

SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY

MARALURU, TUMAKURU-572103

(A Constituent college of Sri Siddhartha Academy of Higher Education, Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Mini Project Report

On

“AUTOMATIC SPEECH RECOGNITION USING DEEP LEARNING”

Submitted in partial fulfilment of the requirement for the completion of V semester of

BACHELOR OF ENGINEERING

Submitted by:

NEHA ACHARYA

20CS051

PRATHUASHA K B

20CS058

Under the guidance of:

Dr. Channakrishnaraju

Professor

Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2022-23

SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY

MARALURU, TUMAKURU-572103

(A Constituent college of Sri Siddhartha Academy of Higher Education, Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that mini project work entitled “**AUTOMATIC SPEECH RECOGNITION
USING DEEP LEARNING**”, is a bonafide work carried out by

NEHA ACHARYA (20CS051)

PRATHUASHA K B (20CS058)

The report has been approved as it satisfies the academic requirements with respect to Mini-
Project work (CS5MP1) prescribed for the course.

.....
Dr. Channakrishnaraju
Professor
Mini-Project Guide

.....
Dr. M Siddappa
Head of the Department

DECLARATION

We, **NEHA ACHARYA (20CS051)** and **PRATHUASHA K B (20CS058)** of fifth semester, Department of Computer Science and Engineering of Sri Siddhartha Institute of Technology, Tumakuru, hereby declare that this Mini project titled, “**Automatic Speech Recognition using Deep Learning**”, has been carried out by us under the supervision of Dr. Channakrishnaraju, Professor, Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumakuru in partial fulfilment of the requirement for the completion of V semester in Computer Science and Engineering.

Date:

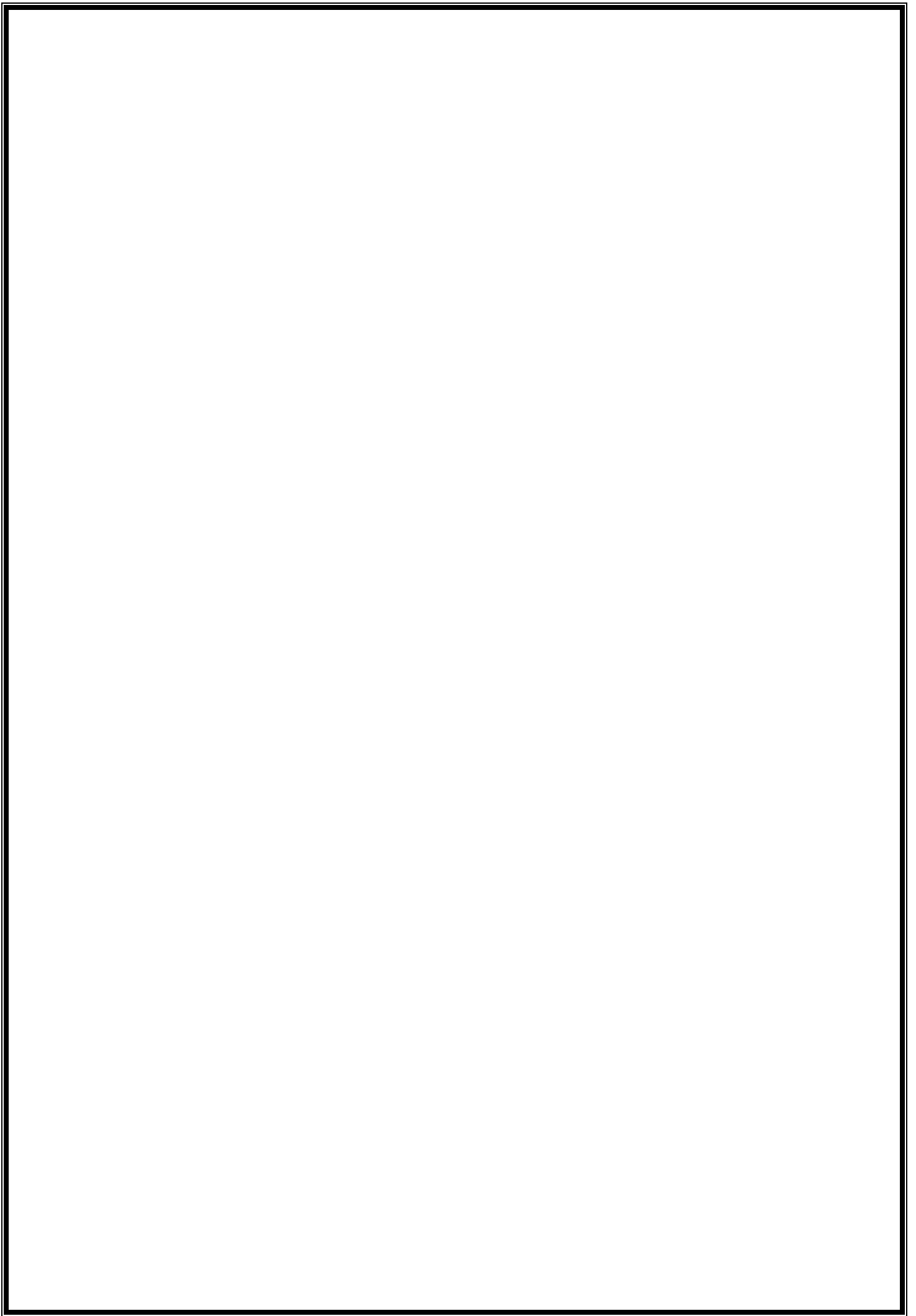
NEHA ACHARYA (20CS051)

Place: Tumakuru

PRATHUASHA K B (20CS058)

INDEX

CONTENTS	PG.NO
Abstract	1
List of Figures	2
List of Abbreviations	3
CHAPTER 1: INTRODUCTION	4-6
1.1 Project Introduction	
1.2 Problem Statement	
1.3 Aims and Objectives	
CHAPTER 2: SURVEY	7-8
2.1 Literature Survey	
CHAPTER 3: SYSTEM REQUIREMENTS AND SPECIFICATION	
3.1 Software Requirements	9-11
3.1.1 Deep Learning	
3.1.2 Hugging Face	
3.1.3 Wav2Vec 2.0 Model	
3.1.4 Python Libraries	
3.2 Hardware Requirements	11
CHAPTER 4: SYSTEM DESIGN	12-15
4.1 System Architecture	
4.2 Implementation	
CHAPTER 5: RESULTS	16-18
CHAPTER 6: CONCLUSION	19
REFERENCES	20



ABSTRACT

Presently, computers have already replaced a tremendous number of humans in many creative professions. Therefore, Artificial Intelligence areas are composed of Machine Learning, Deep learning, Natural Language Processing, Computer Vision, and Robotics. Similarly, speech recognition can be predicted by using computers. Humans communicate preferably through speech using the same language. Speech recognition can be defined as the ability to understand the spoken words of the person speaking. The use of a speech recognition model has become extremely important as Speech Control has become an important type. However, Automatic Speech Recognition, virtual assistants like Apple's Siri or Amazon's Alexa or Google Assistant has reached almost human performance in some controlled scenarios, to find out quick answers on the web or to simply command something. These AI assistants are well known for understanding our speech commands and performing the desired tasks.

Automatic Speech Recognition is a technique that processes human speech into readable text. ASR systems are also known as speech-to-text or transcription systems. This field has grown exponentially over the past decade, the ASR systems are used in a wide range of applications extending from finance to healthcare. It is an important research area for human-to-machine communication.

During Speech Recognition, misspelled or misused words can create problems for text analysis. Autocorrect and grammar correction applications can handle common mistakes, but don't always understand the writer's intention. With spoken language, mispronunciations, different accents, stutters, etc., can be difficult for a machine to understand.

However, as language databases grow and smart assistants are trained by their individual users, these issues can be minimized. The more data NLP models are trained on, the smarter they become. Hence, we use deep learning to allow a multitude of NLP techniques, algorithms, and models to work progressively, much like the human mind does. We now know that speech recognition tasks require huge amounts of data, commonly hundreds of hours of labelled speech.

Given the popularity and demand of this technology, here is an attempt to create a simple speech recognition system that takes our voice as input and produces the corresponding text by hearing the input by using Wav2Vec2.0— a-state-of-the-art speech recognition approach by Facebook and Hugging Face model hub.

LIST OF FIGURES

Figure 1 : Overview of Deep Learning	Error! Bookmark not defined.
Figure 2 : Hugging Face's user interface	10
Figure 3 : Illustration of Wav2Vec 2.0 framework.....	11
Figure 4 : Wav2Vec2, simplified architecture.....	12
Figure 5 : Hugging Face's user interface	16
Figure 6 : Creating Hugging Face account and Setting up a new space.....	17
Figure 7 : Dashboard of new Hugging Face account.....	17
Figure 8 : Spaces/Files and versions	18
Figure 9 : Output interface created spaces	18

LIST OF ABBREVIATIONS

ASR	Automatic Speech recognition
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
NLP	Natural Language Processing
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
HMM	Hidden Markov Model
GMM	Gaussian Mixture Model
OS	Operating System
CTC	Connectionist Temporal Classification
WER	Word Error Rate
NLTK	Natural Language Toolkit
LSTM	Long Short-Term Memory network
API	Application Programming Interface
NLU	Natural Language Understanding
NLG	Natural Language Generation

Chapter-1: Introduction

1.1 PROJECT INTRODUCTION

Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text. Rudimentary speech recognition software has a limited vocabulary and may only identify words and phrases when spoken clearly. Speech recognition uses a broad array of research in computer science, linguistics, and computer engineering. Many modern devices and text-focused programs have speech recognition functions in them to allow for easier or hands-free use of a device.

Automatic Speech Recognition (ASR) is an important technology to enable and improve the human-human and human-computer interactions. This technology allows users of information systems to speak entries rather than punching numbers on a keyboard. ASR is used primarily to provide information and to forward telephone calls. Sophisticated ASR systems allow the user to enter direct queries or responses, such as a request for driving directions or the telephone number of a hotel in a particular town. This shortens the menu navigation process by reducing the number of decision points. It also reduces the number of instructions that the user must receive and comprehend. Applications such as voice-controlled assistants like Alexa and Siri, and voice-to-text applications like automatic subtitling for videos and transcribing meetings, are all powered by this technology. These applications take audio clips as input and convert speech signals to text, also referred as speech-to-text applications.

For data scientists who want to build their own ASR models, many of the latest models can achieve a good performance, such as transformer-based models Wav2Vec2 and Speech2Text. Transformer is a sequence-to-sequence deep learning architecture originally proposed for machine translation. Now it's extended to solve all kinds of natural language processing (NLP) tasks, such as text classification, text summarization, and ASR. The transformer architecture yields very good model performance and results in various NLP tasks; however, the models' sizes (the number of parameters) as well as the amount of data they're pre-trained on increase exponentially when pursuing better performance.

Moreover, as the ASR quickly approaches human accuracy levels, there will be an explosion of applications incorporating ASR technology into their products. Given the popularity and demand of this technology, here is an attempt to create a system for ASR exclusively using Deep Learning approach.

1.2 PROBLEM STATEMENT

One of the main challenges of ASR today is the continual push toward human accuracy levels. While both ASR approaches--traditional hybrid and end-to-end Deep Learning--are significantly more accurate than ever before, neither can claim 100% human accuracy. This is because there is so much nuance in the way we speak, from dialects to slang to pitch. Even the best Deep Learning models can't be trained to cover this long tail of edge cases without significant effort.

Some think they can solve this accuracy problem with custom Speech-to-Text models. However, unless you have a very specific use case, like children's speech, custom models are actually less accurate, harder to train, and more expensive in practice than a good end-to-end Deep Learning model. In regards to model building, we also expect to see a shift to a self-supervised learning system to solve some of the challenges with accuracy discussed above.

End-to-end Deep Learning models are data hungry. An ASR model is trained on 100,000 hours of raw audio and video training data for industry-best accuracy levels. However, obtaining human transcriptions for this same training data would be almost impossible given the time constraints associated with human processing speeds.

This is where self-supervised deep learning systems can help. Essentially, this is a way to get an abundance of unlabelled data and build a foundational model on top of it. Then, since we have statistical knowledge of the data, we can fine-tune it on downstream tasks with a smaller amount of data, making it a more accessible approach to model building. This is an exciting possibility with profound implications on the field.

1.3 AIMS AND OBJECTIVES

The main objective of the project on Automatic Speech Recognition is to allow machines to recognize sounds and act on them using Deep learning approach. It allows machines to attain the ability to identify “receive and interpret” speech and translate it into readable form or text (sequence of sound waves into a string of letters or words).

Transcription of human expression into spoken words is the target of Automated Speech Recognition (ASR). Due to different speaker characteristics, different voice patterns, uncertain ambient sounds, and so on, it is a very difficult activity because human speech signals are highly variable. Also, ASR requires the translation of variable-length speech signals into sequences of words or phonetic symbols of variable length.

Deep learning (DL) is a part of ML. Unlike ML, DL networks are not directly used to retrieve and classify functions. Without engaging the outside observer, the hidden layers of the deep learning network do all these indirectly by itself. DL algorithms have almost all been used to further develop computer skills to understand what people can do, including voice recognition. In particular, voice, being the primary medium of contact between human beings, has gained a great deal of attention from the introduction of artificial intelligence over the past five decades.

Chapter-2: Survey

2.1 LITERATURE SURVEY

Today, there are two main approaches to Automatic Speech Recognition: a traditional hybrid approach and an end-to-end Deep Learning approach.

- **TRADITIONAL HYBRID APPROACH**

The traditional hybrid approach is the legacy approach to Speech Recognition and has dominated the field for the past fifteen years. Traditional HMM (Hidden Markov Models) and GMM (Gaussian Mixture Models) require forced aligned data. Force alignment is the process of taking the text transcription of an audio speech segment and determining where in time particular words occur in the speech segment. This approach combines a lexicon model + an acoustic model + a language model to make transcription predictions.

Downsides of Using the Traditional Hybrid Approach

Though still widely used, the traditional hybrid approach to Speech Recognition does have a few drawbacks. Lower accuracy is the biggest. In addition, each model must be trained independently, making them time and labour intensive. Forced aligned data is also difficult to come by and a significant amount of human labour is needed, making them less accessible. Finally, experts are needed to build a custom phonetic set in order to boost the model's accuracy.

- **END-TO-END DEEP LEARNING APPROACH**

An end-to-end Deep Learning approach is a newer way of thinking about ASR. With an end-to-end system, you can directly map a sequence of input acoustic features into a sequence of words. The data does not need to be force-aligned. Depending on the architecture, a Deep Learning system can be trained to produce accurate transcripts without a lexicon model and language model, although language models can help produce more accurate results.

Advantages of End-to-End Deep Learning Models

End-to-end Deep Learning models are easier to train and require less human labour than a traditional approach. They are also more accurate than the traditional models being used today. The Deep Learning research community is actively searching for ways to constantly improve these models using the latest research as well, so there's no concern of accuracy plateaus anytime soon--in fact, we'll see Deep Learning models reach human level accuracy in the next few years.

Papers Referred

New Types of Deep Neural Network Learning for Speech Recognition and Related Applications (2013) describes the historical context in which acoustic models based on deep neural networks have been developed. It also describes the five ways of improving deep learning methods: (1) better optimization; (2) better types of neural activation function and better network architectures; (3) better ways to determine the myriad hyper-parameters of deep neural networks; (4) more appropriate ways to pre-process speech for deep neural networks; and (5) ways of leveraging multiple languages or dialects that are more easily achieved with deep neural networks than with Gaussian mixture models.

Speech Recognition using Deep Learning (2019) explains how audio files or video files that are large and have many minutes in length, was converted into speech. They did so by using Deep Learning. They had used Google corpus to train the model and received 66.22% of accuracy in the obtained text.

Deep Learning Convolutional Neural Network for Speech Recognition: A Review (2021) describes how the approaches based on deep learning are showing rather interesting outcomes in several applications including speech recognition, and how it attracts a lot of researches and studies. It also describes how developments occurred in the Speech Recognition field and also explains the current researches that are being carried on lately.

Speech to Text with Wav2Vec 2.0 (2021) explains how the pre-trained model Wav2Vec 2.0 by Facebook can be used to convert speech to text.

Automatic Speech Recognition Using Wav2Vec2 (2022) explains how to create a web interface for the Wav2Vec2 model using Gradio Python package and deploy it on Hugging Face Spaces.

Chapter-3: System Requirements and Specifications

3.1 SOFTWARE REQUIREMENTS

3.1.1 DEEP LEARNING

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before. Most deep learning methods use **neural network** architectures, which is why deep learning models are often referred to as **deep neural networks**. Deep learning models are trained by using large sets of labelled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

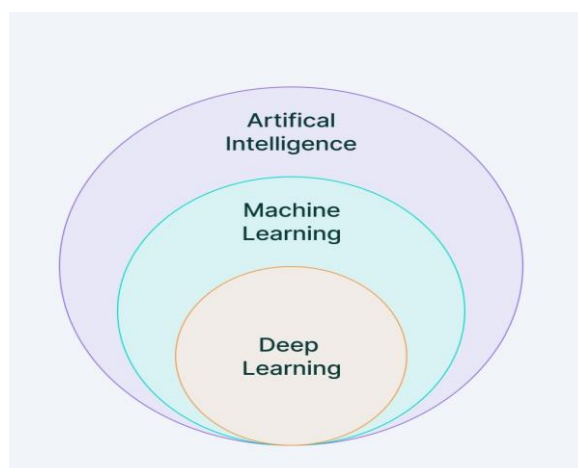


Figure 1: Deep Learning

3.1.2 HUGGING FACE

Hugging Face is a community and data science platform that provides: Tools that enable users to build, train and deploy ML models based on open source (OS) code and technologies. A place where a broad community of data scientists, researchers, and ML engineers can come together and share ideas, get support and contribute to open-source projects.

Hugging Face addresses this need by providing a community 'Hub'. It's a central place where anyone can share and explore models and datasets. They want to become a place with the largest collection of models and datasets with the goal of democratising AI for all.

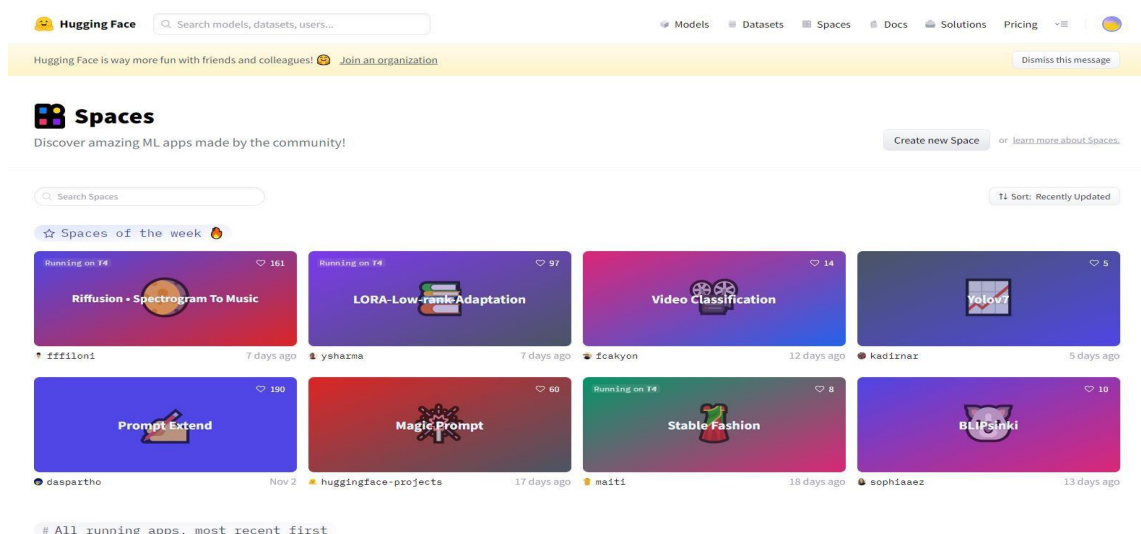


Figure 2: Hugging Face's user interface

3.1.3 WAV2VEC 2.0 MODEL

The Wav2Vec2 model was proposed in wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations by Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli. Wav2Vec2 is a speech model that accepts a float array corresponding to the raw waveform of the speech signal. It masks the speech input in the latent space and solves a contrastive task defined over a quantization of the jointly learned latent representations. Wav2Vec2 model was trained using connectionist temporal classification (CTC) so the model output has to be decoded using Wav2Vec2CTCTokeniser. It also attains 4.8/8.2 WER by pre-training the model on 53k hours of unlabelled data and fine-tuning on only ten minutes of labelled data. This shows that speech recognition can work with limited labelled data. Which can play a key role in devising ASR solutions for indigenous languages and dialects for which it's a little onerous to gather data.

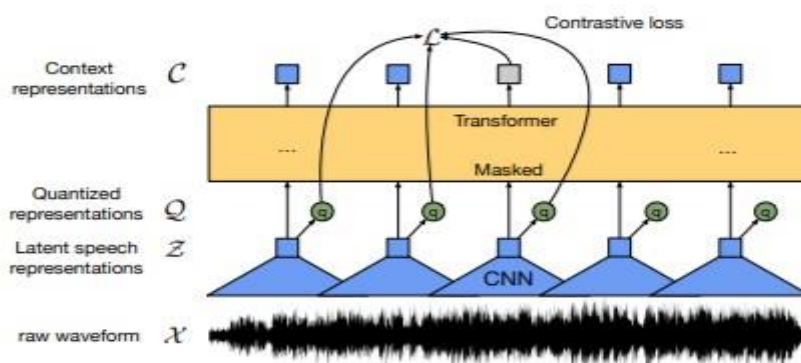


Figure 3: Illustration of Wav2Vec 2.0 framework

3.1.4 PYTHON LIBRARIES

NLTK

NLTK is a leading platform for building Python programs to work with human language data.

Installation: `>pip install nltk`

LIBROSA

LIBROSA is a python package for music and audio analysis.

Installation: `>pip install librosa`

PYTORCH

PyTorch is an open-source machine learning (ML) framework based on the Python programming language and the Torch library.

Installation: `>pip install torch`

GRADIO

Gradio allows you to build demos and share them, all in Python.

Installation: `>pip install gradio`

TRANSFORMERS

Transformers in Python can be used to clean, reduce, expand, or generate features.

Installation: `>pip install transformers`

It includes 2 packages, as follows:

- Wav2Vec2ForCTC is used to instantiate a Wav2Vec2 model according to the specified arguments, defining the model architecture.
- Wav2Vec2Tokenizer is used to prepare the array into input_values, and the Wav2Vec2Tokenizer should be used for padding and conversion into a tensor of type torch.

3.2 HARDWARE REQUIREMENTS

- Processor : intel core-i5
- Hard disk : 10GB (minimum)

- RAM : 1GB

Chapter-4: System Design

4.1 SYSTEM ARCHITECTURE

Wav2Vec2 is a transformer-based architecture for ASR tasks. The following diagram shows its simplified architecture.

As the diagram shows, the model is composed of a multi-layer convolutional network (CNN) as a feature extractor, which takes an input audio signal and outputs audio representations, also considered as features. They are fed into a transformer network to generate contextualized representations. This part of training can be self-supervised; the transformer can be trained with unlabelled speech and learn from it. Then the model is fine-tuned on labelled data with the Connectionist Temporal Classification (CTC) algorithm for specific ASR tasks. The base model we use in this post is [Wav2Vec2-Base-960h](#), fine-tuned on 960 hours of Librispeech on 16 kHz sampled speech audio.

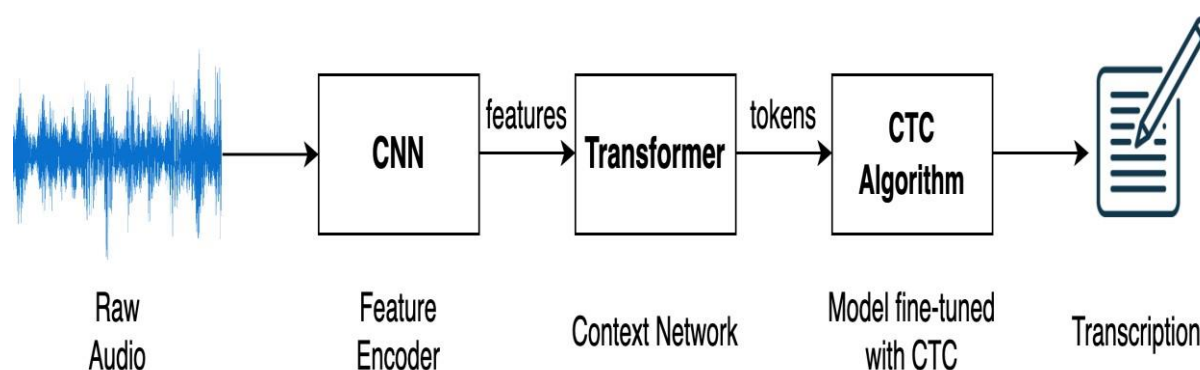


Figure 4: Wav2Vec2, simplified architecture

4.2 IMPLEMENTATION

- **Creating a Hugging Face account and setting up a New Space**

Create a Hugging Face account by visiting the official website of Hugging Face. After creating an account, go to the top-right side of the page and click on the profile icon, and then the 'New Space' button, which allows to create a new space. This will then direct to a new page to collect the details like name of repository that is to be created. Give the space a name, and then choose 'Gradio' from the SDK options before clicking the 'create new space' button. As a result, the repository for one's app will be created.

- **Creating a Requirements.txt File**

Create a requirements.txt file to list the Python packages like nltk, transformers, torch and librosa for the app to run successfully. Those dependencies will be installed with the help of pip install -r requirements.txt.

- **Creating app.py File**

#Importing all the necessary packages

```
import nltk
import librosa
import torch
import gradio as gr
from transformers import Wav2Vec2Tokenizer, Wav2Vec2ForCTC
nltk.download("punkt")
```

#Loading the pre-trained model and the tokenizer

```
model_name="facebook/wav2vec2-base-960h"
tokenizer=Wav2Vec2Tokenizer.from_pretrained(model_name)
model=Wav2Vec2ForCTC.from_pretrained(model_name)
```

“”” Since the Wav2Vec2-Base-90h model has been pre-trained and fine-tuned using 90 hours of Librispeech on 16kHz sampled speech audio, so the speech input must be also sampled at 16kHz. For this the function load_data() makes sure that the speech input has a sampling rate of 16kHz.”””

#Creating a function that makes sure that the speech input has a sampling rate of 16kHz

```
def load_data(input_file):
```

#Reading a file

```
speech,sample_rate=librosa.load(input_file)
```

#Make it 1-D

```
if len(speech.shape)>1:
```

```
    speech=speech[:,0]+speech[:,1]
```

#Resampling the audio at 16kHz

```
if sample_rate!=16000:
```

```
    speech=librosa.resample(speech,sample_rate,16000)
```

```
return speech
```

#Creating the function for correcting the letter casing

“”” The correct_casing() will be needed in order to make necessary changes to the obtained transcript.”””

```
def correct_casing(input_sentence):  
    sentences=nlk.sent_tokenize(input_sentence)  
    return ('.join([s.replace(s[0],s[0].capitalize(),1) for s in sentences]))
```

#Defining a function for getting a transcript of the audio input

```
def asr_transcript(input_file):  
    speech=load_data(input_file)  
    #Tokenize  
    input_values=tokenizer (speech, return_tensors="pt").input_values  
    #take logits  
    logits=model(input_values).logits  
    #take argmax  
    predicted_ids=torch.argmax(logits,dim=-1)  
    #get the words from predicted word ids  
    transcription=tokenizer.decode(predicted_ids[0])  
    #Correcting the letter casing  
    transcription=correct_casing(transcription.lower())  
    return transcription
```

#creating a UI to the model using gr.Interface

“”” Using a Gradio’s Interface class, one can create a UI for the machine learning model by specifying the function, the desired input components, and the desired output components, allowing us to quickly prototype and test the model. Via asr_transcript, one can use microphones or drop an audio file via a file directory to provide audio input. In this case, the following code: gr.inputs=inputs for giving input, use: inputs=gr.inputs.Audio(source="microphone",type="filepath",optional=True,

Automatic Speech Recognition using Deep Learning

label="Speaker"). For output use: outputs= gr.outputs.textbox(label="Output Text") because the intended output is a string. Finally use the launch() method to start the demo.”””

```
gr.Interface( asr_transcript,
inputs=gr.inputs.Audio(source="microphone",type="filepath",optional=True,
label="Speaker"), outputs=gr.outputs.Textbox(label="Output Text"), title="ASR using
Wav2Vec 2.0", description="This application displays transcribed text for given audio
input(Please accord the audio in ENGLISH only)",
examples=[["my-audio.wav"],["secaudio.wav"],["male.wav"]],theme="grass").launch()
```

To upload audio files, click on the following tabs in the listed here:

“Files and versions”-> “Contribute”-> “Upload Files”

If an error occurred, go to the “See log” tab, which is right next to the spot where Runtime Error is shown, take a cue from the error log and fix the error.

Chapter-5: Results

The project result shows that deep neural networks have ability to solve speech recognition challenges. Comparing to other approaches, the approaches based on deep learning are showing rather interesting outcomes in several applications including speech recognition, and therefore, it attracts a lot of research and studies.

SNAPSHOTS

Snapshots of the project implementation from the creation of Hugging Face account to the output of transcription.

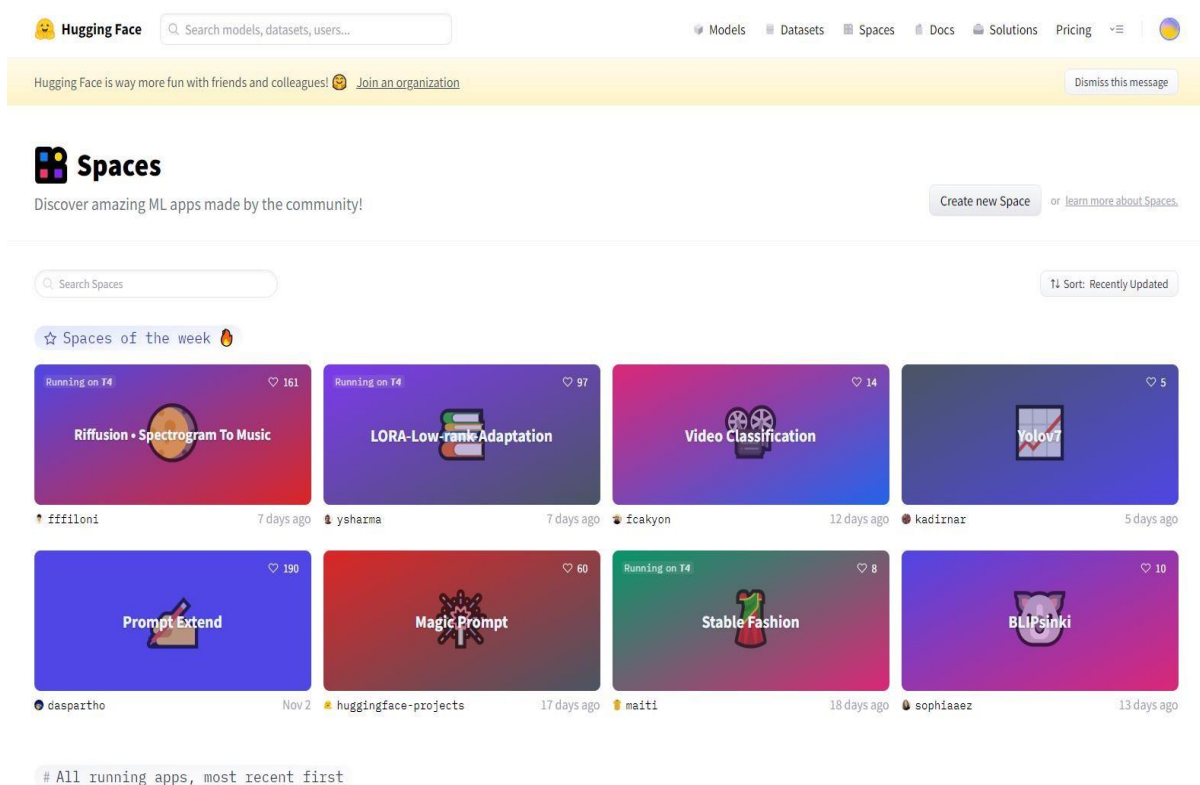
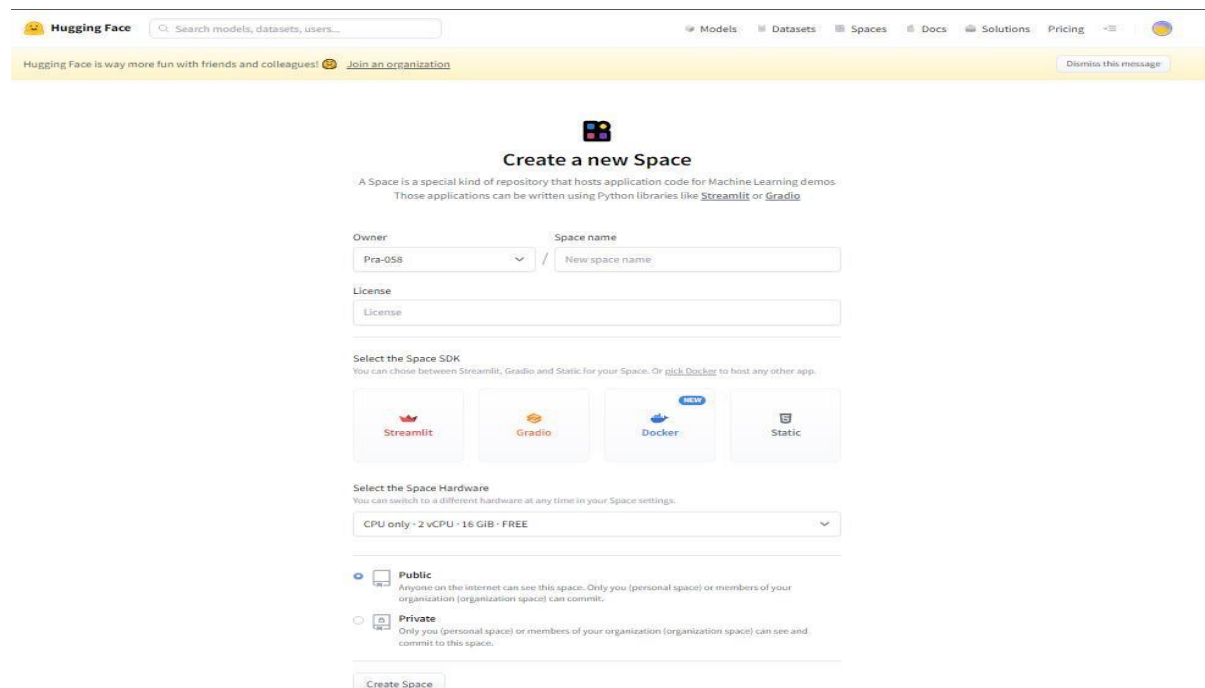


Figure 5: Hugging Face's user interface

Automatic Speech Recognition using Deep Learning



The screenshot shows the 'Create a new Space' form on the Hugging Face website. The form includes fields for 'Owner' (set to 'Pra-058'), 'Space name' (placeholder 'New space name'), and 'License'. Below these are options to 'Select the Space SDK' (Streamlit, Gradio, Docker, Static) and 'Select the Space Hardware' (CPU only - 2 vCPU - 16 GiB - FREE). There are also radio buttons for 'Public' and 'Private' visibility settings. A 'Create Space' button is at the bottom.

Figure 6: Creating a Hugging Face account and Setting up a new space

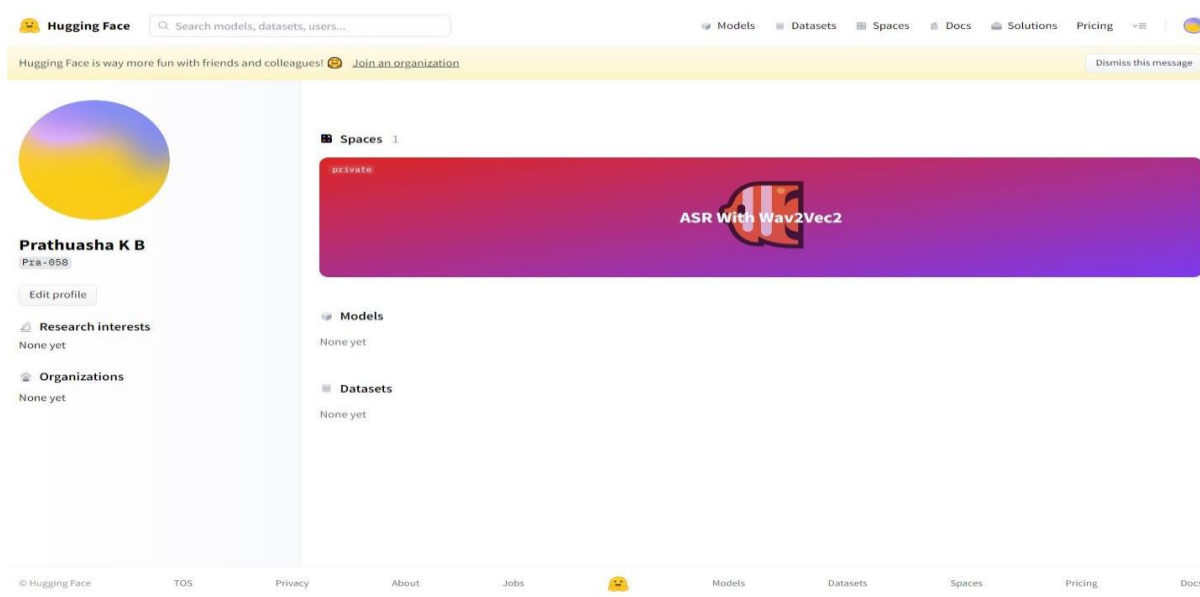


Figure 7: Dashboard of new Hugging Face account

Automatic Speech Recognition using Deep Learning

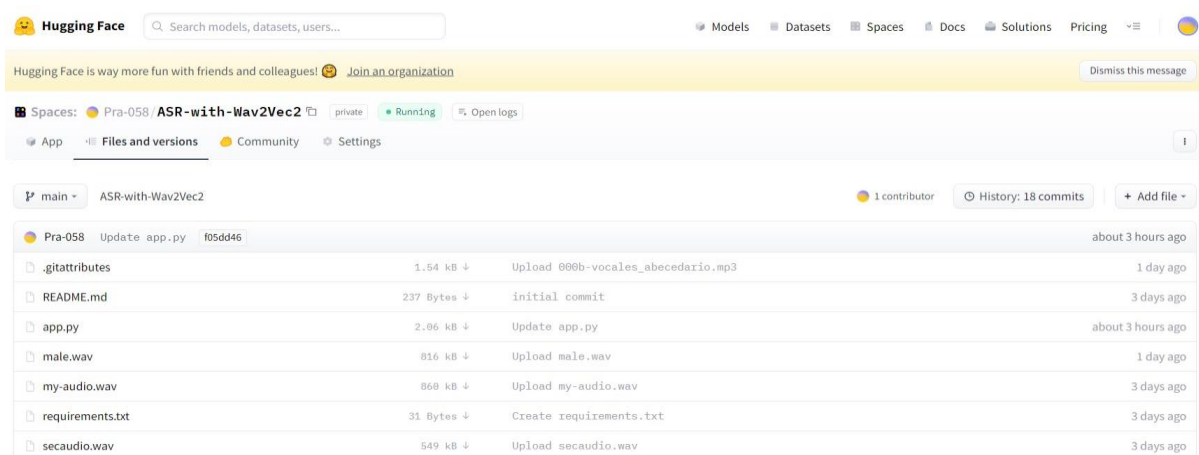


Figure 8: Spaces/Files and versions

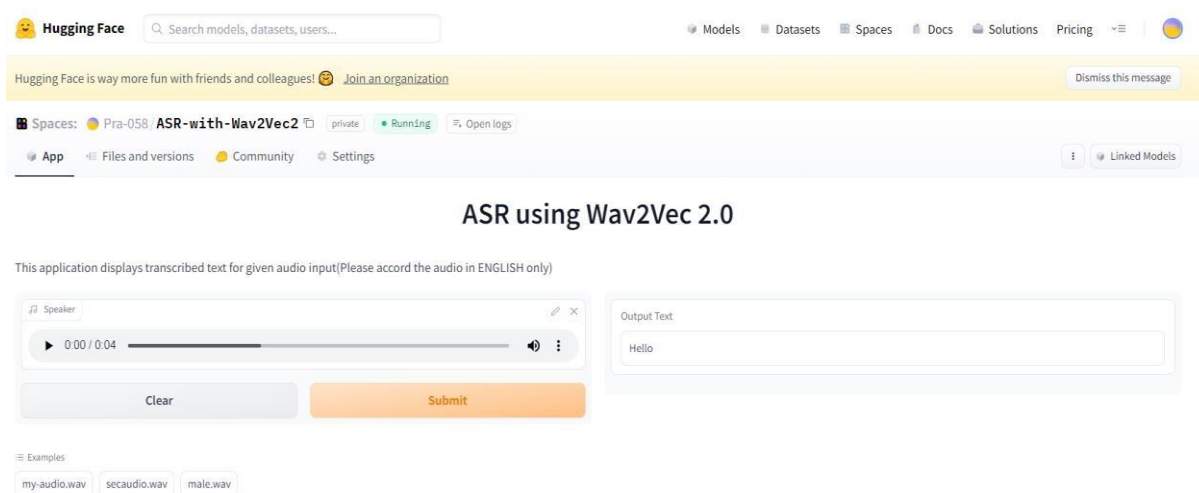


Figure 9: Output interface created spaces

Chapter-6: Conclusion

Speech recognition is a developing field. It's one of several ways people can connect with computers without having to type much. Despite its many intricacies, problems, and technicalities, ASR has one simple goal: to make computers listen to humans. This attribute is taken for granted in one another, but when a thought is given on it, it is realized just how critical it is.

Youngsters learn by paying attention to their parents and teachers. We improve our ideas by listening to the individuals we meet, and we keep our relationships strong by listening to each other.

Wav2Vec2.0 shows great potential when it comes to creating speech recognition models for settings where there is very little labelled training data. It can play a key role in devising ASR solutions for indigenous languages and domains with very limited annotated data.

The project result shows that deep neural networks have ability to solve speech recognition challenges. Compared to other approaches, the approaches based on deep learning are showing rather interesting outcomes in several applications including speech recognition, and therefore, it attracts a lot of research and studies.

As the field of ASR continues to grow, we can expect to see greater integration of Speech-to-Text technology into our everyday lives, as well as more widespread industry applications.

FUTURE ENHANCEMENTS

The Speech Recognition Technology has a huge scope in the future. The accuracy, language, accents, dialects are all challenges of the current Speech Recognition Technology.

There are more than 7000 languages spoken in the world, with an uncountable number of accents and dialects. English alone has more than 160 dialects spoken all over the world. To overcome this challenge, an effective way is to expand the dataset and aim to achieve optimum training for the AI/ML model which powers the technology.

An enhancement that can made in the future is the conversion of speech from one language into text of another language. That is, the translation of language included within the Speech Recognition Technology.

REFERENCES

- [1] Li Deng, Geoffrey Hinton, Brian Kingsbury, “New types of deep neural network learning for speech recognition and related applications: an overview”, IEEE International Conference on Acoustics, Speech, and Signal Processing 2013.

- [2] Phoemporn Lakkhanawannakun, Chaluemwut Noyunsan, “Speech Recognition using Deep Learning”, 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) 2019.

- [3] Dhilip Subramanian, “Speech to Text with Wav2Vec 2.0”, Towards AI 2021.

- [4] Drishti Sharma, “Automatic Speech Recognition Using Wav2Vec2”, Analytics Vidhya 2022.

- [5] Kazheen Ismael Taher, Adnan Mohsin Abdulazeez, “Deep Learning Convolutional Neural Network for Speech Recognition: A Review”, Article, ResearchGate.net 2021.