Speech Synthesis with Transformer Network. . arXiv preprint arXiv:1809.08895v43