Slurred Speech Recognition using Deep learning and Transfer learning

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Fourth Year B.Tech.

by

Kumar Shivam Ranjan
Madhav Bansal

Under the guidance of

Dr. Hari Prabhat Gupta

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
November 2021

Dedicated to

My parents, teachers,.....

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi	**Kumar Shivam Ranjan , Madhav Bansal**
Date: **20/11/2021**	B.Tech. or IDD Student
	Department of Computer Science and Engineering,
	Indian Institute of Technology (BHU) Varanasi,
	Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Slurred Speech Recognition using transfer learning** " being submitted by **Kumar Shivam Ranjan(18075031) and Madhav Bansal(18075036)** carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: **20/11/2021**

Dr. Hari Prabhat Gupta
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

I would like to express our sincere gratitude to our Teaching Assistant **Mr. Atul Chaudhary** and our supervisor **Dr.Hari Prabhat Gupta** who helped us whenever we needed them and our parents for their belief and support.

Place: IIT (BHU) Varanasi
Date: **20/11/2021** **Kumar Shivam Ranjan , Madhav Bansal**

Abstract

Deep learning has evolved a lot to develop end to end state of the art speech Recognition System. End to end deep learning model for automatic speech Recognition is much simpler and efficient than traditional speech systems. The traditional speech recognition systems requires manually labelled dataset and hand-engineered processing phases. Moreover, in case of noise prone environments, their model performance is not considerably efficient.

The introduction to end to end deep learning speech recognition system has changed a lot how speech is synthesised and converted to text transcripts. Deep learning End to End models don't need hand designed components to deal with background noise. Their performance is not affected by speaker variation. Moreover, the concept of phoneme is not that relevant in end to end speech recognition systems unlike traditional speech recognition systems where phenome plays a very important role. In recent times, bidirectional GRU based deep learning model or RNN-CTC based models have shown the results which has provided a new benchmark for end to end models for automatic speech recognition.

The well known Automatic Speech Recognition Systems used these days are amazon's Alexa, apple's Siri etc. Such types of speech recognition systems starts with the spoken audio file which contains human voice in a particular language and extracts the words that were spoken as text transcripts.

Contents

Chapter 1

Introduction

1.1 Overview

The conventional speech recognition system heavily depends on multiple pipelines consisting of several algorithms with hand-engineered processing stages. They tend to perform poorly with noisy environments and variation in speech volunteers. The introduction of end to end deep learning models have not only significantly improved the performance of speech recognition systems but also have decreased the training time and reduced the efforts for heavily engineered processing pipelines.
In this report we present our end to end deep learning model implementation that not only converts an audio signal into human spoken transcripts but also takes into account a special usecase for speech recognition.

1.2 Problem Statement

The state of the art Automatic Speech Recognition(ASR) can greatly improve the lives of those with speech impairments. However , the end to end deep learning Automatic Speech Recognition System trained from 'normal' speech tend to perform poorly for those who have speech impairments either due to accident or disease. The purpose of this project is to improve ASR for people who have speech impairments. In other words, we have to develop an ASR model that works on personalised non-standard speech.[1]

1.3 High level Solution overview

The Automatic Speech Recognition system available in present day starts with the audio data of human speech in a specific language and extracts the words that were spoken in that audio as text irrespective of the reverberation or speaker variation.

For such speech recognition problems, our training data consists of:

- X: Input features i.e audio clips of human speech
- Y: Target label or text transcript of what was spoken

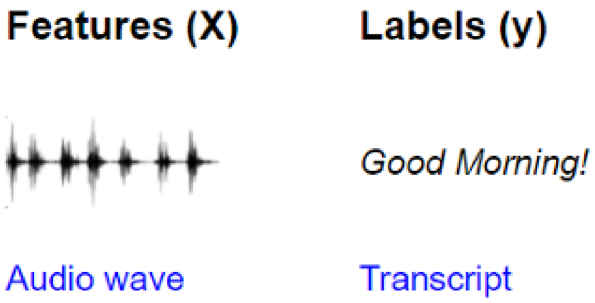


Figure 1.1 ASR uses audio waves as input and text transcript as target

The state-of-the-art ASR models present today can yield high word error rates and character error rates for speakers with speech impairments either due to accidents or disease.So, we cannot use the current end to end ASR models for such people. This is the reason that **Personalised ASR for atypical speech with limited data** is the need of the hour.

We first prepare or make use of already available ASR model with very high accuracy trained on thousands of hours of standard speech. we , then fine-tune different parts of the model especially the last few layers with personalised dataset of non-standard speech.[2] We can also make use of transfer learning technique to improve our model accuracy on atypical speech.

The major issue with slurred speech recognition was the difficulty in collecting enough data to train a state-of-the-art recognizer for individuals. ASR models are trained on hundreds of hours of speech dataset. Acquiring much dataset from a single volunteer is almost impossible.We overcome this issue by first training a base ASR end to end deep learning model on a large corpus of typical speech, and then training a personalized model using a much smaller dataset with the targeted non-standard speech characteristics. [3]

Chapter 2

Theory

2.1 Data pre-processing workflow

The very first phase for this project is to obtain high accuracy ASR end-to-end deep learning model. We will start by implementing the same. We'll be training on a subset of LibriSpeech Dataset, comprising 100 hours of transcribed audio data from different speakers.

For this problem, our training data consists of:

- Input features (X): audio waves of human speech
- Target labels (Y): a text transcript of what was spoken in the audio.

The main objective of our deep learning model is to take audio waves as input and learn to predict the text content or sentences that were spoken in the respective audio data.

Following is the workflow for data pre-processing:

- Load audio files
- Resampling
- Raw audio data augmentation
- Mel Spectograms generation
- Conversion to MFCC
- SpecAugmentation (Spectogram Data augmentation)

2.2 Loading of Audio data

We start by loading audio data in **.flac** format.We use torchaudio library of pytorch to process audio data. The loaded audio data is nothing but sequence of numbers representing amplitude of the audio at a particular instant of time. The size of such sequence is determined by the sampling rate of the audio.

2.3 Raw Data augmentation

Since deep learning requires large amount of labelled dataset, raw data augmentation is crucial in case we don't have much dataset. It also makes our dataset more diverse that will make our model generalised. It is done by shifting our audio data with respect to time either to left or to right by small amount. Changing the speed of the sound or making small variations in pitch of the audio is also part of raw data augmentation.

2.4 Spectrograms

Deep learning models rarely uses raw audio data as input feature. Instead spectrograms of the audio data is used to feed directly into our deep learning model. Spectrograms are the 2D image representation of the audio waves. It decomposes audio into set of frequencies that are present in it. Time is plotted on the X-axis and frequencies are present on the y-axis. The color at particular co-ordinate represents the intensity or amplitude of the audio at a particular instant of time. Brighter the color, more is the amplitude.

2.5 SpecAugmentation

We can also apply data augmentation techniques on spectrograms as well. This is called spectrogram augmentation or specAugmentation. In this augmentation technique, we randomly perform frequency masking which is also called horizontal masking because in spectrograms , frequencies are plotted on Y-axis. We can also perform time masking also called horizontal masking for similar reasons.

The specrtograms obtained at this stage can be directly fed into our deep learning model.

2.6 Model architecture

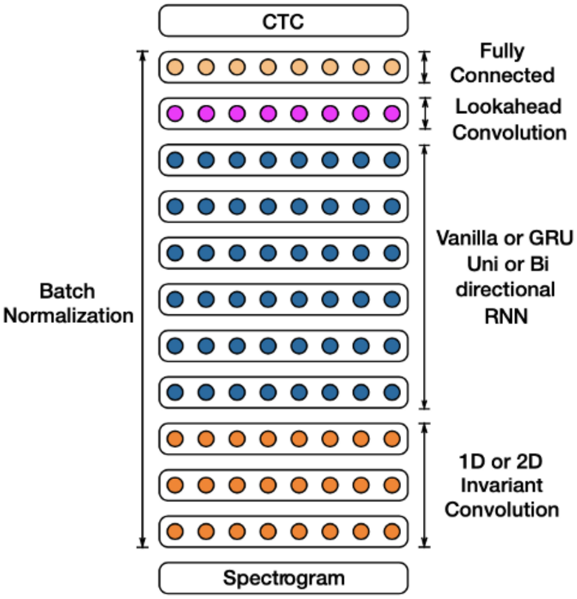

CTC
Fully Connected
Lookahead Convolution
Vanilla or GRU Uni or Bi directional RNN
Batch Normalization
1D or 2D Invariant Convolution
Spectrogram

Figure 2.1 ASR Model Architecture

The core of this architecture is the Convolutional neural network(CNN) and Recurrent neural network(RNN) precisely gated recurrent unit trained to ingest speech spectograms and generate text transcripts.
Suppose our training set consists of :

$$X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ...\}$$

where x is the single utterance and y is the text transcript.
Each data point $x^{(i)}$ is of length $T^{(i)}$ where every time-step is a vector of amplitudes.So i$^{\text{th}}$ data point can be represented as ${x^{(i)}}_t$ where t= 1, 2, ..$T^{(i)}$. Our audio features is the spectrograms that we got as data preprocessing result. The goal of our model is to update the weights of the CNN and RNN layers so as to be able to convert the input array x into character probabilities per time-step.

2.7 Model Workflow

Our Deep learning model architecture is built with CNN layers as well as RNN layers packed with Connectionist temporal classification loss algorithm to mark boundaries for each alphabets or characters of spoken words in the audio file.

A high level model understanding is made by:

- CNN layers composed of few Residual CNNs that take mel-spectograms as input and produce feature maps of the same.
- After CNN, we have few RNN(ranging from 5 to 7) layers , precisely some bidirectional GRU(gated recurrent unit) that takes those feature maps as produced by CNN layers as input and process them as a consecutive sequence of timesteps or frames. The feature maps which are continous representation of the audio waves are eventually converted into discrete representation.
- A linear layer is at the end that take output from the RNNs and produce character probabilities for each timestep or frame. It uses the softmax activation function.

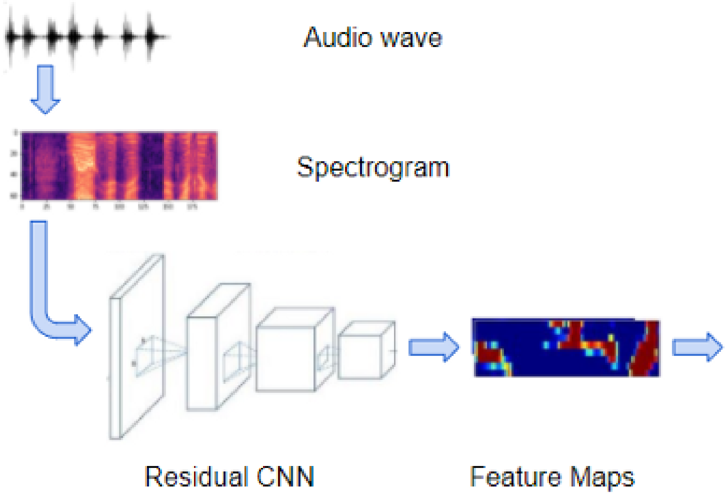


Figure 2.2 Spectrogram as input to CNN

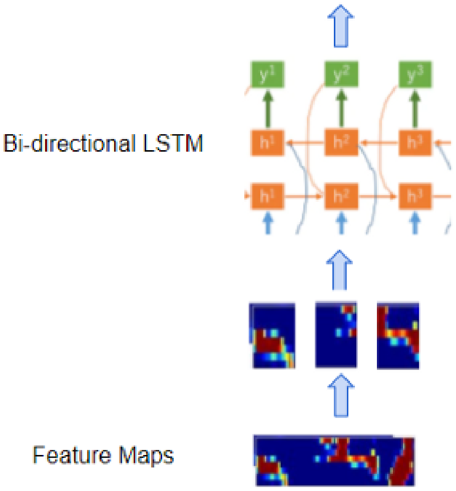
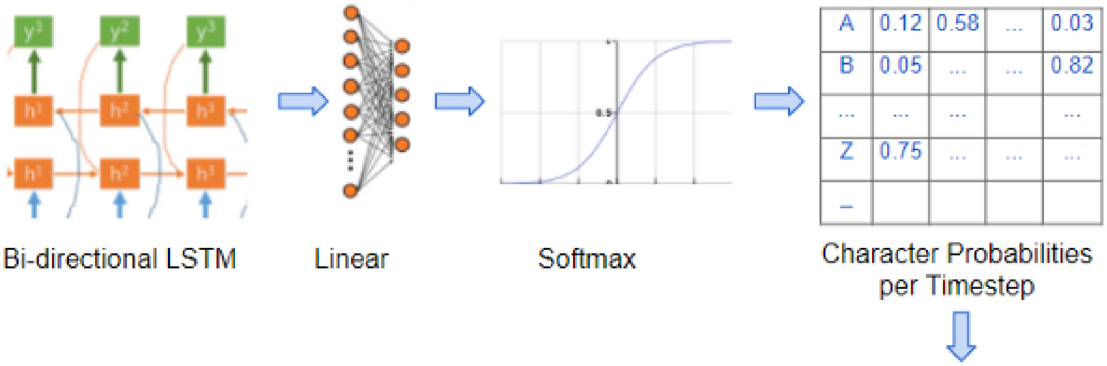


Figure 2.3 Feature maps as input to RNN



A	0.12	0.58	...	0.03
B	0.05	...	...	0.82
...	...	...		...
Z	0.75	...	...	...
–				

Figure 2.4 Character probabilities for each frame

2.8 Model Evaluation metric

Model Evaluation is very important phase of machine learning or deep learning pipeline.Once, we train our model, we must evaluate its performance. This phase is crucial when we have multiple trained models and we have to make a decision of which model should go into production. There are different metrics used for evaluating the performance of a machine learning or deep learning solution depending upon the usecase.

The most commonly used evaluation metric used in Automatic speech recognition systems is called Word error rate(WER) and character error rate(CER). They are used to compare the transcript predicted by our model to actual transcript and generates a number that represents how much difference there is in the predicted and the actual transcript.

There are three terms related to word error rate:

- Deletion: count of words that are present in the transcript but not in prediction
- Insertion:count of words that are present in the transcript but not in prediction
- Substitution: count of words that are altered between prediction and transcript

$$WER = \frac{Deletion + Insertion + Substitution}{total}$$

2.9 Connectionist Temporal classification

Connectionist Temporal classification is used to establish demarcation or boundaries between spoken characters. Since our input is contiguous sound waves and the output is discrete, there is a need to align our input to output because there is no clear boundaries between elements. It removes the hand-engineered processing stages to demarcate the input signals.It also removes need to manually provide the alignment as a part of labelled training set.
CTC algorithm takes the output of the RNN (character probabilities) and generate some valid character sequence. There is concept of ”blank” token introduced in CTC represented by ”-” to handle alignments and multiple repeated characters in the spoken audio. Character probabilities also include the the blank token probability.

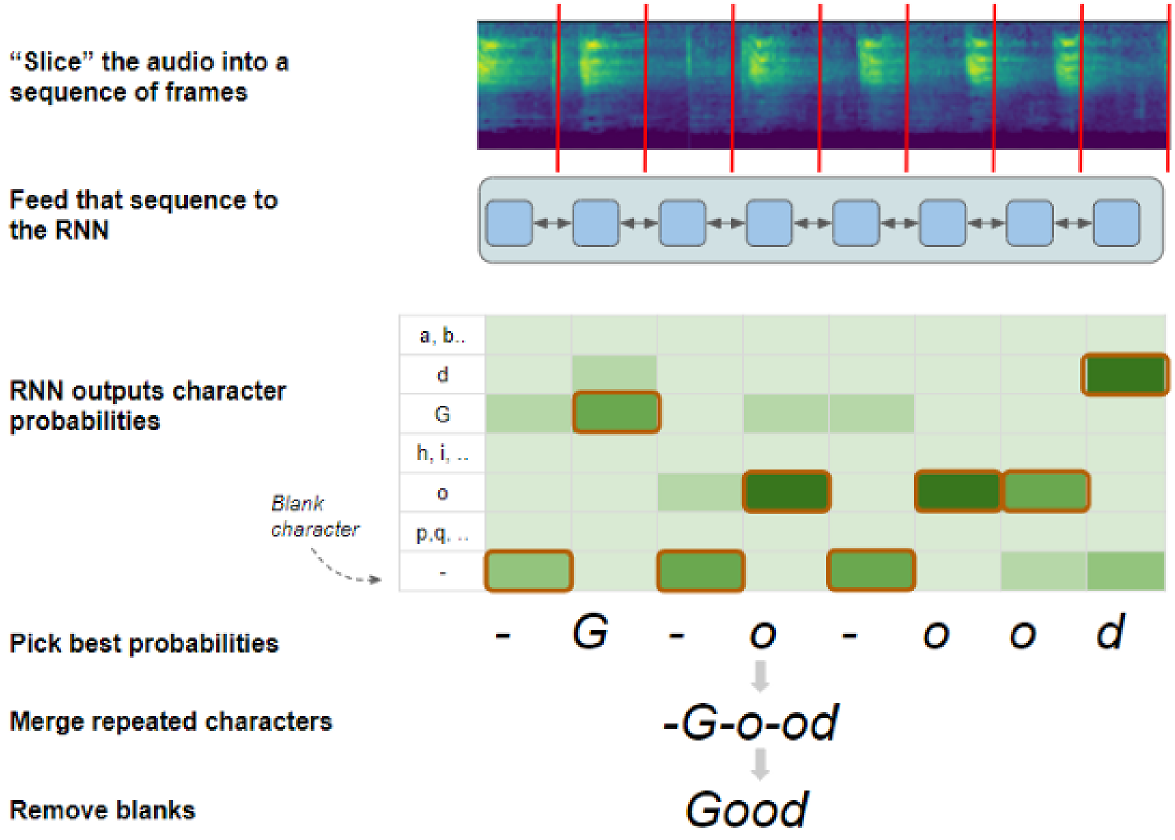

"Slice" the audio into a sequence of frames
Feed that sequence to the RNN
RNN outputs character probabilities
Blank character
a, b..
d
G
h, i, ..
o
p,q, ..
-
Pick best probabilities
- G - o - o o d
Merge repeated characters
-G-o-od
Remove blanks
Good

Figure 2.5 CTC Decoding

2.10 CTC Loss

CTC loss is calculated as probability of our model predicting the correct transcript. This algorithm first generates all valid character sequences. The algorithm follows the following paradigm:

- It keeps track of only those characters that occur in the target and removes the rest
- It selects only those alignments which follows the same order as the target transcript
- It computes the character probabilities for each time-step which in turn helps in calculating the probability of generating all valid character sequences.
- It is computed as negative logarithm probability of all possible valid character sequences.[4]

If we have 3 possible alignments , CTC will be calculating

$$P(y, alignment1|X), P(y, alignment2|X), P(y, alignment3|X)$$

where y is the label and X is the split spectrogram. After that, we would like to eliminate the alignment information to get only P(y |X).

This would mean the following:

$$P(y|X) = P(y, alignment1|X) + P(y, alignment2|X) + P(y, alignment3|X)$$

Loss is computed as:

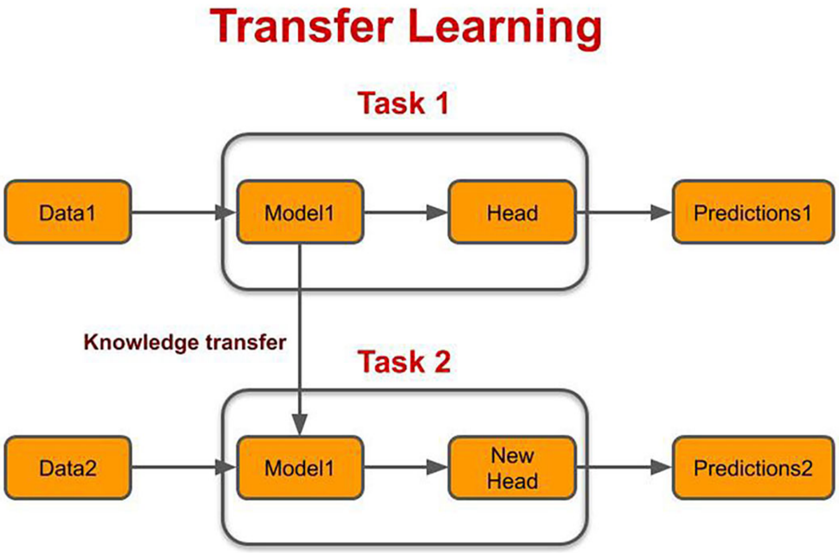
$$loss = -log(P(y|X))$$

- when P(y |X) aproaches 0, loss aproaches infinity
- when P(y |X) aproaches 1, loss aproaches 0
- We are training the model to learn and maximise P(y |X). When P(y |X) aproaches 1 , loss aproaches 0 , then we will not update weights of the model.

2.11 Transfer Learning

The most phase of our Deep learning pipeline is the technique called transfer learning.Conventionally, Transfer learning is a technique which is used to transfer weights from one model to another so that we don't have to build our model from scratch and can use the already available pre-trained model. [5]

Building a new model from scratch from the ground level and gathering new set of training data is very expensive and cumbersome.With the help of transfer Learning ,the need and effort to recollect the massive amounts of training data is reduced drastically. [6] In other words , in the context of Machine Learning and Deep Learning , people can intelligently apply knowledge learned previously for a different task or domain that can be used to solve new similar problems faster or with better solutions.



2.12 Size-similarity matrix

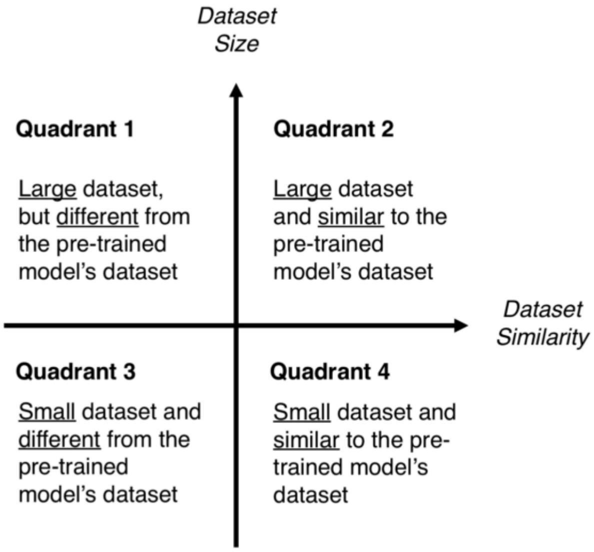

Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset

Figure 2.6 Dataset size-similarity matrix

The whole transfer learning workflow can be summarised as follows:

- **Quadrant 1**: Train a model from scratch. Even if the dataset is not similar, it can still be helpful to initialize our model from a pre-trained model precisely using its architecture and model parameters.
- **Quadrant 2**: Train only the classifier and the top layers or last few layers of the base model.
- **Quadrant 3**: Apply data augmentation techniques. Remove the previous classifier and add a new classifier and retrain all the layers of the model using pre-initialised weights from base model.
- **Quadrant 4**: Train only the classifier or retrain only the last few later freezing the rest of the layers.

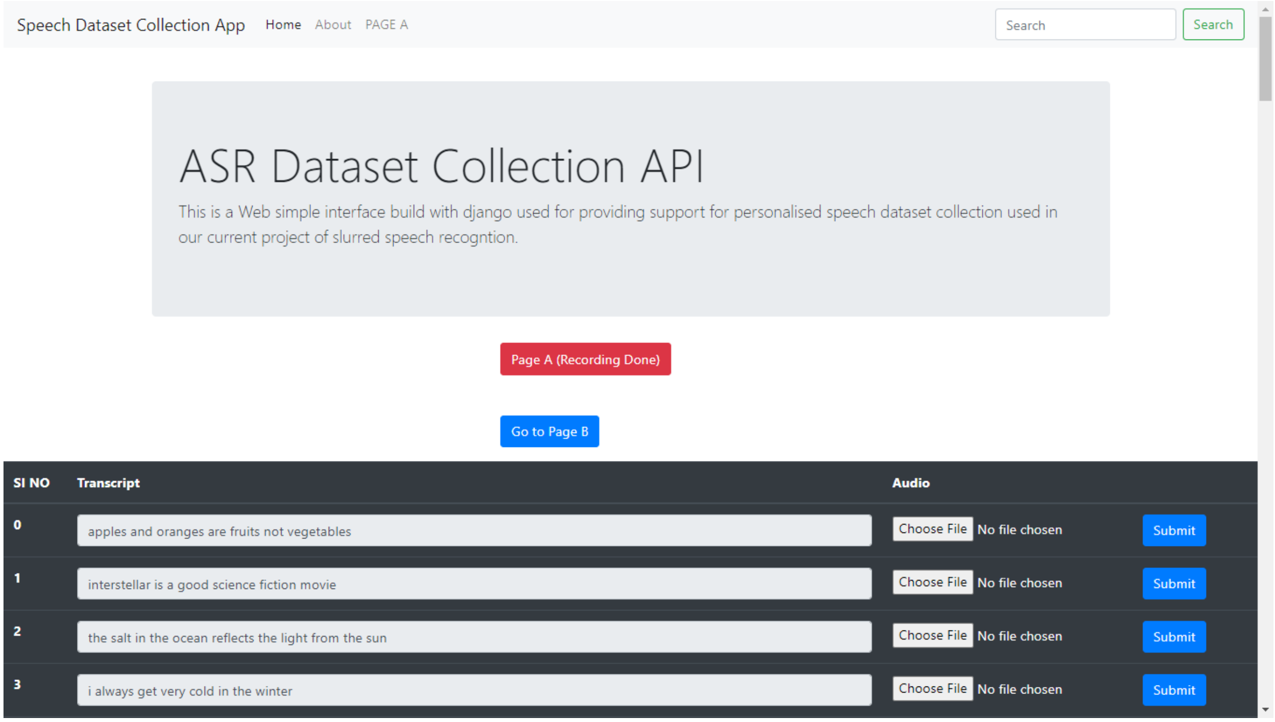
2.13 Collection of personalised atypical speech dataset

We first build our own Automatic Speech Recognition model trained on 100 hours of LibriSpeech Dataset. The results regarding Performance of our base model is mentioned in detail in the result section.

The next phase is collecting the Personalised atypical Speech dataset on which we have to apply transfer learning. The collection of personalised atypical dataset was facilitated by django based web application which uses default sqlite database to store audio recordings and corresponding transcripts. We built this web application solely for data collection purpose.

For generating transcripts, we used Google's ongoing research project called project Euphonia in which they circulated google form where user can record their voice against already available transcripts.

Below is the screenshot of web application we built for collecting and gathering dataset.



ASR Dataset Collection API

This is a Web simple interface build with django used for providing support for personalised speech dataset collection used in our current project of slurred speech recogntion.

SI NO	Transcript	Audio	
0	apples and oranges are fruits not vegetables	Choose File No file chosen	Submit
1	interstellar is a good science fiction movie	Choose File No file chosen	Submit
2	the salt in the ocean reflects the light from the sun	Choose File No file chosen	Submit
3	i always get very cold in the winter	Choose File No file chosen	Submit

2.14 Application of transfer learning in our usecase

After obtaining our base Automatic speech recognition model, we did furthur retraining of our model on our own personalised speech dataset in the following ways:

- **Fine-tuning**: We retrained all the layers with the weights pre-initialised from the base model.
- **Transfer Learning**: We retrained the classifier and last few layers freezing the rest of the layers.

Once we have applied transfer learning and fine-tuned our base model on personalised speech dataset, we can predict the text transcript against the audio file. Once we recieve the predicted text, we can use Google's API for speech to text translation. gTTS (Google Text-to-Speech)is a Python library and command-line interface tool.It is available in gTTS module of python.

Chapter 3

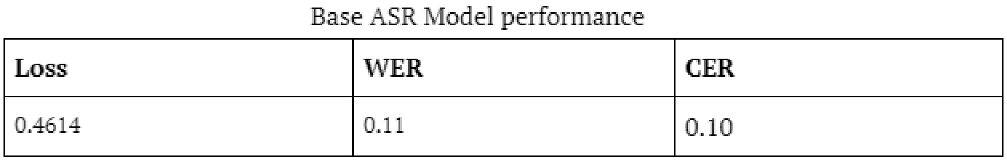
Results

3.1 Libraries/APIs used

- Pytorch
- Torchaudio
- Numpy
- Google gTTS

3.2 Base Model performance

We start with high accuracy automatic speech recognition trained on 100 hours of typical speech.
Performance of base model is listed as below:

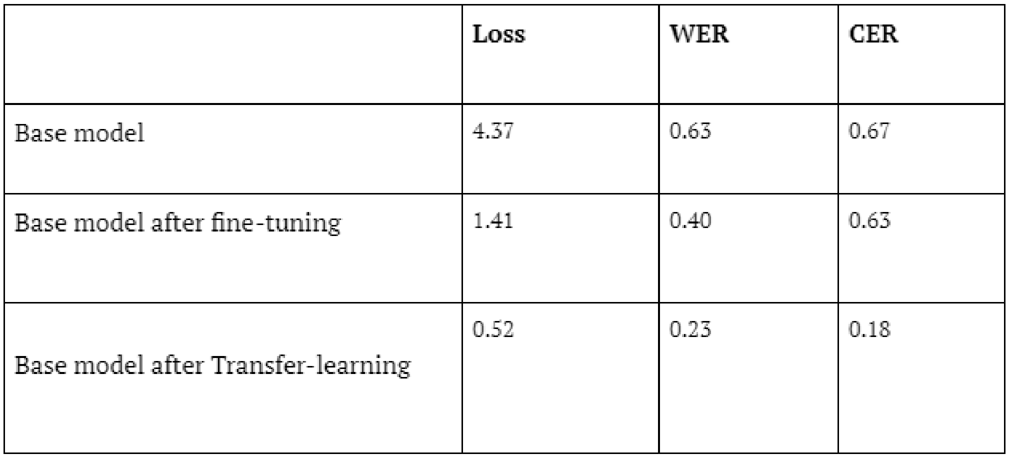


Base ASR Model performance

Loss	WER	CER
0.4614	0.11	0.10

Figure 3.1 Base Model performance metrics

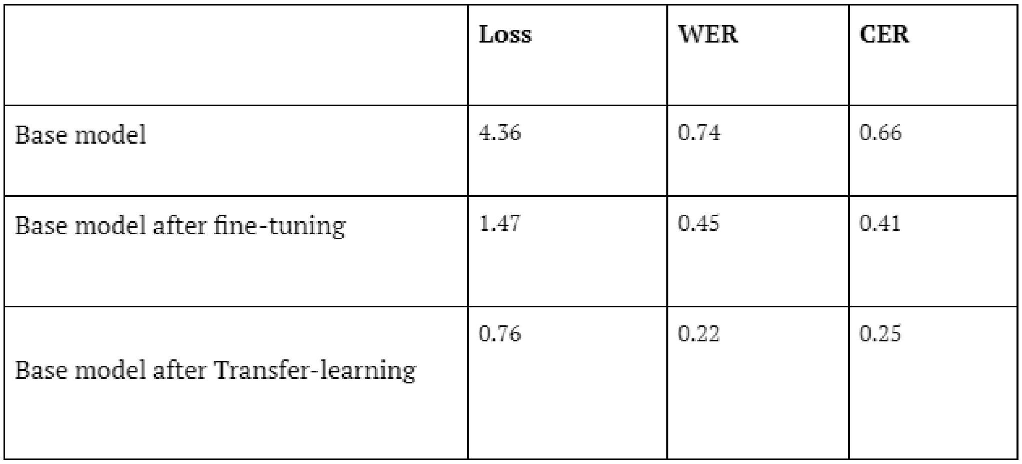
3.3 Transfer learning results(English)



	Loss	WER	CER
Base model	4.37	0.63	0.67
Base model after fine-tuning	1.41	0.40	0.63
Base model after Transfer-learning	0.52	0.23	0.18

Figure 3.2 Performance chart on slurred speech dataset

3.4 Transfer learning results(Hindi)



	Loss	WER	CER
Base model	4.36	0.74	0.66
Base model after fine-tuning	1.47	0.45	0.41
Base model after Transfer-learning	0.76	0.22	0.25

Figure 3.3 Performance chart on slurred speech dataset

3.5 Charts(English)

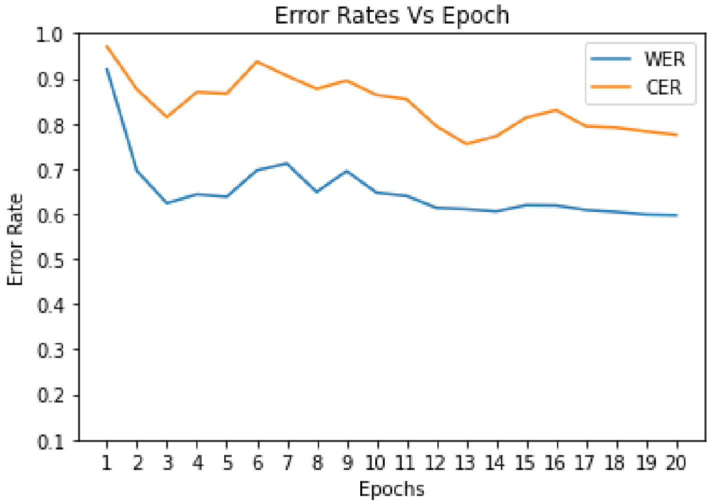


Figure 3.4 Error rates vs epochs after fine-tuning the model

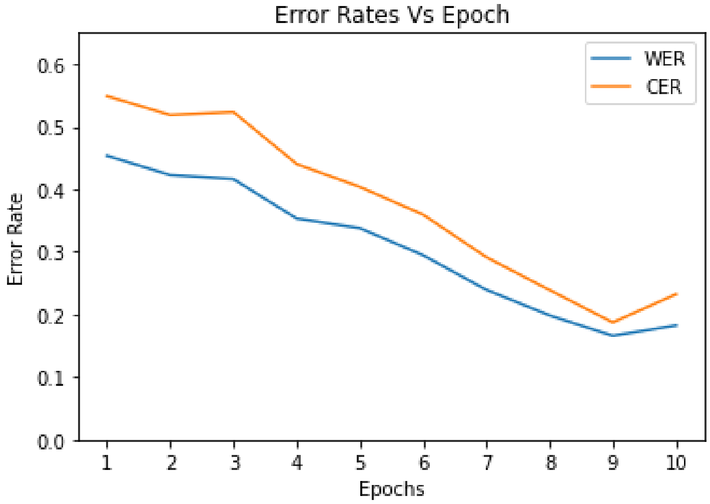


Figure 3.5 Error rates vs epochs after transfer learning

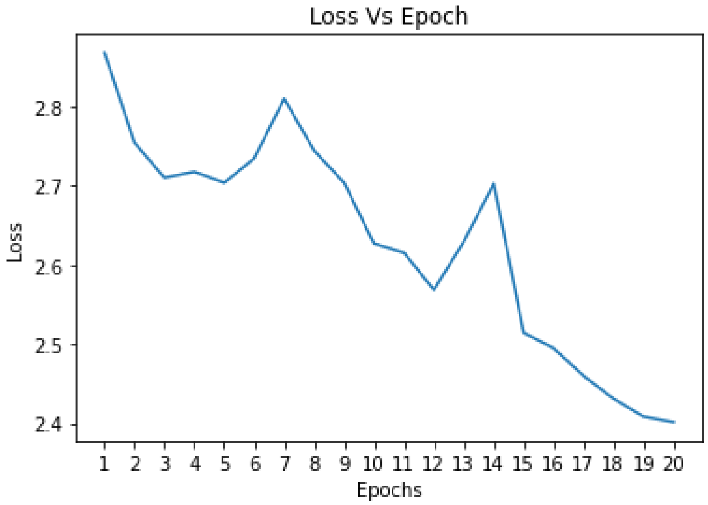


Figure 3.6 Loss Vs Epoch after fine-tuning

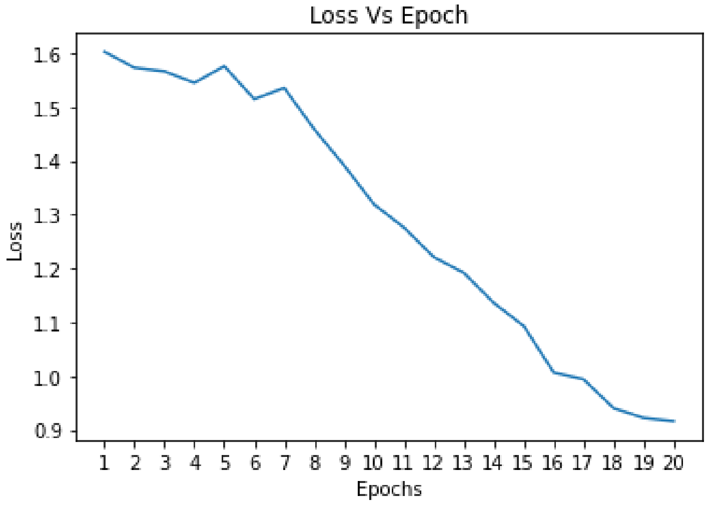


Figure 3.7 Loss vs Epoch after transfer Learning

3.6 Charts(Hindi)

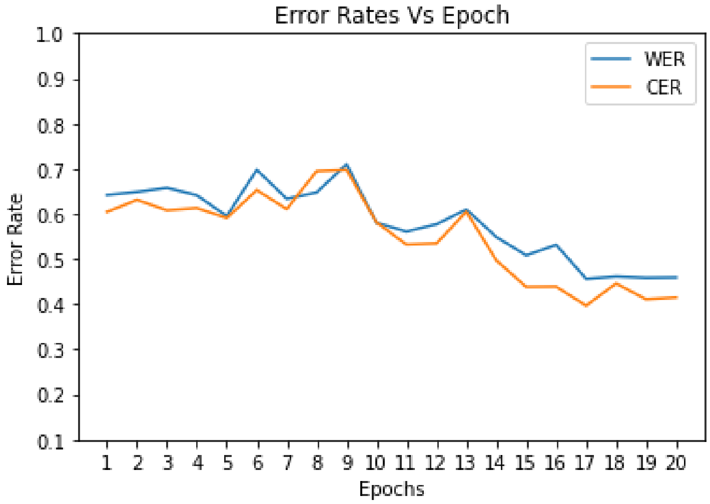


Figure 3.8 Error rates vs epochs after fine-tuning the model

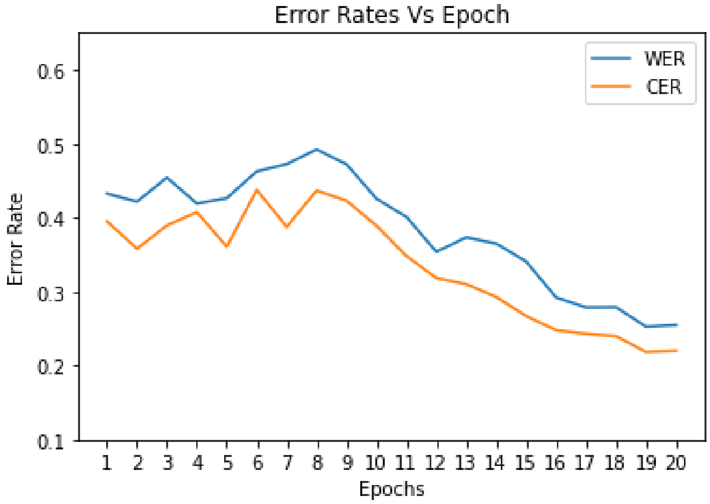


Figure 3.9 Error rates vs epochs after transfer learning

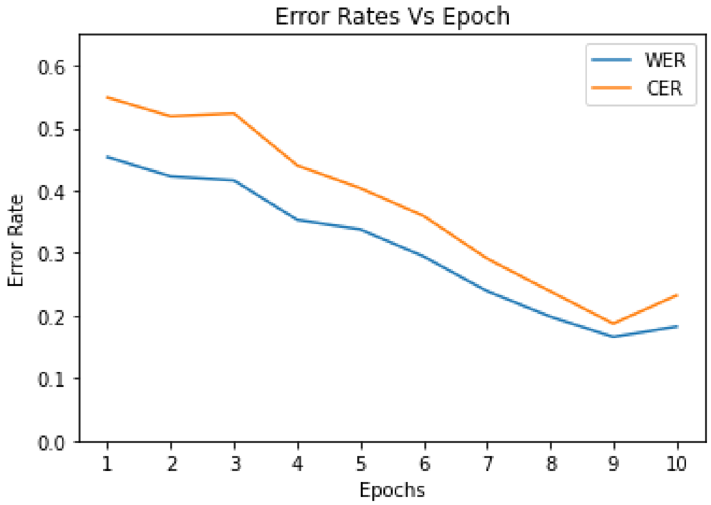


Figure 3.10 Loss Vs Epoch after fine-tuning

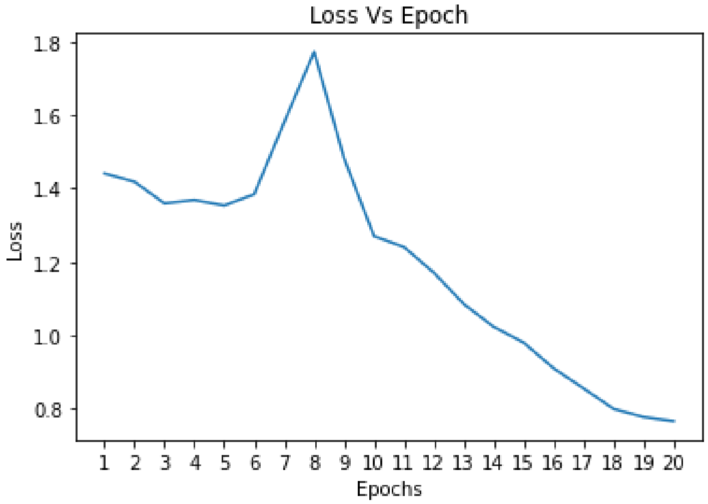


Figure 3.11 Loss vs Epoch after transfer Learning

Chapter 4

Conclusions and Discussion

4.1 Conclusion

In this project, we fine tuned the standard automatic speech recognition models to make it adapt to non-standard speech. We achieved considerable improvement by applying transfer learning. This approach is not limited to single user with non-standard speech but can be extended with any type of speech impairments. Our solution tend to perform good even in noisy environments. The solution can be extended to other languages as well such as Hindi. We can also see that after transfer learning , there is considerable improvement in performance of our model as compared to just fine-tuning.

4.2 Limitations

- The solution we provided in this paper is personalised and will vary when the type of non-standard speech varies.
- No such experimental results for non-standard automatic speech recognition has been published so far. So it's difficult to compare our solutions with industry standards.
- Deep learning demands huge labelled training dataset. Since this solution is personalised, it is difficult to collect large amount of data from single volunteer.

4.3 Future Scope

In the near future, we can try to make the solution feasible for more number of atypical speech types. We can also concentrate our focus to those techniques that can provide meaningful insights for low data regime. We can mark the phenome mistakes done by fine-tuned model. These mistakes can be used to weight certain examples during training phase. This will help user with non standard speech to record training examples that contain most common phenome mistakes.

Bibliography

[1] J. Cattiau, “How ai can improve products for people with impaired speech,” *Online at https://www. blog. google/outreach-initiatives/accessibility/impaired-speech-recognition*, 2019.

[2] L. Clark, B. R. Cowan, A. Roper, S. Lindsay, and O. Sheers, “Speech diversity and speech interfaces: Considering an inclusive future through stammering,” in *Proceedings of the 2nd Conference on Conversational User Interfaces*, 2020, pp. 1–3.

[3] S. M. Sekhar, G. Kashyap, A. Bhansali, K. Singh *et al.*, “Dysarthric-speech detection using transfer learning with convolutional neural networks,” *ICT Express*, 2021.

[4] J. Lee and S. Watanabe, “Intermediate loss regularization for ctc-based speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6224–6228.

[5] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[6] W. Ying, Y. Zhang, J. Huang, and Q. Yang, “Transfer learning via learning to transfer,” in *International conference on machine learning*. PMLR, 2018, pp. 5085–5094.