Affective Social Anthropomorphic Intelligent System

By

Md. Adyelullahil Mamun
ID: 20241044 (17101278)
Hasnat Md. Abdullah
ID: 20241047(17101297)

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

	Hasnat
Md. Adyelullahil Mamun	**Hasnat Md. Abdullah**
ID: 20241044 (17101278)	ID:20241047(17101297)

Approval

The thesis/project titled “Affective Social Anthropomorphic Intelligent System” submitted by

1. Md. Adyelullahil Mamun - 20241044 (17101278)
2. Hasnat Md. Abdullah – 20241047 (17101297)

of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 11-01-2021.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor, Dept. of Computer Science and Engineering
BRAC University

Thesis
~~Program~~ Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor, Dept. of Computer Science and Engineering
BRAC University

Departmental Head:
(Chair)

Sadia Hamid Kazi
Assistant Professor & Deputy Head
BRAC University

Abstract

At present, intelligent virtual assistants (IVA) are not only about delivering the functionalities and increasing their performances; they also need a socially interactive personality. As human conversational styles are measured by our sense of humor, personalities, tone of voice, these qualities have become essential for conversational intelligent virtual assistants. Our proposed system is an anthropomorphic intelligent system that can hold a proper human-like conversation with emotion and personality. It can also be able to imitate any person's voice given; voice audio data is available. Initially, the temporal audio wave data will be converted to frequency domain data (Mel-Spectrogram), which contains distinct patterns for audio features like the notes, pitch, rhythm, and melody. A parallel CNN, Transformer-Encoder, is used to predict the emotion from 7 different audio data classes. This audio is also fed to the deep-speech, an RNN model that consists of 5 hidden layers. From the spectrogram, it generates the text transcription. Then the transcript text is transferred to the multi-domain conversation agent, using blended skill talk and transformer-based retrieve-and-generate generation strategy and beam-search decoding an appropriate textual response is generated, which in turn gets synthesized to audio using WaveGlow that is based on WaveNet and Glow. It learns an invertible mapping of data to a latent space that can be manipulated and generates a Mel-spectrogram frame based on previous Mel-spectrogram frames. Finally, from the generated spectrogram, the waveform is generated using WaveGlow. A fine-tuned system can be used in the following but not limited to applications like dubbing, voice assistant, re-creating new movies with old actors.

Keywords: IVA; NLP; SER; Emotion; Audio-Emotion; Personal-Assistant

Acknowledgment

We are grateful to our supervisor, Dr. Md. Golam Rabiul Alam for guiding us throughout the process of conducting this thesis. Without his direct supervision, it would not have been possible to carry on such research and submit a comprehensive thesis.

We want to express our gratitude to the unmet peoples on the internet for their selfless help and guidance throughout the research period. It would not have been easy to accomplish such a task without their help

We are also grateful to our family and friends who have contributed a lot in accomplishing this task.

Lastly, We would like to thank all our institute's teachers who have helped us gain valuable knowledge and insights throughout the four years of our under-graduation life.

Table of Content

List of Tables

List of Figures

List of Acronyms

IVA	Intelligent Virtual Assistant
SER	Speech Emotion Recognition
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network

Chapter 1

Introduction

1.1 Introduction

Lately, we are witnessing a race between several researchers and tech companies worldwide trying to improve and innovate in the field of artificially intelligent anthropomorphic personal assistants. They are more commonly referred to as Intelligent Virtual Assistant, IVA for short. From IBM Shoebox to Google Assistant, IVA technology has come a long way. Despite the best effort, IVA still falls short in many areas that are innate to human interaction, especially emotions and empathy. Strives are being made to clear the boundary between humans and IVA.

1.2 Problem Statement

Although IVA has been out to the public for some time now, it is still very premature if compared with a human. Most contrasting characteristics with humans are: not getting the full context of the speech, lack of emotional recognition of the speaker, and no presence of personality. Current IVA can hold short conversations like asking, "what is the weather?", "do I have meetings?", "send an email", etc. Unfortunately, it is not yet possible to hold a meaningful, deep conversation. We will dissect three major limitations of current generation IVA: context, emotion, and personality.

Firstly, current IVA technology disregards context, voice tone, etc. It cannot understand if a question is serious, sarcasm, or a joke. An infamous incident[1] took place in 2012, a man 1 killed his friend and allegedly asked "SIRI" (a commercial IVA developed by Apple) where to hide the body. SIRI answered, not getting the full context, and surely enough, the murderer followed the instruction. However, after the incident, Apple, Siri's maker, disabled this conversation, and now Siri only responds with an apology. To add to that, these sophisticated virtual agents do not know anything about languages. When we say "Alexa (a commercial IVA developed by Amazon), what is the weather today," her voice-recognition model converts speech to text based on models of digitized sounds. The reason why we get desired results is that the sentence "what is the weather today" has words like "weather" and "what," which are classified as known intent for weather report searching. If we ask, "Alexa, I do not want to know about the weather today," the response will not be an expected one.

Secondly, another area where IVA is severely lacking is emotion. As humans are driven by emotion, it creates a huge communication gap when communicating with IVA. This becomes more serious when it turns into a mental health issue. For some time now, all public-facing

[1] https://nypost.com/2014/08/13/accused-killer-asked-siri-where-to-hide-roommates-body

commercial IVA has been hard-coded to detect some trigger word. For example, if a user says, "I want to kill myself," to IVA, it will show suicide helpline and prompt them to get help. But, if we rephrase the sentence in a different way, like "I don't want to wake up tomorrow," IVA will not show help; it will try to cancel tomorrow's schedule or do a simple search. This happens because the tone of voice is a big indicator of emotion. Without an emotional context, that sentence can mean many things, from killing oneself to sleeping on the weekend morning. An emotionally aware IVA can help a lot of people who are suffering from mental health issues.

Lastly, current generation IVA does not have a personality. It is easier for humans to relate and connect with others if they have similar personalities[1]. All of the current IVA has two significant challenges to meet: How to transmit both effective and personalized qualities in the form of a consistent and realistic speech when embedded in a computational framework [2]. The IVA needs to adapt its speech to different interaction types that users might use [3]. The benefit of having a personality in an IVA is that it can continue a proper conversation with human beings. The character of IVAs must be natural and believable and must reflect moods, personality, and expressions [4]. That is why an added personality will make a significant change in how we interact with the IVAs now, which will give us the feeling of having a companion.

The issues mentioned above are very complex, and no one simple solution exists to fix those problems. Researchers are trying to come up with solutions that address those lacking.

The issues mentioned above are very complex, and no one simple solution exists to fix those problems. Researchers are trying to come up with solutions that address those lacking.

1.3 Research Objectives

There are many different ways IVAs from the current generation can be improved. Nevertheless, our research's objective is more specific. To develop a novel, emotion aware IVA that can impersonate any person's voice traits given, we feed the person's sufficient audio voice. The primary objectives of our research will be:

- Processing raw audio and extract features that are co-related to emotion.
- With the extracted audio features, classifying the proper emotions.
- Generate proper emotional conversation responses.
- Generating proper emotional tones/cues in the generated voice.
- Mimicking voice with limited audio data.

Chapter 2

Literature Review

Speech Emotion Recognition is not a new research interest. We can find a conference paper [5] dated back to 1996, which tries to incorporate emotion with speech using statistical pattern recognition. Even though research interest in software emotion recognition has not changed for decays, its methodology certainly did.

The traditional approach of recognizing emotion from speech comprises Modeling, Annotation, Audio Features, and Textual Features [6]–[9]. In classical modeling, two methods are commonly used. One is called discrete classes, and the other one is the two axes arousal, both suggested by H. Gune et al. [10]. Well, according to L. Devillers et al., after a model, the most important thing is to gather the proper emotional audio dataset that is not only properly labelled, but also suited to a model that focuses on the representation of emotions. [11]. In automatic emotion recognition, the Observer Rating may be an appropriate label that focuses on what emotion the speaker gave to the dataset rather than what emotion the speaker felt as B. W. Schuller et al. has been mentioned in his paper [12]. To avoid the rating annotations, acting or targeted elicitation was used in many previous works. While having the data labeled, one needs to extract the features of the audio data as well as the kind of emotional words in the audio transcript before experimenting with it in different algorithms.

Because of the traditional approach's limitation, scientists and researchers are transitioning towards more modern techniques like Deep Learning. However, some of the inherent limitations of traditional SER are still present in the new methodology. Before we start, we have to discuss those challenges of SER and how researchers tried to overcome those issues. There are mainly three challenges we will face.

First of all, for robust emotional recognition, a holistic speaker model is needed, taking into account vocal attributes such as tone, pitch, volume, speech speed, voice condition. For example, the acoustic environment, tiredness, intoxication, hoarse voice due to having a cold, and many other factors may significantly impact the speaker's tone. Mismatch of those states and traits are shown to degrade the performance[13][14] of the models. Several DNN approaches have been proposed to address this issue. Glorot et al., in their paper [14], have demonstrated that Stacked Denoising Auto-Encoders can extract audio features without any labeled data or human supervision. This is made possible with the help of a small rectifier unit. Moreover, their approach has improved generalization over the baseline significantly.

Similarly, Deng et al., in their paper[15], address the situation where training and test corpora are from different datasets. They proposed a "shared hidden-layer autoencoder" approach to reduce the discrepancy and extract common features caused by different states, e.g., speakers, environment, language. Experiment results showed it had an imposing result compared with other domain adaptation models.

Second, efficient data collection is the key to an emotional recognition system. There have been fewer audio speech data available since the beginning, which are properly labeled with the emotional states that the audio speech expresses. Efforts have been made in the recent past to gather or generate and properly label audio speech data [7][13][14]. To combat this data scarcity, weakly supervised and semi-supervised approaches are introduced. Semi-supervised learning approaches could successfully label the rest of the dataset after training a model initially [16], [17]. To keep the data quality up, we just cannot rely on machine-generated labels. It is better to keep humans somewhat involved in the labeling process. This hand-to-hand cooperation of human labeling with a machine (Semi-supervised learning) is called Active Learning.

On the other hand, anyone can generate linguistically and emotionally matched speech audio data and compare it to the original reference audio speech in order to filter out less similar ones [12]. Generating audio speech and comparing it with the original reference audio are also done using generative adversarial networks. It's done with the help of two neural networks. The first neural network is trying to generate audio speech. At the same time, the second neural network attempts to calculate the difference and to identify which sample is original and which sample is generated [18]. Moreover, dissimilarity between generated and original speeches can be mitigated by transfer learning. Transfer learning has been successful to fully transfer sentiment from text to image[19]. We believe the same method can be used in SER. One of the recent developments in the Transfer learning field was bought by Google on their paper. The proposed system contains a speaker encoder network, Tacotron 2 based seq-to-seq synthesis network, and an autoregressive Wavenet based vocoder network. The authors modified their Tacotron's text encoder by removing the batch normalization. Instead, they went with instance normalization. They used neural network with decoder functionality which discarded Tacotrons key features which includes two layers called Prenet and Postnet. Attention mechanism based on tanh was mentioned the Singing Voice Synthesis Paper[20] was used. According to the authors, they used a speech synthesis model called Flowtron which generated speech audio samples. The generated samples were quite similar with the reference audios which is proved by the mean opinion score. The value was quite familiar with other state of the art speech synthesis models. Scenarios where no annotated data, no emotional speech synthesizer found, Deep Belief Network (unsupervised learning) can be used[21, p. 20] as it does not require explicit knowledge of emotion. If there is no speech data available, a rule-based approach that exploits the knowledge existing in the literature is used.

Finally, the last issue is the Naturalness of Generated Emotion. Due to vagueness and intrinsic complexity, generating emotion remains an ongoing challenge. However, not a lot has been done in this direction. There are a few papers that try to address the challenge. Lee et al., in their paper[22], introduce a speech synthesizer based on end-to-end mode with context vector and residual connection at recurrent neural networks that can generate emotion given emotion labels. Another notable work[23] is done by Akuzawa et al., where they use VoiceLoop, an autoregressive SS model, with Variational Autoencoder (VAE) and overcome the lack of global characteristics of speech limitation. With this improved method, higher quality speech can be generated compared to VoiceLoop without label and control speech expression. Despite all the effort, voice generation is far from natural due to the inherent complexity and non-linear nature of emotion. Generative adversarial networks have shown promise[24], [25] regarding generating samples more understandable to humans.

However, we would like to emphasize the importance of the Generative Adversarial Network (GAN) in this study. Since the first inception of GAN[26] in 2014 by Goodfellow et al., many GAN variations have been proposed by different researchers and have been successful in many real-life scenarios. From a publicly curated list[2], there are more than 500 variants of GAN already available. In Image manipulation, GAN has achieved unprecedented success. Not as many, but some attempt has been made to incorporate GAN with audio synthesis and generation. Pascual et al., in their paper[27], propose a generative adversarial framework with an end-to-end speech enhancement method, which is an effective alternative to the current approach that works on spectral-domain and exploits some higher-level features. Due to the model's encoder-decoder fully-convolutional structure, it can operate fast on denoising waveform chunks. WaveGAN[28] is a raw audio synthesizer in an unsupervised setting. It can generate text transcription from speech audios as well as generate audios from other sources like musical instrument, animals [28]. Another model named VoiceGAN[29], a novel neural

[2] https://github.com/hindupuravinash/the-gan-zoo

network model for speeches which can generate human likely vocal audios. It trains to imitate the target speakers' vocal attributes rather than focusing on what the speaker is saying and generate mel-spectrograms. Then, the spectrograms are converted using the Griffin-Lim method into time domain for speech audios. Oord et al., proposes WaveNet[30], which converts mel-spectrograms in to time domain audio waves that we hear as speeches. In the paper of Parallel WaveGAN[31], they proposed a method of generating audio waves with the mechanism of generative adversarial network. In this method, WaveNet is trained in which current mel-spectrograms do not depend on its previous time slice. Also, adversarial losses are minimized along with the training. In the field of text to speech model, one of google's papers proposed "Tacotron," which synthesizes speech directly from characters[32]. This model is an end-to-end generative text-to-speech model.

Moreover, in another paper[31], researchers from google proposes a system that uses recurrent neural network that predicts the next feature according the previous features and context as well generates Mel-spectrograms with respect to the character embeddings. These Mel-spectrograms are converted to a time-domain waveform via vocoder, a modified wavenet[33]. Combination of these two mechanisms generates good quality audio speeches from text transcriptions,which is also known as "Tacotron 2". Lastly, MelGAN[28] overcame the previous limitation of models by introducing architectural changes and simplifying training techniques. This model does not predict Mel-spectrogram based on previous Mel-spectrograms and also it follows convolutional approach which leads to fewer parameters than other fully connected neural networks like the competing models. One of the most advantages of this model is the fast-training speed compared to competing models.

The next thing that comes up is learning features from the labeled data. All the basic features like happiness, sadness, anger, fear, and many more should be optimally fitted to the labeled data. After that, the unquantifiable features like acoustic environment, tiredness, etc., should be learned. There are two popular ways of learning feature representation. The first one is "speech to words," which focuses on phonetic level features of the audio and maps them to the text transcriptions [34]. The other way is "audio-word re-tagging" of hierarchical clustering—for instance, parts of speech tagging in textual word handling. In recent past, we have seen a new model architecture that has been able to generate great performance in the field of speech emotion recognition by utilizing convolutional neural network approach with long short term recurrent neural network added at the last layer [7].

On the other hand, proposes Glow [35], which uses an invertible 1 x 1 convolution to generate a simple type of generative flow. By the use of this convolution, they have shown improvement in "log-likelihood" of any data on standard benchmarks, which will further be used in speech-generating models like WaveGlow[36], which is intended to generate high-quality speeches from Mel-spectrograms. WaveGlow provides high-quality audio waves by merging model architectures both from WaveNet [30] and Glow [35].

In early 2020, google presented an open-domain chatbot Meena [37], which is an enormous 2.6B parameter neural network model. The researchers have found a way to show human-level rationality by boosting the probability of the next token on huge conversational data mined from social media. In the Facebook AI's recently published paper [38], they have presented different versions of conversational agents with big amounts of parameters. These models blend the attributes that a natural human-like conversation might have liked having a personality, giving proper time to the writer, having knowledge as well as empathy, raising different points to continue a healthy conversation etc.

The confidence measures are an integral part of emotion recognition engines. Learning emotion, along with human agreement side by side, can be an effective way of measuring confidence. It means that the model does not classify emotions directly. Rather, the model will classify how many people agreed on which emotion the audio was, and will eventually give us the standard deviations of the prediction from the reference labels. [12]. Another way could be training a compression auto-encoder which takes all the features into consideration and tries to differ between what was given into the model and what label was generated. The variations in the difference represents the confidence in speech emotion recognition results. Estimating acoustic degradation or word error rate is another means of confidence measure.

The reliability of the SER system is known by looking through the papers where researches mentioned about the obstacles, they faced in the related domain lately. However, lack of properly labeled data and various perceptions about the same labeled data are creating barriers to get desired accuracy in speech emotion recognition domain.

Chapter 3

Dataset Description

We used multiple datasets to get well-rounded data for our models. Some of the data were created professionally, and some were crowdsourced. We were careful not to introduce any bias in our model, so we tried to balance every data with an equivalent counterpart. For example, SAVEE had only male actors, so to compensate, we have added a TESS dataset that contains only female actors. To ensure we get an accurate representation of the real world, we have added the CREMA-D dataset, which is very diverse and contains audio with different accents and quality. We have added a summary of the datasets we have used below.

3.1 RAVDESS

RAVDESS[39] is the short form of "Ryerson Audio-Visual Database of Emotional Speech and Song." Although the full dataset contains speech and song, audio, and video, we will be only using the speech audio-only files (16bit, 48kHz .wav) for our purpose. This speech audio dataset consists of 1440 files (60 Trial x 24 Actors), which were done by professional actors (12 female and 12 male) vocalizing two lexically similar statements in a North American accent. The speech dataset includes neutral, calm, happy, sad, angry, fearful, surprise, and disgust expressions with two levels (normal, strong) of emotional intensity.

3.2 SAVEE

The full meaning of SAVEE[40] is Surrey Audio-Visual Expressed Emotion. This dataset has high-quality audio of only male voices. There are four native English male speakers who are from the University of Surrey. The use of this male-only dataset will create biases in the models that will be trained. That is why it is advised to use this dataset with other datasets with more females (in our case, we used TESS) speakers. There are seven emotional categories of data in this dataset: anger, disgust, fear, happiness, sadness, surprise, and neutral. The age of the male voices was from 27 to 31 years. The text material consists of 15 TIMIT sentences for each emotion. For one emotion, there are three common, two emotion-specific, and ten generic sentences that were different for each emotion and phonetically balanced. The three common and 12 emotion-specific sentences were recorded as neural to give 30 neutral sentences. In Total, there are 120 utterances per speaker.

3.3 TESS

TESS[41] is the short form of Toronto Emotional Speech Set. This dataset consists of the voices of two actresses aged 26 and 64. As a whole, there are a set of 200 target words that were spoken in this dataset. The audio recordings resemble each of the seven emotions: anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral. There are 2800 audio files of the wav format in which the two actresses uttered 200 target words with respective emotions. It is recommended to use this dataset with male-only datasets to avoid biases in the generated model.

3.4 CREMA-D

CREMA-D[42] stands for "Crowd-sourced Emotional Multimodal Actors Dataset." This is a dataset of 7442 audio clips from 48 male and 43 female actors between the age of 20 and 74. The actors come from different races and ethnicities like African American, Asian, Caucasian, Hispanic, and Unspecified. This is the most diverse dataset of all the datasets we have included in this paper. The actors are assigned to speak from a selection of 12 sentences using one of six emotion categories. The emotion categories were Anger, Disgust, Fear, Happy, Neutral, and Sad. They had four levels of intensity (Low, Medium, High, and Unspecified). The participants rated the emotion and emotion level judging from audio-only, video only, and audiovisual presentation. The process was crowdsourced, and a total of 2443 participants each rated 90 unique clips consisting of 30 audio, 30 visual, and 30 audio-visuals. 95% of the clips have more than seven ratings.

3.5 Indic TTS

Indic TTS [43] is a project that uses a consortium of a high-quality corpus for building text to speech synthesis systems for 13 major Indian languages [44], which includes Bengali too. The dataset includes audio speeches along with text transcriptions. For each primary language, there is a male and a female speaker who utter the lines from various domains such as newspapers, fiction, science, etc. Moreover, audio speeches are recorded in a quiet and echo-less environment. The sampling rate of the recorded audio signals is 48KHz. The recorded audio data uttering English sentences in Bengali accent and Hindi accent of both male and female is 15.23 hours and 15.75 hours. This speech corpus [44] is intended to create various speech synthesis systems in the Indian language and English, where the systems will work better for the Indian accent.

3.6 LJ speech

LJ speech [45] is a dataset created by Keith Ito and Linda Johnson. This is an entirely public domain dataset. One speaker who is Linda Johnson herself uttered 13100 short audio clips from 7 non-fiction books. The total length of audio clips is almost 24 hours. English transcription is created for each audio clip. Also, the audio clips are not fixed in length, varying from 1 second to 10 seconds. The dataset authors manually matched text transcription to the audio, and a QA was passed to prove that the transcripted words correctly matched with the audio speeches.

3.7 Libri TTS

LibriTTS [46] is an extensive dataset totaling 585 hours of audio speeches of the English language. This multi-speaker English Corpus is created for building Text-to-Speech models and further research in this field. The audio signals sampling rate is 24kHz for this dataset. This dataset is generated from another corpus called LibriSpeech [47], changing the original dataset's different characteristics. The changes include changing the sampling rate to 24kHz, adding contextual information, excluding background noises, and including original and normalized texts in the dataset.

Chapter 4

Features & Augmentation

4.1 Features

Different data cleaning procedures like noise removal, making the audios of equal lengths, and equally padded with silence at the beginning and end of the audio clips have been done with our datasets. We need the correct data and a good representation of our data for classification and predictive models, which we call features. We have identified multiple features of our audio data that we used to feed different models to experiment and get better results.

4.1.1 Short-Time Fourier Transform (STFT)

Short-Time Fourier Transform (STFT) is the baseline of all the features that we are going to discuss. STFT divides the audio waves into different equal segments, which are short and overlapping. After that, the Fourier transform of each segment is used to generate power spectrograms. The goal of making power spectrograms is to identify resonant frequencies in the waveforms. The advantage we get from doing an STFT is that it identifies the changes in the audio signals in time series data.

4.1.2 Mel-Spectrogram

The Mel-spectrogram is a Mel-scale representation of frequencies created by fast Fourier transformation. The audio wave signals are converted from the time domain to the frequency domain by short-time Fourier transform using short and overlapping segments over the audio signals, and this is called the spectrogram. The spectrogram's frequency axis is then converted into a log-scale as we humans have a minimal range of recognition of frequencies and amplitudes. Also, the color dimension is converted to decibels. Finally, the frequency axis is mapped on the non-linear Mel Scale to generate a Mel-spectrogram. Mel-spectrograms are simplified analog representations of the power spectrograms in the Mel-frequency scale. This is another feature that can be used in different classification models.

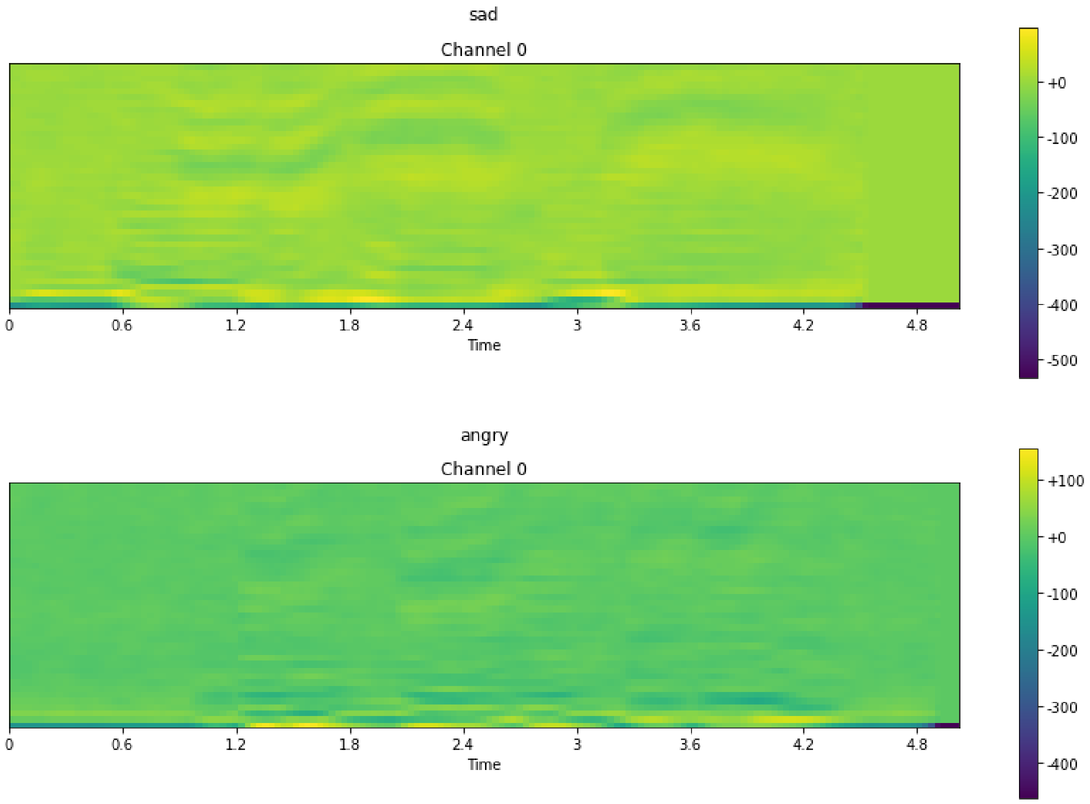


Figure 1: Mel-Spectrogram (Sad, Angry)

4.1.3 Mel-Frequency Cepstral Coefficients (MFCC)

Mel-frequency cepstral coefficients identify the changes in the pitch of audio signals. It is a mathematical function to transform power spectrograms of an audio signal generated by STFT into a small number of coefficients, representing the power of that audio signal in the frequency domain. There are some mathematical procedures that are done one after another for this transformation. First, STFT is used to generate audio power spectrums. Then, frequency bins are generated by applying triangular, overlapping window functions to the power spectrograms and taking the sum of each window's energy. After that, the frequencies of the audio signal's power spectrograms are mapped in the Mel Scale.

This mapping helps to finalize the number and position of window functions and the width of the frequency bins. The reason for using Mel Scale is that humans hear the audio pitches based on frequency ratios, and it is a non-linear pitch scale that represents the audio pitches in "mels" of audio in terms of its frequency. Window functions and frequency bins altogether are called mel filterbanks. Then, the log of the sum of audio signals power spectrogram, also known as cepstrum, is taken for each filterbank. Finally, for each filterbank, discrete cosine transform (DCT) is applied to the log of the sum of the power spectrograms to decorrelate them since there are correlations between filterbank energies. The benefit of using a discrete cosine transform is that it generates coefficients so that the audio signal is fairly represented by only the top few coefficients. So, the amplitudes of the discrete cosine transform of the log of the sum of the filterbank powers with respect to time are mel-frequency cepstral coefficients. MFCC paves us the way to deconvolutionize audio signals to identify resonant frequencies.

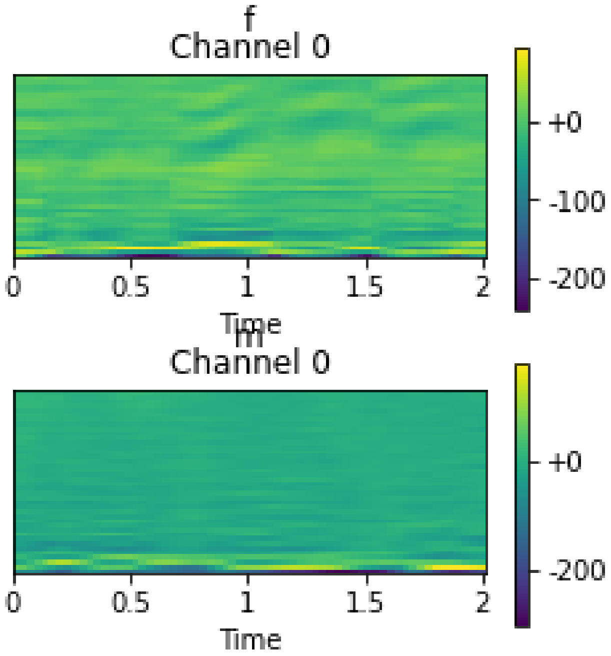


Figure 2: MFCC (Excited, Neutral)

4.1.4 Delta

Delta is the derivative of coefficients. In other words, Delta gives us an overview of the changes in coefficients. It helps us to identify the audio speeches better. With respect to time, the Delta of MFCC will represent a better understanding of the dynamics of power spectrums of audio signals. We will be using MFCC with the Delta of MFC coefficients combinedly as a feature for our models.

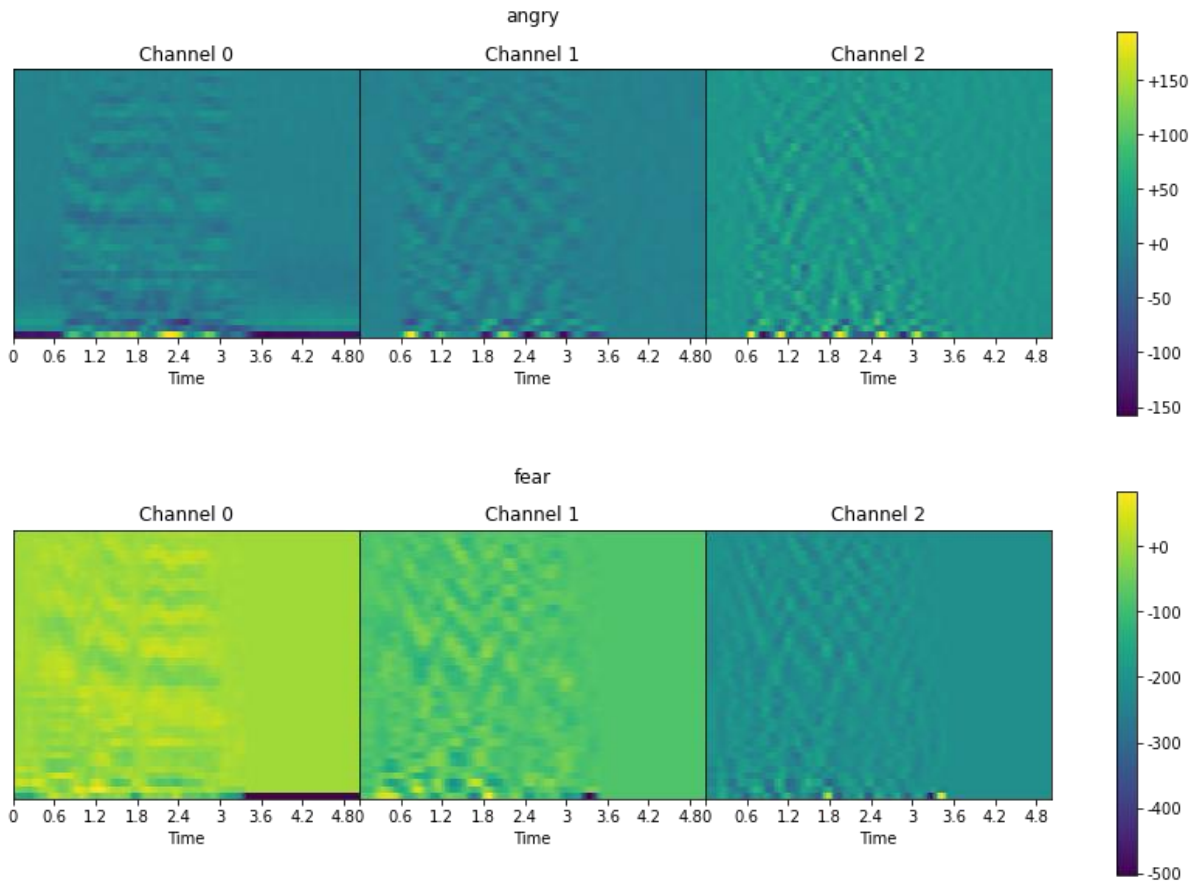


Figure 3: Delta Features (Angry, Fear)

4.2 Augmentation

4.2.1 Add noise

Adding noise to the audio signal data can help the machine learning models to generalize the function better. For audio emotion recognition models, adding noise to the dataset can give the model an edge for better accuracy. Let us see how a typical audio speech sample looks after plotting the signal:

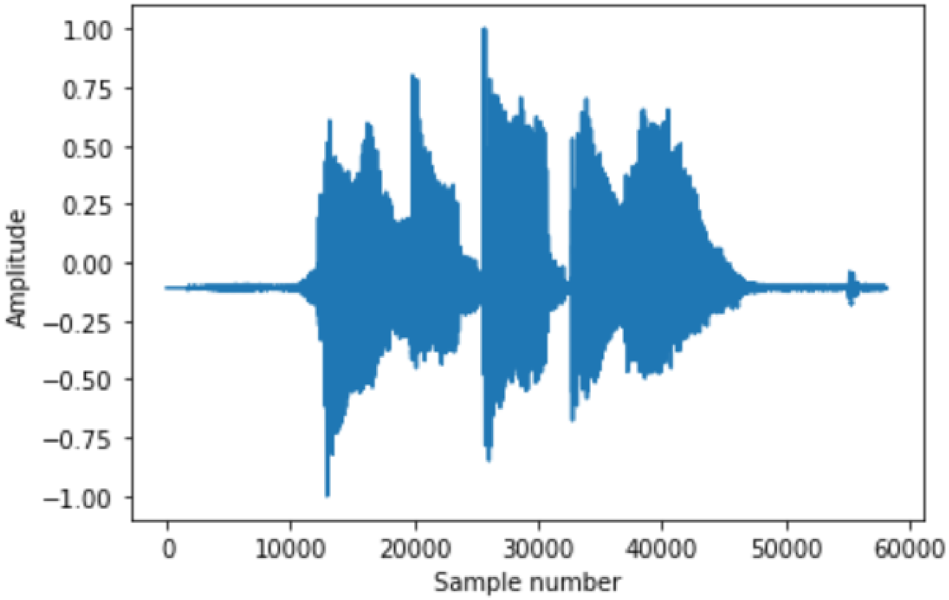


Figure 4: Noiseless audio

After adding noise such as Additive White Gaussian Noise with a sample audio speech, here is what the plot looks like:

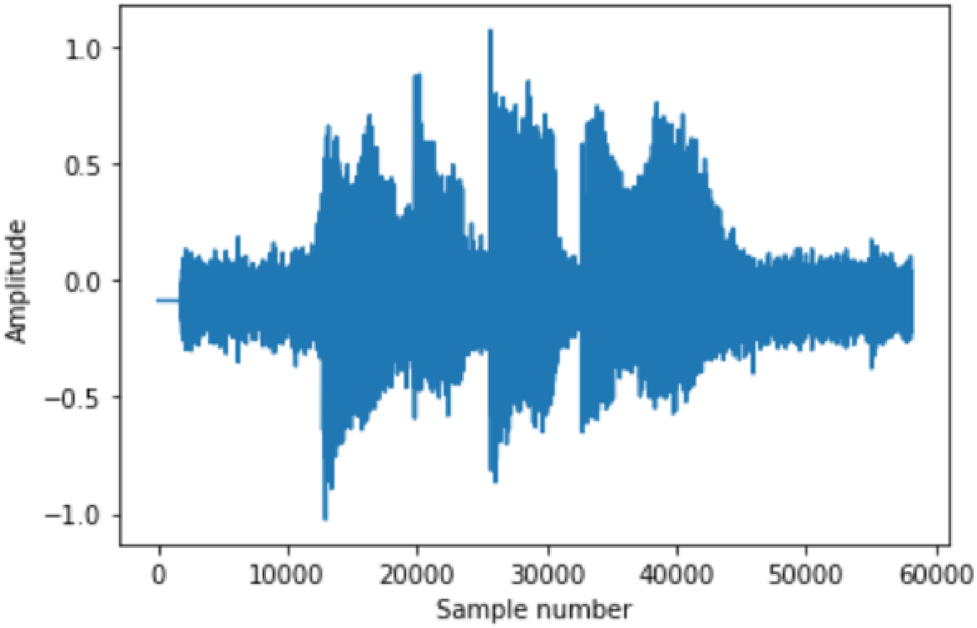


Figure 5: Audio with Gaussian noise.

4.2.2 Signal Loss

Recording audio can suffer from loss of signals in the natural environment due to different hardware and latency issues. Most of the audio datasets are created in a noise-free environment for the clarity of the data. The machine learning models need to perform better in natural environments too. That is why the dataset it is training on should resemble characteristics of the natural environment. Hence, signal loss is applied to audio signals for augmentation.

4.2.3 Change volume

Generally, well-curated audio speech datasets maintain a steady level of volume. The people who create audio datasets are given proper rest to lessen their fatigue from long hours of audio sessions to maintain the same level of tone throughout the recording sessions. On a real-life environment, people do not talk like we trained actors with the systems. Sometimes they talk loudly. If the machine learning system is not robust to the loudness of the speech or environment, it can perform inaccurately. That is why for augmentation, we seldom change the volume of some data just for the machine learning model's generalization purpose.

4.2.4 Spec Augmentation

Google has introduced a new augmentation method called SpecAugment [48] for automatic speech recognition. Conventional augmentation procedures are done over the audio signals. Googles SpecAugmentation applies the augmentation process on the spectrogram of the audio which is an image representation of the signal. This method does not cause any additional computational cost or data like other augmentation methods. After creating the spectrograms, SpecAugment distorts the spectrograms in horizontal and vertical axis which are respectively Mel-frequency channels and time steps. This augmentation technique gives the neural network models generalization above loss of information in the speech signals.

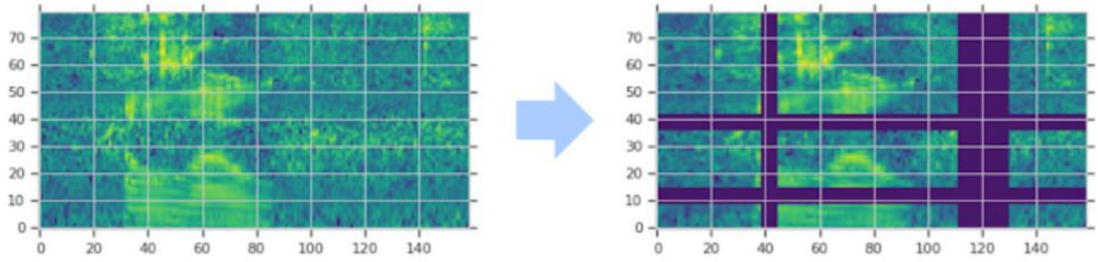


Figure 6: Augmented Spectrogram

Chapter 5

Methodology

In the beginning, speech audio signals are taken as input from the user, which can be any command or conversational speech. The system analyzes the audio signals and extracts various useful features like Mel-spectrogram, Mel-frequency cepstral coefficients (MFCC), Delta of coefficients. These features work as a direct input to different models of our systems, which generates expected results.

After that, the system uses the Parallel CNN and Transformer-Encoders [49] model taking Mel-spectrograms as features to classify the audio signal's seven emotional states of the user's speech, i.e., anger, happiness, disgust, sadness. The emotional state will help further models to get the context of the speech.

Then, the spectrograms of speech audios are fed into an RNN based speech-to-text model DeepSpeech [50], to generate an English text transcript. The default DeepSpeech model does not produce expected transcription well for the south east Asian accent for the English language. We get the word error rate (WER) of 0.44. That is why we tuned the DeepSpeech model on a consortium of a high-quality corpus of 13 major Indian languages [44], which achieved a WER of 0.18.

At this moment, our system knows the "emotional state" of the user's speech and a transcription of what the user says to the system. These two attributes will be used by a multi-domain conversational agent[38] to generate contextual reply text for the user. The reply texts of the conversational agents will further be used for text-to-speech models.

Furthermore, After getting reply texts from the conversational agent, the system will feed them into Flowtron [51] which will not only generate Mel-spectrograms for speech synthesis from text but also control different aspects of speech synthesis such as pitch, tone, speech rate, accent. This will make the synthesized speech as human likely as possible. Also, emotional states can be added with these synthesized voices by transferring styles of given data. For example, if we want to generate an angry state of the synthesized voice, we will give angry emotional audio clips to the trained model function, and it will manipulate the synthesized speech to generate an angry version. Mel-spectrograms generated by Flowtron will be used by another model called WaveGlow [36] to generate speech audio signals. Thus, the user will hear conversational agents' reply with proper human-like voice along with human-like emotional states poured into the synthesized voice.

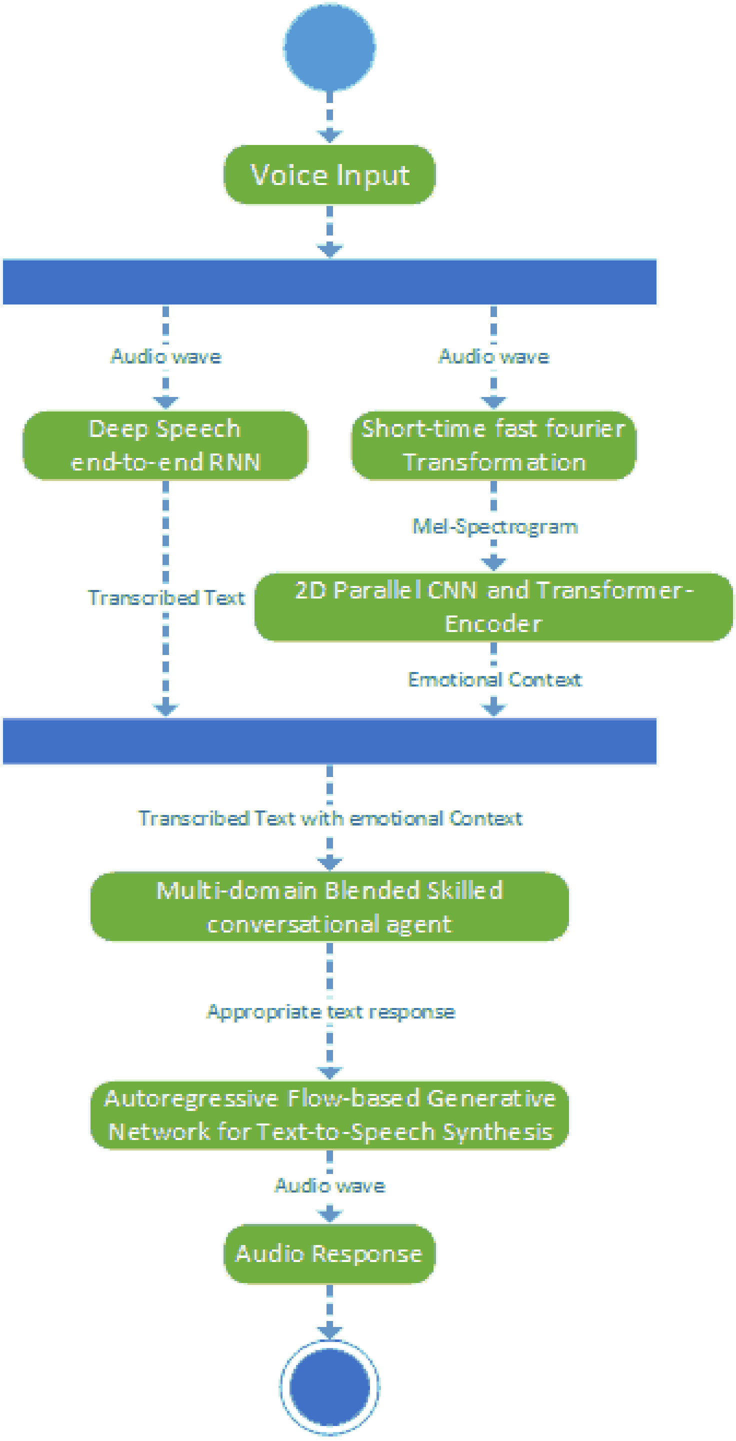

Voice Input
Audio wave
Audio wave
Deep Speech end-to-end RNN
Short-time fast fourier Transformation
Mel-Spectrogram
Transcribed Text
2D Parallel CNN and Transformer-Encoder
Emotional Context
Transcribed Text with emotional Context
Multi-domain Blended Skilled conversational agent
Appropriate text response
Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis
Audio wave
Audio Response

Figure 7: System's Flow Diagram

Chapter 6

Model Architecture

6.1 2D Parallel CNN and Transformer-Encoders

To take advantage of CNN's image classification and feature representation capacity, we need to represent our extracted audio features like MFCC, the Mel Spectrogram graph as an image. Each value of the MFCC/Mel Spectrogram is the amplitude of the audio at a given Mel frequency range at a given time. Transformers are particularly good at predicting future frames/data. Since this is time-series information, we can use the Transformer to find the temporal relationship between pitch change and predict the future frequency distribution of particular emotions. This approach is the successor to the LSTM-RNN model that we tried earlier in our experiment. We use Mel spectrogram as our experimental feature for this model. Like all previous ones, this classification has seven emotional groups and four emotional datasets. Not all data is distributed proportionally. We need to divide them into train, test, validation data while preserving proportionality.

Utilizing the wisdom of previous CNN papers' findings, the proposed model was developed[49]. Conv, Pool, Conv, Pool, FC layer pattern was implemented in the architecture of LeNet. AlexNet presents the idea of increasing the sophistication of features by channel expansion using stacked CNNs. Parallelization was inspired by GoogLeNet [52] and Inception, which lets us diversify the features we learn from the data. The idea of using a smaller size kernel comes from VGGNet, which replaces AlexNets (11 x 11), stride 5 with (3 x 3) kernel, and gains significant improvement over it.

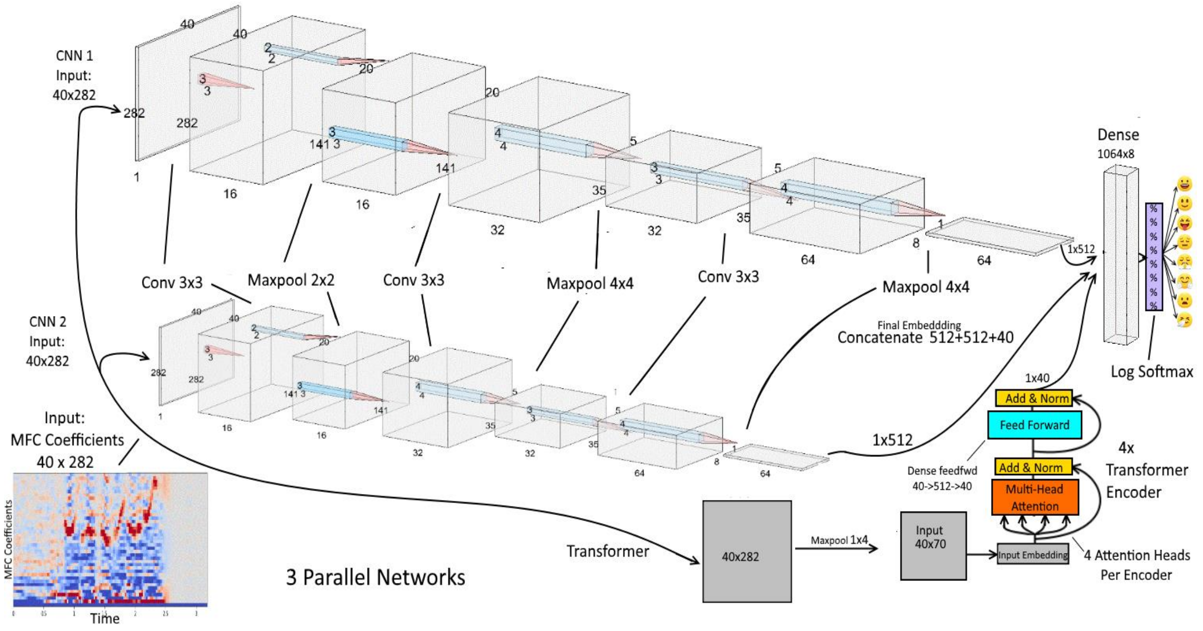


Figure 8: Architecture of 2D Parallel CNN & Transformer-Encoder

CNN with 2D Conv layers is the de facto methodology for image processing. For our case, we have to imagine the Mel-Spectrogram plot as a single channel black and white image. There are two primary reasons for using two stacked filters: feature sophistication and efficiency. If we stack three layers of (3 x 3) kernels, in the second stack, the kernel will sell (5 x 5) view, and the third stack will see (7 x 7) view of the original input. On the other hand, If we used a single (7 x 7) layer, it would have performed only a linear transformation. Moreover, we have been able to minimize excessive computation by using a stacked kernel. If we take the channel as constant, then for (3 x 3) kernel, we will have 27C^2 parameters, whereas (7 x 7) kernel will have 49C^2 parameters. In summary, using smaller stacked kernels, we are getting more intricate features and making the model more efficient. The sequential expansion of filter complexity and reduction in feature maps will give us the best hierarchical features with the lowest possible computation cost.

The motivation for the transformer encoder is to learn the temporal features and hope that it will be able to learn the frequency distribution of different emotions according to the global structure of the Mel-spectrogram of each emotion. RNN-LSTM was a possible candidate for this job, but it would have learned to predict the frequency changes according to time steps. The nature of the Transformer allows it to look at multiple different timestamps using a multi-head self-attention layer, which will, in turn, let us predict the next. As the transformers are very good at generating sequential data, the author expected it to perform well by looking at the entire sequence of frequencies, not just one timestamp. Max-Pooling the input Mel-Spectrogram map to the Transformer dramatically reduces the complexity and number of parameters.

Initially, "Adam" optimizer was used because it usually works decently out-of-the-box. But due to the fact that better performance is achievable by the good old SGD, the author changed the optimizer later with the highest momentum leading to convergence and acceptably long training time.

6.2 DeepSpeech: End-To-End RNN

At Silicon Valley AI Lab, Baidu researchers have made a well-optimized end-to-end RNN speech training system called "Deep Speech" with novel data synthesis techniques to obtain ample amounts of varied data for training achieving a 19.1% error rate on noisy speech dataset produced by them. [50]

The system takes spectrograms of speech audios and generates the text transcription in English. The training set that is arranged for this system is, $X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(y)}), \ldots\}$where x is a single utterance and y is denoted as a label. A single utterance $x^{(i)}$is a collective of vectors of audio features in a time-series of length $T^{(i)}, x^{(i)}\quad_t;\ t\ =\ 1,2,\ldots,T^{(i)}$. The objective of the RNN is to convert an input sequence x into a character probability sequence for the text transcription $\hat{y}_t = \mathbb{P}(c_t \mid x); c_t\ \in\ \{a,..,z, space, blank, apostrophe\}$[50].

The RNN model comprises five hidden layers. The units of the hidden layer are denoted as $h^{(l)}$; l represents each layer. Among the hidden layers, the first three layers are non-recurrent. The fourth layer is a bi-directional recurrent layer [53]. The fifth hidden layer unites both forward and backward units of bi-directional recurrent layers.

At first, the first layer takes spectrogram frame x_t ; $t = each\ time\ slice$ as well as a context of C frames. For each time step t, the second and third non-recurrent layers work by taking independent data. The computational function of the first three layers is:

$$h^{(l)}{}_t = g\left(W^{(l)} h^{(l-1)}{}_t + b^{(l)}\right); W^{(l)} = weight, b^{(l)} = bias, l = current\ layer$$

where $g(x) = min\{max\{0, x\},20\}$is a rectified-liner (ReLu) activation [54] function. After that, the fourth bi-directional layer is created by two hidden units: forward recurrence $h^{(f)}$and backward recurrence $h^{(b)}$. The computational function of both units is:

$$h^{(f)}{}_t = g(W^{(4)} h_t{}^{(3)} + W^{(f)}{}_r\, h^{(f)}{}_{t-1} + b^{(4)})$$

$$h^{(b)}{}_t = g\left(W^{(4)} h_t{}^{(3)} + W^{(b)}{}_r\, h^{(b)}{}_{t-1} + b^{(4)}\right)$$

In the case of forwarding recurrence, for each utterance i, $h^{(f)}$is computed sequentially $t = 1$to $t = T^{(i)}$. On the other hand, for backward recurrence $h^{(b)}$is computed sequentially in reverse order, $t = T^{(i)}$to $t = 1$. Both forward and backward hidden layers are combined and fed into the fifth layer. The computational function of the fifth layer, which is not recurrent, is:

$$g(W^{(5)} h_t{}^{(4)} + b^{(5)})\ ;\ h_t{}^{(4)} = h^{(f)}{}_t + h^{(b)}{}_t$$

Finally, the output layer predicts the character probabilities with the help of the standard SoftMax function:

$$h^{(6)}{}_{t,k} = \hat{y}_{t,k} \equiv \mathbb{P}(c_t = k \mid x) = exp(W_k{}^{(6)} h_t{}^{(5)} + b_k{}^{(6)})\ /\ \Sigma_j exp(W_j{}^{(6)} h_t{}^{(5)} + b_j{}^{(6)}$$

$$W_k{}^{(6)} = k^{th}\ column\ of\ weight\ matrix, b_k{}^{(6)} = k^{th}\ bias\ \ [50]$$

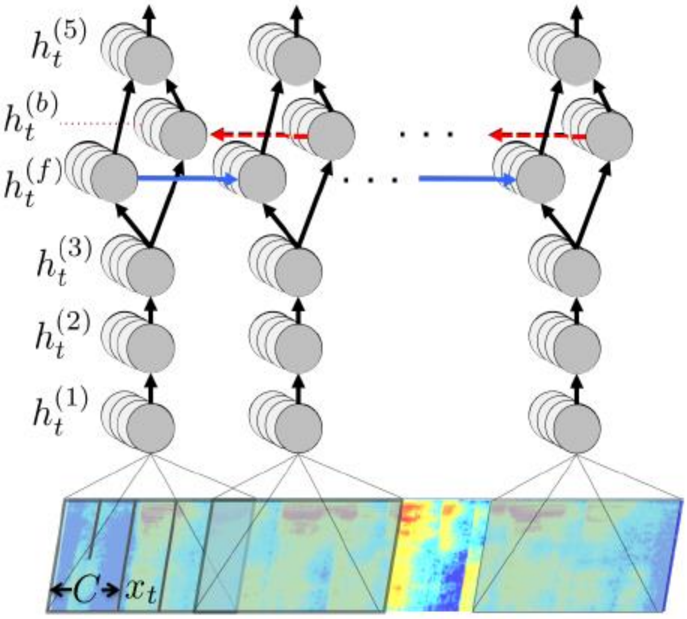


Figure 9: Structure of DeepSpeech's RNN model and notation

6.3 Multi-domain conversational agent

To have a neutral conversation, an agent must have several skills, such as engaging, knowledgeable, and empathetic, while sticking to the personality. Many prior approaches sought to acquire these abilities in isolation, but the actual human-like conversation goal was not accomplished. A team of Facebook researchers showed that these skills could be taught to a broad number of models if we provide adequate training data and generation strategy. Recently published, Blended Skill Talk[55] (BST) offers conversational context and training data that can be used to train multi-domain human-like conversational agents.

The generation algorithm is also an indispensable part of the process. A model with the same accuracy but with a different generation algorithm can produce a completely different result. The authors also noted that the length of utterance plays a significant role in human judgment, whether it is engaging or not. According to the experiment, a too-short utterance can make the human judge perceive the bot as dull and uninterested. On the other hand, too-long utterances make the judge feel the bot is not listening and distracted. Despite the previous report of beam searching being inferior to sampling[37], [56], the study shows that by tweaking the minimum beam length, control over the dull versus spicy response generation can be achieved.

In this study, three types of architecture were used: Retriever, Generative, and retrieve-and-refine models. All of which were derived from the Transformer model.

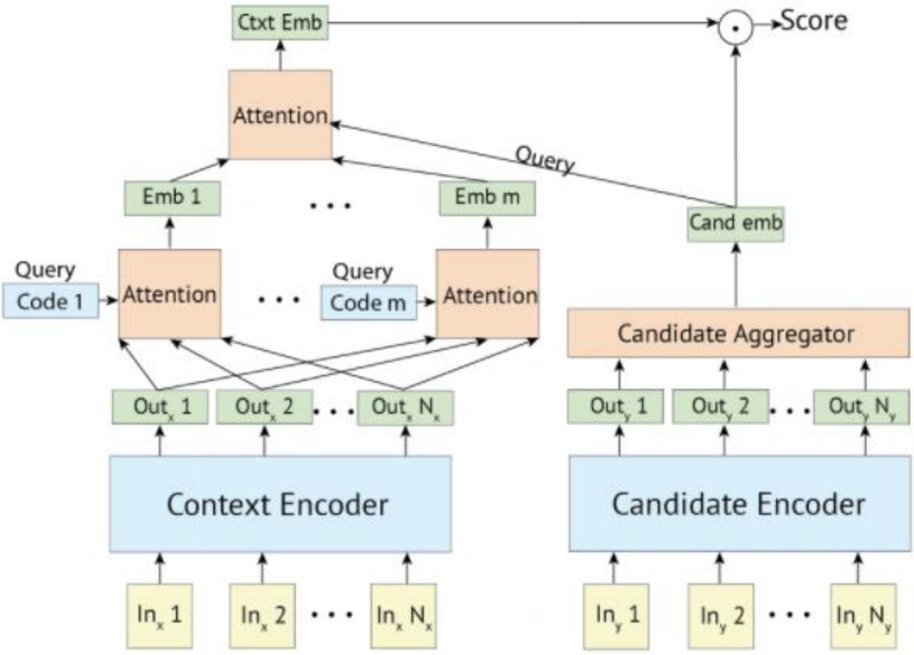
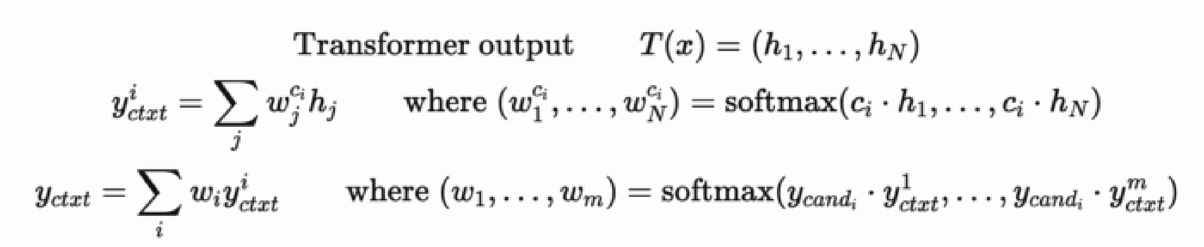


Figure 10: The Poly-encoder[57] Transformer architecture for retrieval encodes global features of the context using multiple representations (codes)

The Retriever model works by scoring the set of possible responses and outputting the highest probable one, given we have conversation history as input. The researchers used poly-encoder architecture[57] to encode global features of the context using several representations attended to by each potential candidate response[58]. The final attention mechanism allows us to achieve better performance over a single global vector representation. It generates context embedding and dot product it with each response candidate. The embeddings are created in two steps. Firstly, the model gets the candidate embedding using a transformer-based encoder and an aggregator function that takes the classifier embedding output or the token's mean. After that model encodes the context using another transformer and performs an "m" attention block. Each attention uses the Transformer output as keys and values, and the learned c_i code is unique for each attention. On top of this embedding, another attention is calculated. The key and values are the output from the other attention.



$$\text{Transformer output} \qquad T(x) = (h_1, \ldots, h_N)$$

$$y^i_{ctxt} = \sum_j w^{c_i}_j h_j \qquad \text{where } (w^{c_i}_1, \ldots, w^{c_i}_N) = \text{softmax}(c_i \cdot h_1, \ldots, c_i \cdot h_N)$$

$$y_{ctxt} = \sum_i w_i y^i_{ctxt} \qquad \text{where } (w_1, \ldots, w_m) = \text{softmax}(y_{cand_i} \cdot y^1_{ctxt}, \ldots, y_{cand_i} \cdot y^m_{ctxt})$$

One of the most significant benefits of poly-encoders is that it gives a state-of-the-art performance on some dialogue tasks compared to other retrieval methods on ConvAI2 competition tasks based on human evaluation.

The generator approach is similar to the seq2seq model proposed in the Transformer [58] paper, but the main difference is that it is a lot bigger. For comparison, Google's Meena[37] has 2.7B parameters, whereas the blender model has 90M, 2.7B, 9.4B parameter versions.

Lastly, there is the retrieve and refine the approach. It mixes the previously mentioned two models. The retrieval model's output goes as an input of the generative model using a unique separator token. Utilizing this method, the authors tried to mitigate the known shortcomings like knowledge hallucination, disability to read new and external knowledge, dull and repetitive answers. They worked with two types of retrieval models: dialogue retriever and knowledge retriever. Dialogue retriever uses dialogue history to generate a response. Knowledge retriever gets its information from a large knowledge base. In this scenario, a transformer is trained to determine whether a knowledge retriever should be used.

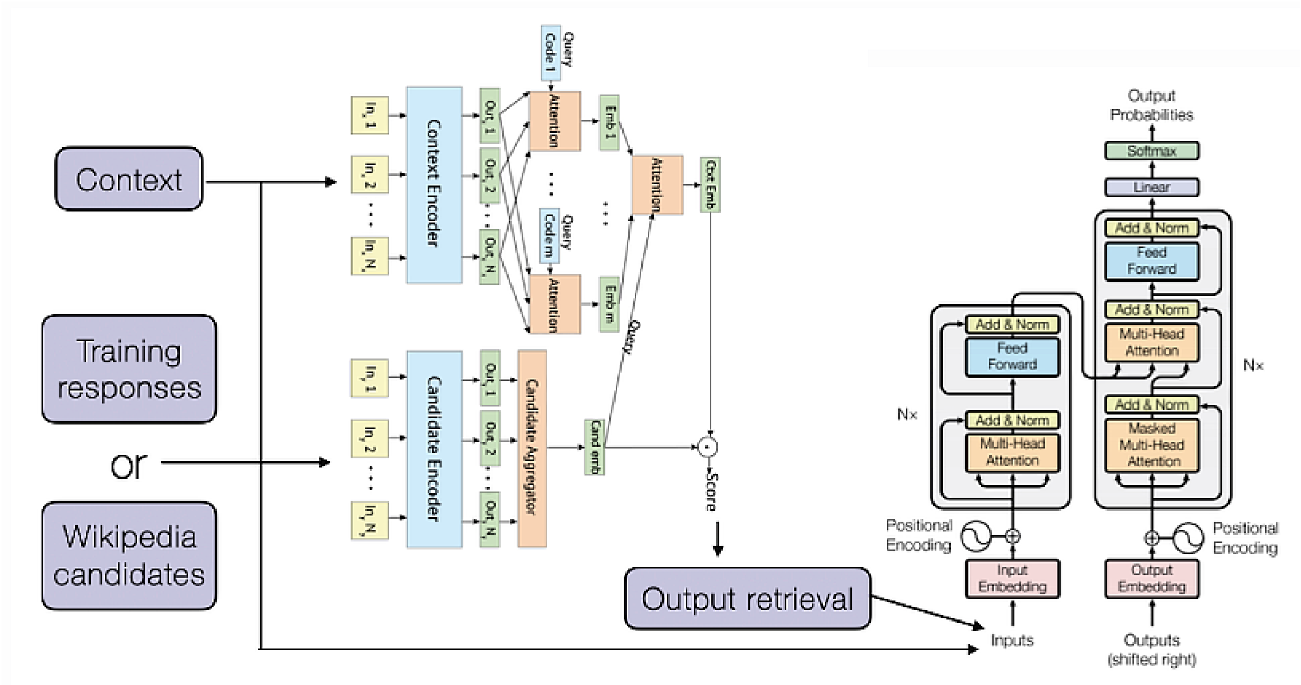


Figure 11: Retrieve and Refine architecture.

For decoding, Beam Search, Top-K-sampling, sample-and-rank-sampling strategies were used. There are many different algorithms to decode the final output sequence as our model gives a probability distribution over the vocabulary. We need to select one word at a time until we reach the end of the statement. We can use greedy algorithms to choose the best word each time, but the final result may not be the overall best probable sentence. To mitigate this, we predict beam_size (possible sentences). At each step, we predict the next beam_size token for each sentence and select the one with the most probable beam_size. We stop if we reach the end character (complete "n" sentences) or after t steps. Next comes the Top-K sampling algorithm. Here, at each step, the word "i" is chosen by sampling the model distribution from the "k" most likely candidates.

Along with the decoding process, some additional constraints were tested. Minimum length forced the model to produce a result with a defined length. Another one was a predictive retriever model, which predicts the sentence's length and limits the generation to that length.

The last one was beam blocking, where the model was forced not to generate any trigram, a group of 3 words, in the next utterance if that is in the input or utterance itself.

6.4 Autoregressive Flow-based Generative Network for TTS Synthesis

Text is required for Mel-spectrogram synthesis, which will have non-textual information such as tone, accent, pitch. Also, non-textual information needs to follow the style of the given audio data. If we give the model some audio data of a particular emotion, such as anger, surprise, etc., the synthesized Mel-spectrogram should copy the style we refer to as "Style Transfer." NVIDIA researchers have introduced a model called "Flowtron," which does exactly the same thing as mentioned above. Flowtron does this by maximizing the probability of training data. Flowtron learns an invertible mapping of data to a latent space that can be manipulated to control many aspects of speech synthesis[59], including pitch, accent, speech rate, tone, etc. It generates a Mel-spectrogram frame based on previous Mel-spectrogram frames.

The whole sequence of frames is $p(x) = \prod p(x_t \mid x_{1:t-1})$. Two types of distributions, p(z), are used to be sampled by the neural network, which is used as a generative model in the flowtron. The first distribution is a zero-mean spherical Gaussian, z ∼ N (z; 0; I). The other one is a mixture of spherical Gaussian with fixed or learnable parameters. The samples are transformed into p(x) from p(z) by going through "affine transformations," which are invertible and parameterized transformation. We know that flowtron uses an invertible neural network. Invertible neural networks are constructed using coupling layers [35] [60], in this case, affine coupling layer [61]. For each input x_{t-1}a scale, s , and a bias is produced. This scale and bias affine transforms the next input x_t:

$$(log\ s_t, b_t) = NN(x_{1:t-1}, text\ , speaker)$$

$$x'_t = s_t \odot x_t + b_t$$

Here, $NN()$denotes any autoregressive causal transformation. A zero vector is concatenated with other inputs of $NN()$ to implement this. The $NN()$needs not to be invertible, but the affine coupling layer preserves the whole network's invertibility. In the autoregressive structure, every t-th variable z'_tdepends on its previous timesteps from the star $z_{1:t-1}$: [62]

$$z'_t=f_k(z_{1:t-1})$$

Flowtron is maximizing the data's log-likelihood by utilizing the parameterized affine transformation and the autoregressive structure mentioned above. These are possible by using the change of variables:

$$log\, p_\theta(x) \ = \ log\, p_\theta(z) \ + \sum^{k}{}_{i=1} log \mid det(J\,(\,f_i{}^{-1}(x))) \mid$$

$$z \ = \ f^{-1}{}_k \circ f^{-1}{}_{k-1} \circ ...\, f^{-1}{}_0(x)$$

Mel-spectrograms are converted as vectors and run through several affine coupling layers conditioned on the text and fixed dummy speaker embedding. Each affine coupling layer is called the "flow." Finally, the processed vectors are forwarded to pass through the neural network.

Randomly sampled z values from Gaussian Mixture or spherical Gaussian with fixed or flowtron predicted parameters are run through the trained neural network to infer. The inferred Mel-spectrograms are decoded into waveforms using a single pre-trained WaveGlow [36] model trained on a single speaker.

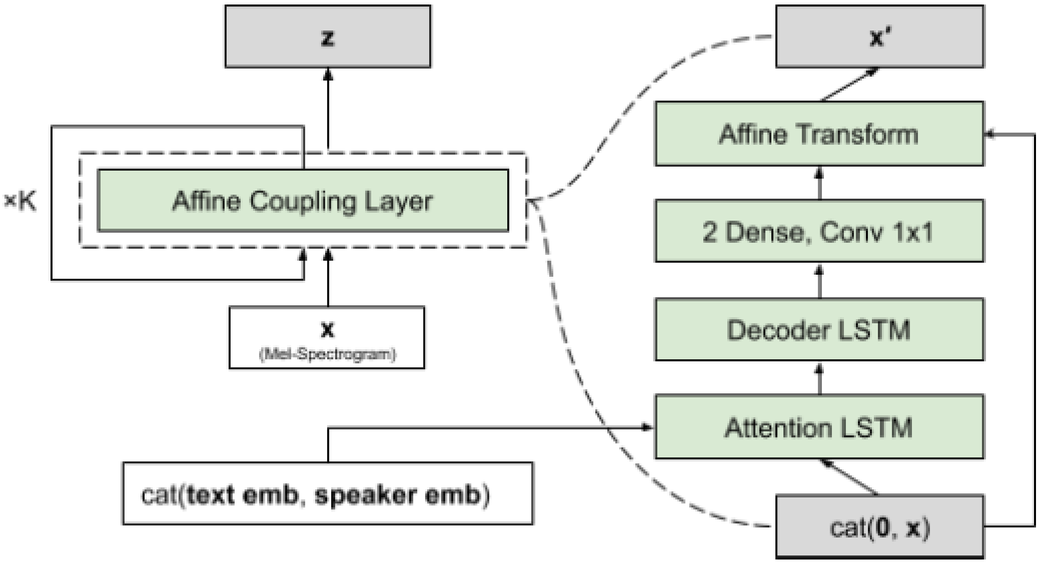


Figure 12: Flowtron Network[59]. Text and speaker embeddings are channel-wise concatenated. A 0-values vector is concatenated with x in the time dimension.

Chapter 7

Implementation Result

7.1 Measurements

7.1.1 Word Error Rate (WER)

The word error rate is based on the Levenshtein distance[63]. It is computed by the minimum number of operations, i.e., insertion, deletion, substitution, to be performed to generate a text hypothesis that is similar to the reference text data. The computational function of WER is:

$$\mathrm{WER} \quad = \quad \frac{1}{N^{*}_{ref}} \sum_{k=1}^{K} \min_{r} d_L(ref_{k,r}, hyp_k)$$

here $d_L(ref_{k,r}, hyp_k)$ is the Levenshtein distance from hyp_k to $ref_{k,r}$.

7.1.2 F1- Score

F1 score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is used to balance precision and recall. F1 score is better than accuracy in cases where class distribution is uneven. So, for our case, we took the F1 score as our measurement Matic.

7.1.3 Mean Opinion Score

Mean opinion score (MOS) is a measure used in the video, audio, and audiovisual, representing the overall quality of a stimulus or system. It is the arithmetic mean over all individual "values on a predefined scale that a subject assign to his opinion of the performance of a system quality." Such ratings are usually gathered in a subjective quality evaluation test, but they can also be algorithmically estimated. It is expressed as a single rational number, typically in the range 1–5, where 1 is the lowest perceived quality, and 5 is the highest perceived quality. This metric is calculated using the arithmetic mean over a single rating performed by humans.

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N}$$

Here, R is the rating given for the clip, and N is the number of participants. We will compare two of our models using this metric to determine which one is better and also provide a real demonstration.

7.2 Model Implementation

7.2.1 Sequential 1D CNN

We started our emotion classifier from scratch. To get the feel of the MFCC feature, we used the mean value of the feature to determine the class of the emotion using a 1D Convolutional Neural Network. The approach was too naive and was not able to produce a good result. We removed the gender class (reducing it to 7 from 14) to make it more predictable, but the best result we could produce is 50.28% accuracy with 100 Epoch and MFCC value of 13.

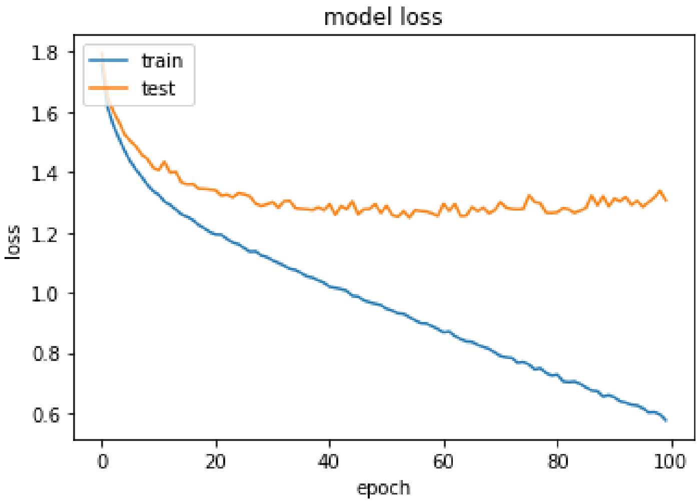
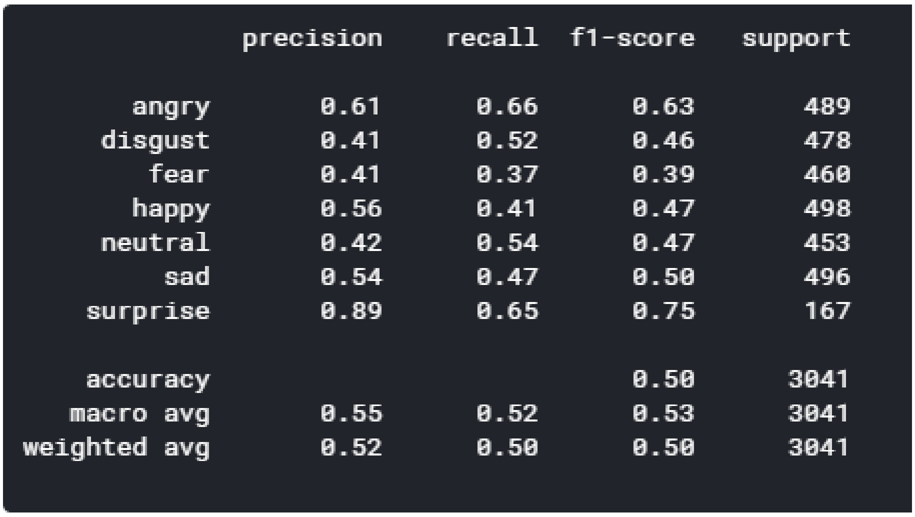


Figure 13: Best 1D CNN loss graph



```
              precision    recall  f1-score   support

       angry       0.61      0.66      0.63       489
     disgust       0.41      0.52      0.46       478
        fear       0.41      0.37      0.39       460
       happy       0.56      0.41      0.47       498
     neutral       0.42      0.54      0.47       453
         sad       0.54      0.47      0.50       496
    surprise       0.89      0.65      0.75       167

    accuracy                           0.50      3041
   macro avg       0.55      0.52      0.53      3041
weighted avg       0.52      0.50      0.50      3041
```

Figure 14: Best 1D CNN accuracy

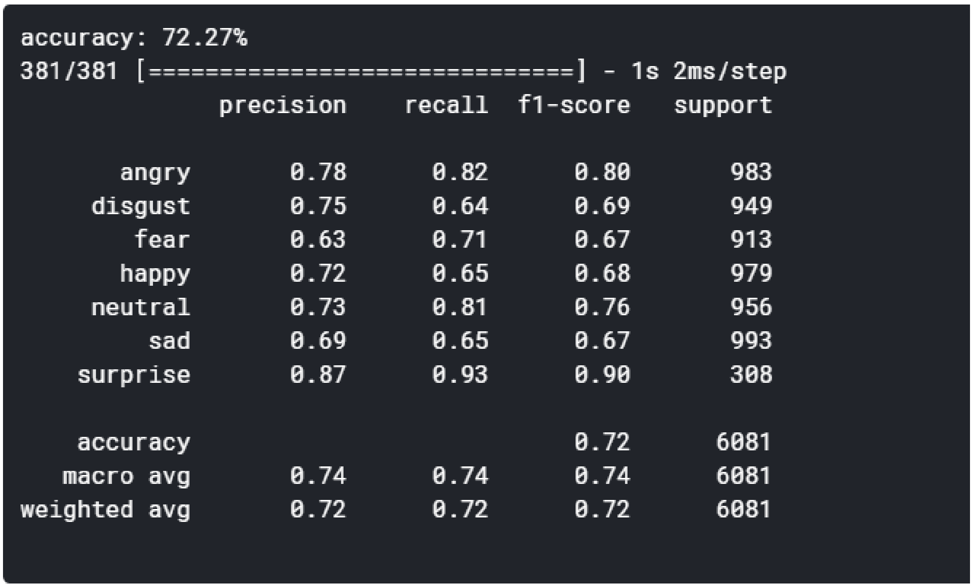
Table 1: Comparison of 1D models

Model	**Epoch**	**Optimizer**	**Class**	**Accuracy**	**Parameters**	**Augmentation**
1D_CNN	100	RMSprop	14	42.98%	n_mfcc=13	no
1D_CNN	100	RMSprop	14	49.02%	n_mfcc=13	no
1D_CNN	100	RMSprop	7	50.28%	n_mfcc=13	no

After seeing the result, we came to the conclusion that it would not be a very smart idea to spend on this approach, so we moved onto the next method.

7.2.2 Sequential 2D CNN

Then we used the MFCC values to create an image and use Convolutional Neural Network to classify the image. By classifying the image, we were able to classify the emotion as well. The first trial gave us a somewhat hopeful result, so we went further with it. We tried different parameters and tried to tweak the model. The initial accuracy was 67.08%. Using only the augmented data seemed to reduce the accuracy even more. When we added augmented data and real data together and trained the model with it, we got the best result. MFCC coefficient also plays a decent role in increasing accuracy as it increases the resolution of the feature. The number of epochs also positively influences the accuracy result to a certain degree. After that, we get diminishing returns. For 100 epochs, we got an accuracy of 72.26%. After that, we increased the epochs value by 50%, making it 150, but the result accuracy was increased by 0.01%.



```
accuracy: 72.27%
381/381 [==============================] - 1s 2ms/step
              precision    recall  f1-score   support

       angry       0.78      0.82      0.80       983
     disgust       0.75      0.64      0.69       949
        fear       0.63      0.71      0.67       913
       happy       0.72      0.65      0.68       979
     neutral       0.73      0.81      0.76       956
         sad       0.69      0.65      0.67       993
    surprise       0.87      0.93      0.90       308

    accuracy                           0.72      6081
   macro avg       0.74      0.74      0.74      6081
weighted avg       0.72      0.72      0.72      6081
```

Figure 15: Best 2D CNN accuracy

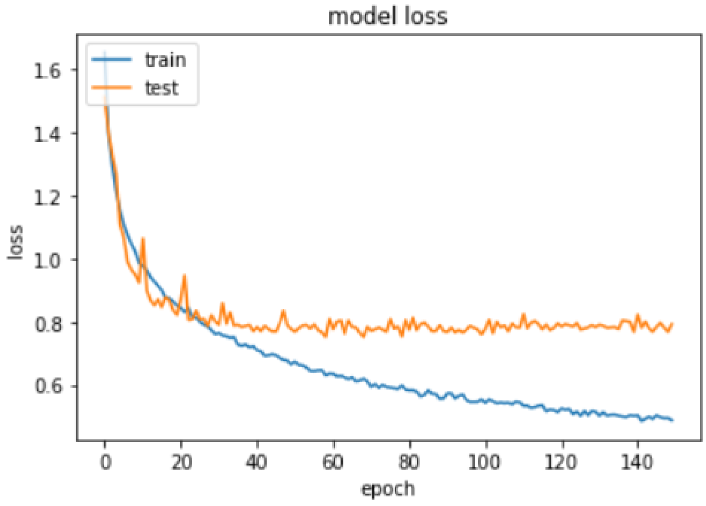


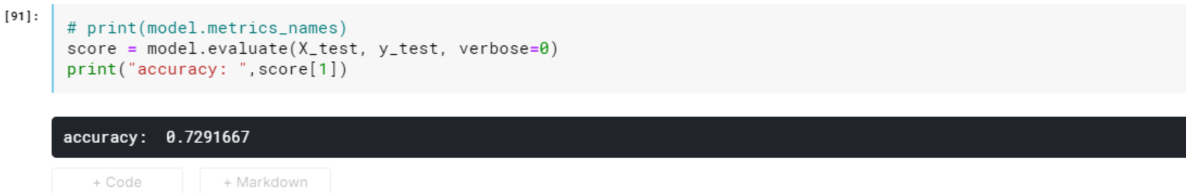
Figure 16: Best 2D CNN loss graph

Table 2: Comparison of 2D CNN models

Model	Epoch	Optimizer	Class	Accuracy	Parameters	Augmentation
2D_CNN	100	Adam	7	67.08%	mean(n_mfcc = 30)	no
2D_CNN	50	Adam	7	67.58%	mean(n_mfcc = 30)	no
2D_CNN	35	Adam	7	69.58%	mean(n_mfcc = 30)	no
2D_CNN	100	Adam	7	69.42%	mean(n_mfcc = 30)	yes
2D_CNN	50	Adam	7	66.95%	mean(n_mfcc = 30)	yes
2D_CNN	35	Adam	7	68.07%	mean(n_mfcc = 30)	yes
2D_CNN	100	Adam	7	68.33%	mean(n_mfcc = 50)	no
2D_CNN	50	Adam	7	68.79%	mean(n_mfcc = 50)	no
2D_CNN	35	Adam	7	69.45%	mean(n_mfcc = 50)	no
2D_CNN	100	Adam	7	72.26%	mean(n_mfcc = 30)	yes
2D_CNN	50	Adam	7	70.84%	mean(n_mfcc = 30)	yes
2D_CNN	35	Adam	7	69.79%	mean(n_mfcc = 30)	yes
2D_CNN	150	Adam	7	72.27%	mean(n_mfcc = 30)	yes

7.2.3 CNN-LSTM

We also tried the CNN-LSTM model. We only ran this model on a RAVDESS dataset, but the result it produced was indeed promising, but from model loss graphs, we can get the idea that this model is not stable. The first time we ran the model, we got an accuracy of 59.02%, but after running the model again, the accuracy jumped to 72.91%. We ran the experiment several times but could not make the model stable enough for our use. Thus, we abandoned the method.



```
[91]: # print(model.metrics_names)
      score = model.evaluate(X_test, y_test, verbose=0)
      print("accuracy: ",score[1])
```

```
accuracy:  0.7291667
```

Figure 17: CNN-LSTM best accuracy

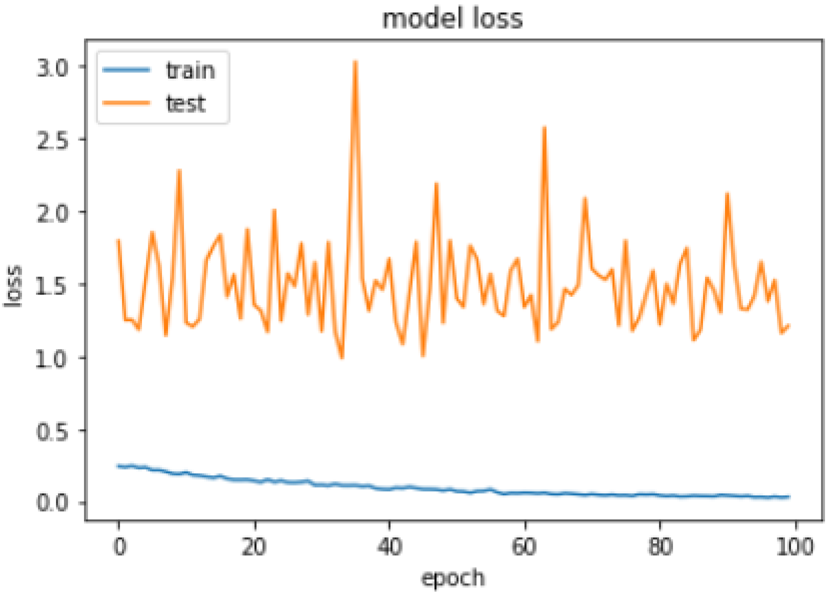


Figure 18: Unstable loss graph of CNN-LSTM

Table 3: CNN-LSTM Accuracy Result

Model	**Epoch**	**Optimizer**	**Class**	**Accuracy**	**Feature**	**Augmentation**
CNN-LSTM	100	SGD	8	72.91%	Log-mel spectrogram	yes

7.2.4 XResnet Models (Transfer Learning)

xResnet50[64] is one of the most popular architectures in computer vision research. We tried to experiment with xResnet50 and xResnet18, which are relatively small in size but good in terms of training speed and accuracy. We used both heavy and light augmentation for the experiment and observed the performance. The augmentation includes removing silence, the addition of white noise, signal loss, changing volume, resize, using different spectrogram-augmentation [48], which modifies the spectrogram to gain robustness against deformation of spectrograms in the time direction. For features, we have chosen MFCC images with delta, Mel-Spectrogram with parameters optimized for voice speeches to experiment with. We did not get any significant accuracy improvement from these experiments. We also found that applying heavy augmentation decreases the accuracy of the models.

Table 4: Accuracy comparison of various xResNet models

Model	Feature	Data	Accuracy	Epoch	Augment
xresnet50	MFCC + Delta	RAVDESS, CREMA-D, TESS, SAVEE	0.698602	25	RemoveSilence,AddNoise, SignalLoss, ChangeVolume, resize
xresnet18	MFCC + Delta	RAVDESS, CREMA-D, TESS, SAVEE	0.701891	25	AddNoise, resize
xresnet50	Mel Voice	RAVDESS, CREMA-D, TESS, SAVEE	0.715049	15	AddNoise, resize
xresnet50	Mel Voice	RAVDESS, CREMA-D, TESS, SAVEE	0.680921	15	RemoveSilence(Trim), Resize, MaskTime(size=4), MaskFreq(size=10)

7.2.5 VGG19 (Transfer Learning)

VGG19 [65] is another version of the VGG model with 19 layers, including 16 convolution layers, three fully connected layers along with five max-pooling layers, and one SoftMax layer. This model is a relatively older and simple mode, but still, it is an effective one. That motivated us to experiment with this model. The reason behind our experimentation with VGG19 is that it is just another decent classification architecture for images that works well despite being very simple and transfer learning is possible with this architecture. Just like xresnet models, we used MFCC and delta features for VGG19. We also augmented the datasets by removing silence, resizing the images, changing volume, adding noise, and signal loss. However, the model did not perform up to the mark for the datasets that we used.

Table 5:Accuracy comparison of VGG19 models

Model	Feature	Data	Accuracy	Epoch	Augment
VGG19	MFCC + Delta	RAVDESS, CREMA-D, TESS, SAVEE	0.707237	15	RemoveSilence(Trim), AddNoise, SignalLoss, ChangeVolume, resize
VGG19	MFCC + Delta	RAVDESS, CREMA-D, TESS, SAVEE	0.685033	15	AddNoise, resize

7.2.6 Parallel 2D CNN with Transformer-Encoder

Parallel 2D CNN has a sequential expansion of filter complexity. It reduces feature maps gradually, which gives us high-quality features with better computational performance. Moreover, the transformer encoder learns the frequency distribution of different emotional categories by focusing on the temporal features. Mel-spectrograms of the audio signals has been used as a feature for this model. We have added Gaussian white noise to the data as a procedure of augmentation. We have got noticeable accuracy for the respective dataset that we used for this model.

Table 6: Accuracy of Parallel 2D CNN with Transformer Encoder model

Model	Feature	Data	Accuracy	Epoch	Augment
Parallel 2D CNN with Transformer Encoder	Mel Spec	RAVDESS, CREMA-D, TESS, SAVEE	0.8667	400	AddGausian WhiteNoise

7.2.7 DeepSpeech Tuning

We have used the DeepSpeech [50] model for the generation of English transcripts from speech audio clips. However, the baseline model did not come up with a promising output for input data given by us. We assumed that it might happen for our accent as we are from the south east Asian region and not native English speakers. That's why we tuned the DeepSpeech model with the Indic TTS [43] project. We had to clean the articles from the datasets and prepared them in a structure that is suitable for the deep speech model. As per our assumption, the tuned model gave a much better result. Better accuracy is possible if we train it more dataset and for a longer time.

Table 7: DeepSpeech accuracy improvement

Metric	Trained Model	Deepspeech Model
Word Error Rate	0.18	0.44

Chapter 8

Conclusion & Future Work

We faced different challenges while implementing the models. A lot of things have changed in the recent few months. NLP is a fast-evolving domain; it is very hard to keep up with the latest and greatest technology. All the tech giants like Google, Facebook, Microsoft, Baidu, Amazon, are competing in this area. While we feel excited to read the papers of the industry titans, but the pressure to keep apace is equally overwhelming. With Flowtron's pre-trained models, we added various style data to further tweak and extend our models. The open-domain conversational model worked as hoped it would. We were also very successful with the accuracy of the Speech-to-Text model that was tweaked for our sub-continent. Overall, the system we build performed way better than we initially thought it would. Unfortunately, there are still many paths which were left unexplored. We noticed, there are not any emotion style datasets available to the public. We would like to create an emotional style dataset, which will be aimed towards audio style transfer. We also think there are ways to tweak the Parallel 2D CNN Transformer-Encoder model. We would like to investigate the possible modification of the model in the future. We think better accuracy in emotion recognition is very much possible by using different audio features, augmentations and tweaking new transformer based models.

Bibliography

[1] R. M. Montoya, R. S. Horton, and J. Kirchner, “Is actual similarity necessary for attraction? A meta-analysis of actual and perceived similarity,” *J. Soc. Pers. Relatsh.*, vol. 25, no. 6, pp. 889–922, Dec. 2008, doi: 10.1177/0265407508096700.

[2] M. Neff, Y. Wang, R. Abbott, and M. Walker, “Evaluating the Effect of Gesture and Language on Personality Perception in Conversational Agents,” in *Intelligent Virtual Agents*, vol. 6356, J. Allbeck, N. Badler, T. Bickmore, C. Pelachaud, and A. Safonova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 222–235.

[3] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, “A Persona-Based Neural Conversation Model,” *ArXiv160306155 Cs*, Jun. 2016, Accessed: Apr. 04, 2020. [Online]. Available: http://arxiv.org/abs/1603.06155.

[4] D. M. Perez Garcia, S. Saffon Lopez, and H. Donis, “Everybody is talking about Virtual Assistants, but how are people really using them?,” presented at the Proceedings of the 32nd International BCS Human Computer Interaction Conference, Jul. 2018, doi: 10.14236/ewic/HCI2018.96.

[5] F. Dellaert, T. Polzin, and A. Waibel, “Recognizing emotion in speech,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP ’96*, Philadelphia, PA, USA, 1996, vol. 3, pp. 1970–1973, doi: 10.1109/ICSLP.1996.608022.

[6] B. Schuller and A. Batliner, *Computational paralinguistics: emotion, affect and personality in speech and language processing*, First Edition. Hoboken, N.J: Wiley, 2014.

[7] G. Trigeorgis *et al.*, “Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network,” in *2016 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), Shanghai, Mar. 2016, pp. 5200–5204, doi: 10.1109/ICASSP.2016.7472669.

[8] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, “Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011,” *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 155–177, Feb. 2015, doi: 10.1007/s10462-012-9368-5.

[9] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognit.*, vol. 44, no. 3, pp. 572–587, Mar. 2011, doi: 10.1016/j.patcog.2010.09.020.

[10] H. Gunes and B. Schuller, “Categorical and dimensional affect analysis in continuous input: Current trends and future directions,” *Image Vis. Comput.*, vol. 31, no. 2, pp. 120–136, Feb. 2013, doi: 10.1016/j.imavis.2012.06.016.

[11] L. Devillers, L. Vidrascu, and L. Lamel, “Challenges in real-life emotion annotation and machine learning based detection,” *Neural Netw.*, vol. 18, no. 4, pp. 407–422, May 2005, doi: 10.1016/j.neunet.2005.03.007.

[12] B. W. Schuller, “Speech emotion recognition: two decades in a nutshell, benchmarks, and ongoing trends,” *Commun. ACM*, vol. 61, no. 5, pp. 90–99, Apr. 2018, doi: 10.1145/3129340.

[13] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[14] X. Glorot, A. Bordes, and Y. Bengio, “Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach,” p. 8.

[15] J. Deng, R. Xia, Z. Zhang, Y. Liu, and B. Schuller, “Introducing shared-hidden-layer autoencoders for transfer learning and their application in acoustic emotion recognition,” in

2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 4818–4822, doi: 10.1109/ICASSP.2014.6854517.

[16] J. Deng *et al.*, "Semisupervised Autoencoders for Speech Emotion Recognition," *IEEEACM Trans. Audio Speech Lang. Process. TASLP*, vol. 26, no. 1, pp. 31–43, Jan. 2018, doi: 10.1109/TASLP.2017.2759338.

[17] J. Liu, C. Chen, J. Bu, M. You, and J. Tao, "Speech Emotion Recognition using an Enhanced Co-Training Algorithm," in *Multimedia and Expo, 2007 IEEE International Conference on*, Beijing, China, Jul. 2007, pp. 999–1002, doi: 10.1109/ICME.2007.4284821.

[18] J. Chang and S. Scherer, "Learning Representations of Emotional Speech with Deep Convolutional Generative Adversarial Networks," *ArXiv170502394 Cs Stat*, Apr. 2017, Accessed: Apr. 02, 2020. [Online]. Available: http://arxiv.org/abs/1705.02394.

[19] Q. You, J. Luo, H. Jin, and J. Yang, "Robust Image Sentiment Analysis Using Progressively Trained and Domain Transferred Deep Networks," *ArXiv150906041 Cs*, Sep. 2015, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1509.06041.

[20] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing Voice Synthesis Based on Deep Neural Networks," Sep. 2016, pp. 2478–2482, doi: 10.21437/Interspeech.2016-1027.

[21] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3687–3691, doi: 10.1109/ICASSP.2013.6638346.

[22] Y. Lee, A. Rabiee, and S.-Y. Lee, “Emotional End-to-End Neural Speech Synthesizer,” *ArXiv171105447 Cs Eess*, Nov. 2017, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1711.05447.

[23] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, “Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder,” *ArXiv180402135 Cs Eess*, Feb. 2019, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1804.02135.

[24] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, “Generative adversarial networks: introduction and outlook,” *IEEECAA J. Autom. Sin.*, vol. 4, no. 4, pp. 588–598, 2017, doi: 10.1109/JAS.2017.7510583.

[25] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative Adversarial Networks: An Overview,” *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.

[26] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” *ArXiv14062661 Cs Stat*, Jun. 2014, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1406.2661.

[27] S. Pascual, A. Bonafonte, and J. Serrà, “SEGAN: Speech Enhancement Generative Adversarial Network,” *ArXiv170309452 Cs*, Jun. 2017, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1703.09452.

[28] K. Kumar *et al.*, “MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis,” *ArXiv191006711 Cs Eess*, Dec. 2019, Accessed: Apr. 03, 2020. [Online]. Available: http://arxiv.org/abs/1910.06711.

[29] Y. Gao, R. Singh, and B. Raj, “Voice Impersonation using Generative Adversarial Networks,” *ArXiv180206840 Cs Eess*, Feb. 2018, Accessed: Feb. 05, 2020. [Online]. Available: http://arxiv.org/abs/1802.06840.

[30] A. van den Oord *et al.*, “WaveNet: A Generative Model for Raw Audio,” *ArXiv160903499 Cs,* Sep. 2016, Accessed: Mar. 14, 2020. [Online]. Available: http://arxiv.org/abs/1609.03499.

[31] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” *ArXiv191011480 Cs Eess*, Feb. 2020, Accessed: Apr. 04, 2020. [Online]. Available: http://arxiv.org/abs/1910.11480.

[32] Y. Wang *et al.*, “Tacotron: Towards End-to-End Speech Synthesis,” *ArXiv170310135 Cs,* Apr. 2017, Accessed: Oct. 06, 2020. [Online]. Available: http://arxiv.org/abs/1703.10135.

[33] J. Shen *et al.*, “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions,” *ArXiv171205884 Cs*, Feb. 2018, Accessed: Oct. 06, 2020. [Online]. Available: http://arxiv.org/abs/1712.05884.

[34] M. Schmitt, F. Ringeval, and B. Schuller, “At the Border of Acoustics and Linguistics: Bag-of-Audio-Words for the Recognition of Emotions in Speech,” Sep. 2016, pp. 495–499, doi: 10.21437/Interspeech.2016-1124.

[35] D. P. Kingma and P. Dhariwal, “Glow: Generative Flow with Invertible 1x1 Convolutions,” *ArXiv180703039 Cs Stat*, Jul. 2018, Accessed: Dec. 24, 2020. [Online]. Available: http://arxiv.org/abs/1807.03039.

[36] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A Flow-based Generative Network for Speech Synthesis,” *ArXiv181100002 Cs Eess Stat,* Oct. 2018, Accessed: Dec. 28, 2020. [Online]. Available: http://arxiv.org/abs/1811.00002.

[37] D. Adiwardana *et al.*, “Towards a Human-like Open-Domain Chatbot,” *ArXiv200109977 Cs Stat*, Feb. 2020, Accessed: Mar. 21, 2020. [Online]. Available: http://arxiv.org/abs/2001.09977.

[38] constanza fierro, “Recipes for building an open-domain chatbot,” *Medium*, Jun. 05, 2020. https://medium.com/dair-ai/recipes-for-building-an-open-domain-chatbot-488e98f658a7 (accessed Dec. 31, 2020).

[39] S. R. Livingstone and F. A. Russo, “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English,” *PLOS ONE*, vol. 13, no. 5, p. e0196391, May 2018, doi: 10.1371/journal.pone.0196391.

[40] “Surrey Audio-Visual Expressed Emotion (SAVEE) Database.” http://kahlan.eps.surrey.ac.uk/savee/ (accessed Oct. 06, 2020).

[41] M. K. Pichora-Fuller and K. Dupuis, “Toronto emotional speech set (TESS).” Scholars Portal Dataverse, 2020, doi: 10.5683/SP2/E8H2MF.

[42] “CheyneyComputerScience/CREMA-D: Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D).” https://github.com/CheyneyComputerScience/CREMA-D (accessed Oct. 06, 2020).

[43] “Indic TTS.” https://www.iitm.ac.in/donlab/tts/index.php (accessed Jan. 07, 2021).

[44] A. Baby and A. L. Thomas, “Resources for Indian languages,” p. 8.

[45] K. Ito and L. Johnson, *The LJ Speech Dataset*. 2017.

[46] H. Zen *et al.*, “LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech,” *ArXiv190402882 Cs Eess*, Apr. 2019, Accessed: Jan. 06, 2021. [Online]. Available: http://arxiv.org/abs/1904.02882.

[47] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 5206–5210, doi: 10.1109/ICASSP.2015.7178964.

[48] D. S. Park *et al.*, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” *Interspeech 2019*, pp. 2613–2617, Sep. 2019, doi: 10.21437/Interspeech.2019-2680.

[49] I. Zenkov*, transformer-cnn-emotion-recognition*. GitHub, 2020.

[50] A. Hannun *et al.*, “Deep Speech: Scaling up end-to-end speech recognition,” *ArXiv14125567 Cs*, Dec. 2014, Accessed: Apr. 05, 2020. [Online]. Available: http://arxiv.org/abs/1412.5567.

[51] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, “Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis,” *ArXiv200505957 Cs Eess*, Jul. 2020, Accessed: Oct. 05, 2020. [Online]. Available: http://arxiv.org/abs/2005.05957.

[52] C. Szegedy *et al.*, “Going Deeper with Convolutions,” *ArXiv14094842 Cs*, Sep. 2014, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1409.4842.

[53] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[54] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” *ArXiv180308375 Cs Stat*, Feb. 2019, Accessed: Dec. 31, 2020. [Online]. Available: http://arxiv.org/abs/1803.08375.

[55] E. M. Smith, M. Williamson, K. Shuster, J. Weston, and Y.-L. Boureau, “Can You Put it All Together: Evaluating Conversational Agents’ Ability to Blend Skills,” *ArXiv200408449 Cs*, Apr. 2020, Accessed: Dec. 24, 2020. [Online]. Available: http://arxiv.org/abs/2004.08449.

[56] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The Curious Case of Neural Text Degeneration,” *ArXiv190409751 Cs*, Feb. 2020, Accessed: Dec. 30, 2020. [Online]. Available: http://arxiv.org/abs/1904.09751.

[57] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, “Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring,” *ArXiv190501969 Cs*, Mar. 2020, Accessed: Dec. 31, 2020. [Online]. Available: http://arxiv.org/abs/1905.01969.

[58] S. Roller *et al.*, “Recipes for building an open-domain chatbot,” *ArXiv200413637 Cs*, Apr. 2020, Accessed: Dec. 24, 2020. [Online]. Available: http://arxiv.org/abs/2004.13637.

[59] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, “Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis,” *ArXiv200505957 Cs Eess*, Jul. 2020, Accessed: Oct. 06, 2020. [Online]. Available: http://arxiv.org/abs/2005.05957.

[60] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Jan. 2017, Accessed: Dec. 24, 2020. [Online]. Available: http://arxiv.org/abs/1412.6980.

[61] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using Real NVP,” *ArXiv160508803 Cs Stat*, Feb. 2017, Accessed: Dec. 25, 2020. [Online]. Available: http://arxiv.org/abs/1605.08803.

[62] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improving Variational Inference with Inverse Autoregressive Flow,” *ArXiv160604934 Cs Stat*, Jan. 2017, Accessed: Dec. 24, 2020. [Online]. Available: http://arxiv.org/abs/1606.04934.

[63] M. Popović and H. Ney, “Word error rates: decomposition over Pos classes and applications for error analysis,” in *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT ’07*, Prague, Czech Republic, 2007, pp. 48–55, doi: 10.3115/1626355.1626362.

[64] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of Tricks for Image Classification with Convolutional Neural Networks,” *ArXiv181201187 Cs*, Dec. 2018, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1812.01187.

[65] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556.