SPEECH TO SIGN LANGUAGE TRANSLATION FOR E- LEARNING

A PROJECT REPORT

Submitted by

Prachi Bhatt (19BAI10111)
Toshini Agarwal (19BAI10115)
Shivangi Singh (19BAI10097)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Specialization in

Artificial Intelligence and Machine Learning

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE

MADHYA PRADESH - 466114

May 2021

VIT BHOPAL UNIVERSITY,KOTHRIKALAN, SEHORE
MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **“SPEECH TO TEXT TRANSLATION FOR E-LEARNING”** is the bonafide work of “**PRACHI BHATT (19BAI10111), TOSHINI AGARWAL (19BAI10115) & SHIVANGI SINGH (19BAI10097)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. Sountharrajan ,
Senior Assistant Professor
School of AI & ML division
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. Amit Mishra ,
Assistant Professor
School of AI & ML division
VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on 4th May 2021.

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr Sountharrajan, Head of the Department, School of Computer Science and Engineering specialisation in Artificial Intelligence and Machine Learning for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide Mr Amit Mishra,for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Aeronautical Science, who extended directly or indirectly all support.

Last, but not the least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF FIGURES

FIGURE NUMBER	TITLE	PAGE NUMBER
Fig. 1	Logo Page	15
Fig. 2	User Detail Page	15
Fig. 3	Welcome Page	16
Fig. 4	User Direction Page	16
Fig. 5	Home Page	16
Fig. 6	Book Page	16
Fig. 7	Make Notes Page	17
Fig. 8	Translator Page	17

ABSTRACT

Due to the widespread dependence on voice contact in certain activities, individuals with hearing impairments, including partial or total loss of hearing, are unable to participate in live (synchronous) cooperative e-learning activities.Although speech-to-text systems provide a future route for Deaf people to pursue contact, they do not enable them to engage in the same mode.Not all deaf people can communicate in the same way: some use oral language, writing, and lip-reading to communicate, while others use Sign Language. Some deaf people are bilingual and use both types of communication.

People are now using various methods to make video streaming on the internet more available to all, such as adding captions to images, including audio description, and adding transcript to videos, both of which are beneficial to people with hearing disability. Through using our app, we want to make streaming videos more available to all.

In our app, we'll use sign language to assist people who are deaf or hard of hearing. Our app accepts audio and translates it into Indian Sign Language. Through our app, we aim to make the virtual world a more accessible environment for hearing-impaired people.By automatically producing a video-based speech to signing translation, our app aims to eliminate communication barriers by making the audio track material from online videos more accessible to hearing-impaired people.

Keywords: Hearing-impaired people, Indian Sign Language, speech-to-text

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	List of Abbreviations	iii
	List of Figures	iv
	Abstract	v
		5
1	**INTRODUCTION**	
	1.1 Introduction	8
	1.2 Motivation for the work	8
	1.3 About Introduction to the project including techniques	9
	1.5 Problem Statement	9
	1.6 Objective of the work	9
	1.7 Organization of the thesis	9
	1.8 Summary	9
2	**LITERATURE SURVEY**	
	2.1 Introduction	9
	2.2 Core area of the project	9

	2.3 Existing Algorithms	10
	2.3.1 Algorithm 1	
	2.4 Research issues/observations from literature Survey	10
	2.5 Summary	12
3	**SYSTEM ANALYSIS**	
	3.1 Introduction	12
	3.2 Disadvantages/Limitations in the existing system	13
	3.3 Proposed System	13
	3.4 Summary	14
4	**SYSTEM DESIGN AND IMPLEMENTATION**	
	4.1 Introduction	14
	4.2 Module 1 design & implementation	14
	4.3 Module 2 design & implementation	17
	4.4 Module 3 design & implementation	17
	4.5 Summary	
5	**PERFORMANCE ANALYSIS**	
	5.1 Introduction	17
	5.2 Summary	17
6	**FUTURE ENHANCEMENT AND CONCLUSION**	
	6.1 Introduction	18
	6.2 Limitation/Constraints of the System	18
	6.3 Future Enhancements	18
	6.4 Conclusion	

		18
	Appendix A Appendix B References	18 56 58

1. INTRODUCTION

1.1 Introduction

Since the COVID -19 pandemic has disrupted the conventional lifestyle of individuals across the world, the virtual world has come to the rescue. Amongst several educational establishments faculties have shifted their base to virtual platforms to conduct classes online. But there are many students around the globe who struggle to attend their online classes properly because of internet connectivity issues, not having proper devices, not having proper places to study etc.

But there are people who struggle to understand what's being taught in the virtual classes because of certain sensory, cognitive or functional disabilities.

Specifically, individuals with hearing impairments, with partial or complete loss of hearing are prevented from participating in live (synchronous) cooperative e-learning activities, because of the common reliance on voice communication in such activities. whereas speech-to-text systems offer an eventual pathway for the Deaf individuals to follow the communication, it doesn't enable them to participate within the same mode. Not all deaf individuals have the same communication skills: some people communicate using oral language and/or writing and lip-reading, whereas others use Sign Language. Some are bilingual deaf people that use both kinds of communication.

Our project is based on this problem. We are making an app that converts audio to Indian Sign language and helps people with hearing impaired in their daily online activities. Our app can convert the audio from an ongoing meeting or a virtual class or any video on a mobile phone to ISL.

1.2 Motivation for the work

Nowadays, to make video streaming on the internet more accessible for everyone, people are using different ways to achieve this, such as adding captions in videos, including audio description or adding transcript to videos which are helpful for people with hearing impairment. We want to make streaming videos a bit more accessible for everyone by using our app.

In our app we are going to use Sign language to help people with hearing impairment. Our app takes audio as an input and converts it into Indian Sign Language. With our app we hope to make the virtual world a better facility for the hearing impaired individuals . Our app aims to get rid of communication barriers by creating the audio track content from online videos obtainable to hearing-impaired people by mechanically generating a video-based speech to signing translation.

Our app will be easy to use and easily accessible for everyone.

1.3 About Introduction to the project including techniques

The main idea of our app is that our app detects a video that is currently being played on a mobile phone, converts it into an audio script and then converts that audio script to Indian Sign Language.
Therefore our project model is divided into four modules : Module1 - this module is the outlook of our app (UI of the app) , Module2 - this module converts audio to text using Google Speech API, Module3 - this module converts the text into ISL words, Module 4 - this module represents ISL words for the deaf people to understand what is being said in the video.

1.4 Problem Statement

To help people with hearing impaired make E-Learning a bit easier.

1.5 Objective of the work

- Make video and audio streaming accessible for hearing impaired people.
- Make more and more people become aware of the problems faced by disabled people.

1.6 Organization of the thesis

In the first section, we have introduced our proposed application and how beneficial it will be for the hearing imapired people.

In the second section i.e. literature survey we see where the logic behind this app comes from. The various research papers are helpful in the making of this app as those papers tell us about the statistics of deaf people around the world, the difficulties faced by them, the ways we can help them and different work done in the text-to-speech model.

In the third section i.e. System Analysis, we have explained the basic workflow of our application, the various advantages and disadvantages of existing work.

In the fourth section i.e. System Design and Implementation, includes the information about the various modules used in our project in detail and their implementation.

In the sixth section i.e. Future Enhancements and Conclusion, we proposed the various improvements we will add in our app to make our application more accessible for the deaf people.

1.7 Summary

In this section, we proposed the idea that can help hearing disabled people and improve their E-learning experience. For many people hearing loss can become a big problem in learning stuff online. To solve this problem we are working on an app that can bridge the gap between the virtual world and deaf people.

LITERATURE SURVEY

2.1 Introduction

Many papers have been published regarding the difficulties faced by deaf people and how it can be improved. Many researches have been done in the area of speech-to-text conversion and have thus improved the applications which use it. This Literature survey will briefly tell about these papers and researches.

2.2 Core Area of the Project

The core area of the project is mainly focused on how AI models can be used to convert audio scripts to text scripts and then later these text scripts can be converted to Indian Sign Language.

2.3 Existing Algorithm

2.3.1 Algorithm 1

The implementation of the V2S system as an alternative solution to minimise the cost of hiring special professional workers for media agencies, according to the authors of the 'Voice to Sign Language Translation System for Malaysian Deaf People' Paper[5]. They used three phases in their proposed work in their work: A.Sound Recording – a microphone is used to capture and record sound. Digital Signal Processing (DSP) is a type of digital signal processing.

2.4 Research issues/observations from literature Survey

Authors of Paper[1] mentioned that India has the largest Deaf population in the world. Perhaps "one of every five people who are deaf in the world, live in India" as the country with the large number of deaf people so it is highly important to use some mode of communication between deaf people and normal people hence sign language is required But still ISL is in developing stage.Hence because of these our research to develop a learning system will totally based on indian sign language to gain the status of sign language as a minority language.

Paper[2] reports on an investigation how VCs and instructors are working on identifying the barriers and problems of distant learning for deaf students managing them effectively in King Saud University (KSU) as during COVID-19 pandemic schools are shut down so, there is sudden need for some system which is capable of providing effective distant learning. Hence our research will directly help in solving the problems in this field.

Paper [3] says Voice recognition can be separated into two main categories: speech recognition and speaker recognition.Both are different in functionalities but in many cases they work hand in hand to provide "speaker-dependent" speech recognition. Speech recognition is nothing but the ability for a device to recognize individual words or phrases from human speech. Speaker Recognition:Every human being has a unique voice, and by taking this advantage we can use it for security purposes as a biometric of personal data and files.. After this model automatically generates the vocal patterns for each user. In comparison with physical passwords or keys, voice biometrics are more stronger.

Authors of paper [4] applied a method of using a synthetic named animation making approach; they have converted Malayalam language to Indian sign language. The intermediate representation in this method for sign language is being used by HamNoSys(basically it is a mobile application). In this method the application accepts some words, generates it in an animated portion then the interactive system will convert it into HamNoSys designed structure. Here this system helps as a communication mode between deaf and normal people . But there should be only one language to communicate to decrease lots of confusion and time required to learn different regional sign language. That's why Our system will focus on only ISL.

Authors of 'Voice to Sign Language Translation System for Malaysian Deaf People ' Paper[5], says , The development of this V2S system as an alternative solution to reduce the cost of employing special skilled employees for media agencies. In their work they used three stages in proposed work : A.Sound Recording – for capturing and recording the sound using a microphone. B. Digital Signal Processing – The "trained" sound from the sound recording is then fed into the Digital Signal Processing(DSP) part.C. Vector Quantization – Vector quantization is used to perform speech recognition. The main drawback of V2S is, it only act as as an interpreter considering economic factors but the main need of current time is education for deaf people by introducing an efficient e learning application using ISL

.Paper [6] ‘Sign Language translation using Machine Learning’ The main objective of this project is to recognize the gestures(using camera) and display the corresponding word. Our research is just opposite to this which is it converts speech to ISL using deep learning.An architecture was proposed in [7],It was a speech-to-gesture translator and it was divided into four modules namely, speech recogniser,semantic analysis, gesture sequence generation and gesture sequence animation.

The main work was done on gestures generation and animation.The phoenix parser was used for the semantic analysis.The drawbacks include how the designer chose a finite set to perform the gesture animation and were not able to generate all the possible positions of a gesture.

During the modulation in[8],it was examined that for sign language HamNoSys was used which is a very useful notation compared with Stokoe and SignWriting.It was able to write sign notations in every sign language.

Furthermore,it has the capability to provide written form of sign language in English or any other language.Here though they only experimented with Punjabi language which can be diversified to Indian Sign language.

Mohamed Abu Elsoud proposed a framework in [9]which was web-based and provided e-learning as well as a dictionary system for Arab deaf students, it was an open application where the user can enter the URL which they want to convert to sign language and get a output as a photo album display or reading line by line .The request is processes at server end and then line by line response is sent to the user .Also,it has provided a dictionary system which will help improve their skills.In [10],named “Sign language translator using deep learning”, the proposed system is designed to do hand gestures into written English text .Here, the hand movements are considered as inputs which are captured through the video and is translated to text.Here, every word is broken down into alphabet which is then converted to gesture, through this the words can be converted followed by sentences.A translator for communication between a signer and a non-signer.The dynamic hand gesture recognition in [11] is done for international sign language, it is made for a smart home environment where there is a touchless interface which provides a sign language recognition. Here the neural networks were set as are for sign language .Here the S2V conversion has shown sufficiently good rate of recognition.Although small sized vocabularies were not there and there was lack of continuous recognition as well.

Authors of paper [12] have mentioned that, automatic sign language recognition system creates a new way of human-computer-interaction by transferring sign gestures into text/speech, and this technology can be implemented by computer-aid technology, like deep learning.Dynamic sign language recognition also has challenges in dealing with the complexity of the sign activities of finger motion from the large scale body background. Another difficulty is that extracting the most discriminating features from images or videos.

In paper [13] the authors mention that traditionally, there have been two main approaches to sign language recognition: vision-based and glove-based systems. In vision-based systems, a signer performs the different signs which are captured with a video camera.For glove-based systems, the signer wears certain instrumented gloves equipped with a number of sensors that generate a set of electrical signals that characterize the intended sign.

In paper [14], the authors mention the difficulties faced in visual sign language recognition specifically in the British sign language system. They use the visual, rather than auditory, modality but are as lexically rich and grammatically complex as spoken languages. The expression in sign languages of both non-topographic space and topographic space requires linguistic representations, with non-topographic space using spatial relationships to represent abstract relationships, and topographic space appearing to recruit real-world image structure directly.

In paper [15] it is mentioned that since sign languages make use of manual and facial means of expression, both channels are employed for recognition. The classification stage is designed for recognition of isolated signs, as well as of continuous sign language. In the latter case, a stochastic language model can be utilized, which considers uni- and bigram probabilities of single and successive signs.

According to paper [16], the authors mention that the principal constituent of any sign language recognition system is hand gestures and shapes normally used by deaf people to communicate among themselves.Static gestures refer to a certain pattern of hand and finger orientation whereas dynamic gestures involve different movement and orientation of hands and face expressions largely used to recognize continuous streams of sentences.

2.5 Summary

This was the review of various research papers that have been read for the implementation of our project. We can see from the various reviews that how this audio to sign language can help hearing disabled people learn from online platforms if used correctly.

3. System Analysis

3.1 Introduction

The android application that takes audio as an input and converts it into
Indian Sign Language. The audio can be taken from an ongoing virtual meeting or some kind of video playing on the phone. App is used in split screen mode so that the user can see the video as well as understand the audio of the video through sign language.

Further, the app is coded in Java language using Android Studio and we have provided it with features in reference to our title statement, “SPEECH TO SIGN LANGUAGE TRANSLATOR FOR E-LEARNING ”.

The system is defined into three stages namely speech recognition, synchronous translation and representation. In speech recognition the audio will be converted and preprocessed to useful dataset and then through neural network model (Neural Machine Translation),it will be transformed to sign language which will be represented by hand animation which will be visible to the user to understand the lecture easily and without any interruptions.

In Short the application's workflow is as follows:

1.App starts with a splash screen of 3 secs

2. Then login activity appears

3. Tabbed activity starts with 3 tabs

4. Welcome(tab1) activity with a motivational quote

5. Direction Activity(tab2) –steps are given to use split screen mode in android so that user can see the video lecture side by side

6. Translation Activity(tab3)- it has an image button by clicking on it, it starts taking audio and converts it to text (using google api)then text is transferred to a deep learning model to generate hand gestures for the audio.

7. Next Navigation bar activity starts with the home page which consists of creating an online meet feature.(Future work)

8. Next fragment is 'Books' here pdfs and links of books will be provided.

9. Next is 'Make notes' fragment to create study notes here users can edit the notes and can save it.

3.2 Disadvantages / Limitations in existing work

In India, there are no apps that convert audio to Indian Sign Language.

There are a lot of apps available that teach Indian Sign Language but they do not convert audio to Indian Sign Language.

The existing apps don't convert audio to Sign language real time, they convert one sentence at a time.

3.3 Proposed System

Module 1:-Android Features:-

This module consists of Authentication-that is a login screen , welcome Activity with a welcome quote, then User guide on how to split screen, and also some attractive features like online meet creating option in "Home" fragment , Books fragment int is fragments some book pdfs will be attached to read books, and the last but not least an interactive feature i.e. "Make Notes" fragment to edit and save important notes and study material.

Module2: Recognition:-Using Google API the audio is converted into text.

Module3:-Translation:-In this Module the text is converted to Indian sign language and the layout for this is tabbed activity with title “Translator”

Module4:- Representation:-And finally the translated sign sequence would be fed to the model which will produce the animated hand gestures using animation.

3.4 Summary

The system provided by us is a flexible system, easy to use. Proper instructions and graphics have been provided with a decent basic color in the background. As the user completes the login a page with a motivation quote welcomes the user to read instructions to be followed for split screen mode in android and finally the translator is ready to use. Some extra features are also provided in the application like a virtual library to read books and also a fragment to write and save notes.

4. SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

Our application focuses on providing translation of ongoing lectures into Indian Sign Language for educational purposes. As the deaf community have been devoid of education due to pandemic which in turn reduces their job opportunities, in this we have tried to reduce that by providing a helping aid through our Speech2Sign App so that they could learn with everyone and practice and compete with equal mindsets. The app has been designed in such a way that it's user friendly and is visually appealing.

4.2 Module 1 design & Implementation

Module 1:-Android Features:-

This module consists of Authentication, welcome Activity, User guide on how to split screen, Home, Books, Make Notes.

Splash activity, tabbed activity, and navigation bar activity have been used to create the pages. Responsibilities include splash activity, set time to Handler, and call Handler().postDelayed, it will call a run method of runnable after a set time(3000) and redirect to the main app screen. .Secondly, the tabbed activity allows navigation between sibling fragments or screens and it is also known as horizontalPaging.Lastly, in the navigation bar activity, Navigation drawer, toolbar, and back button. The development has been done through Java and XML and the platform is the android studio and the database has been done through SQLite.

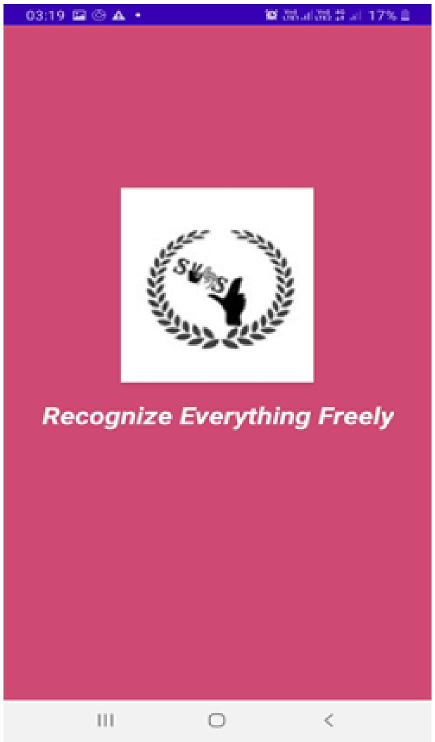
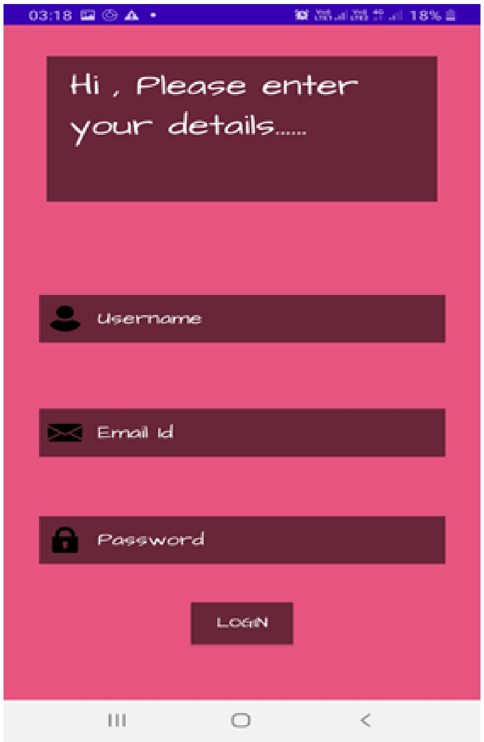
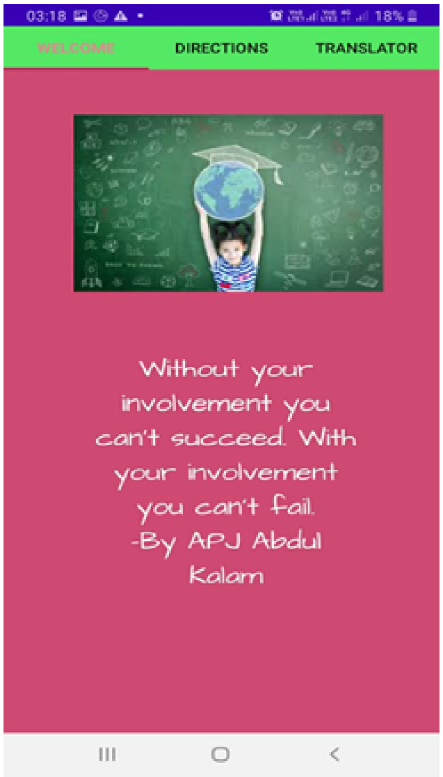


Fig. 1



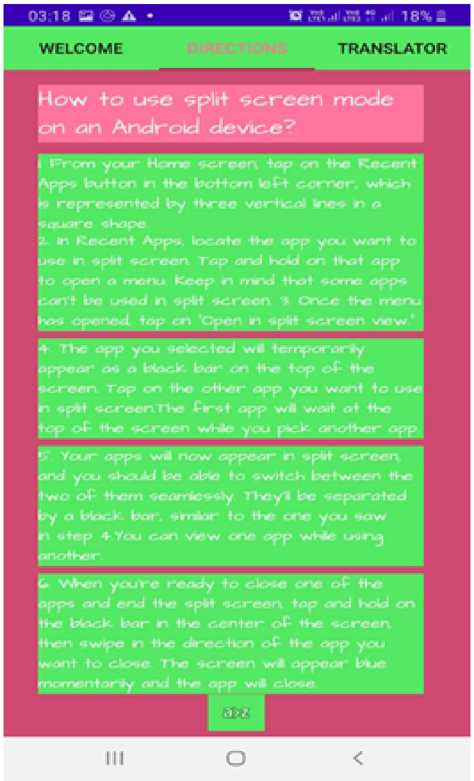
Hi , Please enter your details......

Fig.2



Without your involvement you can't succeed. With your involvement you can't fail.
-By APJ Abdul Kalam

Fig.3



How to use split screen mode on an Android device?

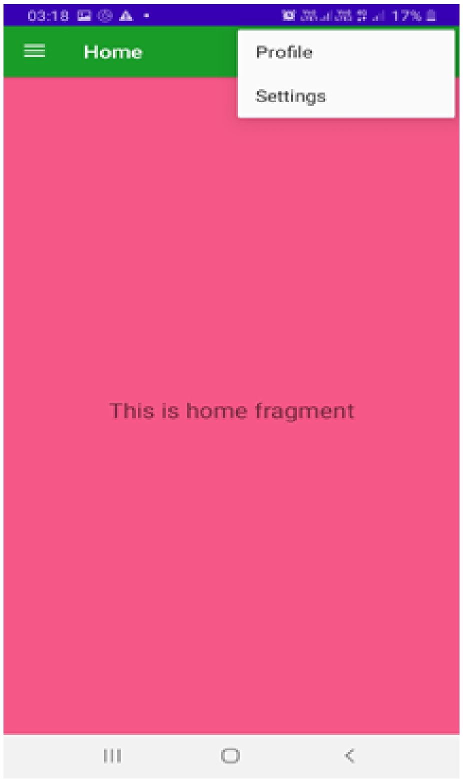
1. From your Home screen, tap on the Recent Apps button in the bottom left corner, which is represented by three vertical lines in a square shape.
2. In Recent Apps, locate the app you want to use in split screen. Tap and hold on that app to open a menu. Keep in mind that some apps can't be used in split screen. 3. Once the menu has opened, tap on "Open in split screen view."

4. The app you selected will temporarily appear as a black bar on the top of the screen. Tap on the other app you want to use in split screen.The first app will wait at the top of the screen while you pick another app.

5. Your apps will now appear in split screen, and you should be able to switch between the two of them seamlessly. They'll be separated by a black bar, similar to the one you saw in step 4.You can view one app while using another.

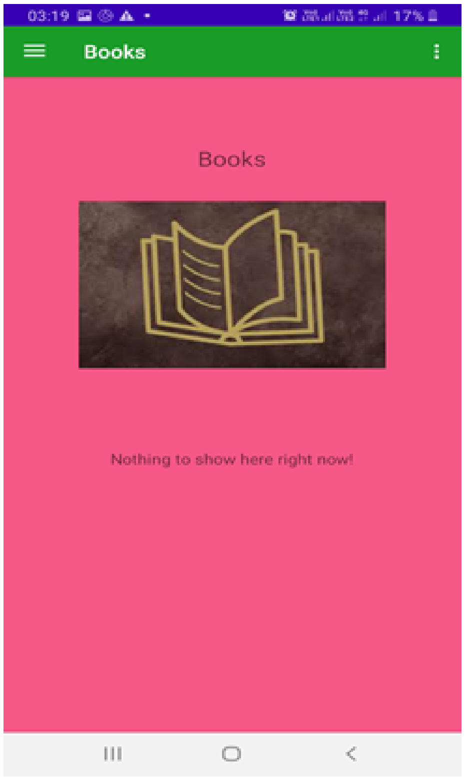
6. When you're ready to close one of the apps and end the split screen, tap and hold on the black bar in the center of the screen, then swipe in the direction of the app you want to close. The screen will appear blue momentarily and the app will close.

Fig.4



This is home fragment

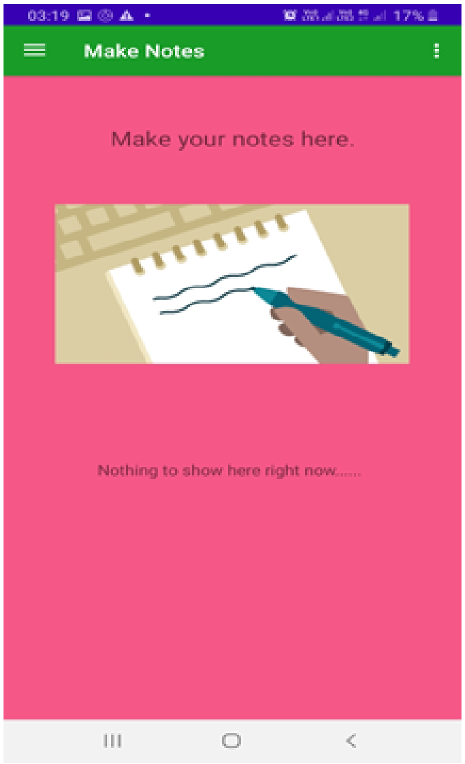
Fig. 5



Books

Nothing to show here right now!

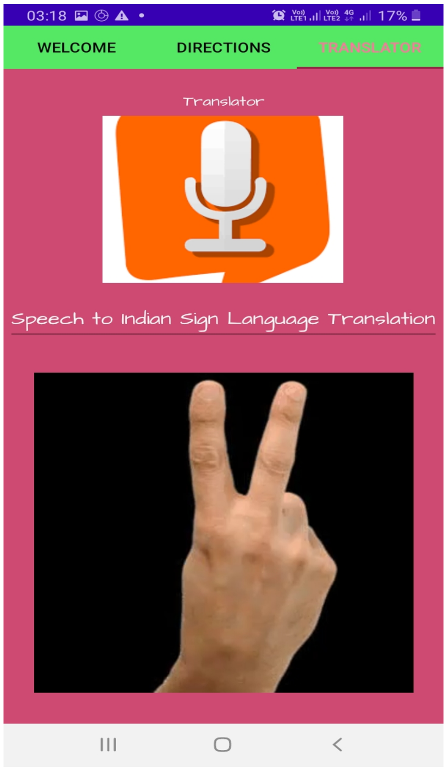
Fig.6



Make your notes here.

Nothing to show here right now......

Fig.7



Translator

Speech to Indian Sign Language Translation

Fig.8

4.2 Module 2 design & Implementation

Module2: Recognition:-Using Google API the audio is converted into text.

STEP1:Extract the audio from a video and transcode it into a format the Speech API accepts.(FLAC file).
STEP 2:-Send the FLAC file to the Speech API and upload transcriptions to Cloud Storage.
STEP 3:-make a request to Cloud Speech with the client.
STEP 4:-Once processed, the API will return a response.

4.3 Module 3 design & Implementation (Work in progress)

Module3:-Translation:-Input, Natural speech and preprocessing and feature extraction to get prepared and trained data(from Recognition) Next , when the dataset is ready it will go through an acoustic model An acoustic model is used in automatic speech recognition to represent the relationship between an audio signal and the phonemes or other linguistic units that make up speech. The model is learned from a set of audio recordings and their corresponding transcripts. We use jasper (CNN model) for this. We have found that the model achieves better WER with longer training but even with a small number of epochs, the model can achieve a good WER. Next, there will be a language model which will help in removing the semantic and syntactic errors which will be CTLM (cache trigram language model) as it reduces the perplexity of dictated documents By 23% Then MongoDB will be used to store all these datasets. When we will have the dataset, then the word sequence will be passed on for language translation for sign language. The model selected (preferably -LSTM) where modelling will be done through an encoder and decoder where we would have a trained model for accurate translation. Furthermore, the synchronous translation will happen through the use of Tensorflow and the neural machine translation framework which was proposed where an agent learns to make a decision on when to translate from the interaction with a pre-trained NMT environment.

4.4 Module 4 design & Implementation (Not implemented yet)

Module4:- Representation:-And finally the translated sign sequence would be fed to the model which will produce the animated hand gestures fed to the model from the 3d blender tool.

5.PERFORMANCE ANALYSIS

5.1 Introduction

The app performance is good when we see from the user point of view. It starts with positive quotes which makes oneself feel good. While using the app, clicking its logo it opens and smoothly moves on to the next page. We have to click on “Image Button”in the direction activity(tabbed activity) to move on to the Navigation bar activity fragments. While using the app, it’s simple well guided features helps the user to explore more features into it. A page with a beautiful quote and inspirational quote helps the user to connect with the application. The Translator fragment of the app has been made very well managed. There is an option of split screen view to see the video content and sign language side by side. This really makes an app perform better and understandable.

5.2 Summary

The app performs very well according to the instruction and coding provided. It is well managed with proper scheduled functions and commands. It is easy to understand and use. There is very less percentage of error in the app.

6. FUTURE ENHANCEMENT AND CONCLUSION

6.1 Introduction

There are always ways in which new technologies can be improved. Similarly, in the Speech2ISL app there are few things which can be installed and improved in order to make the user feel more comfortable with E-Learning and understand the things properly.

6.2 Limitation/Constraints of the System

- The main limitation of our system is that synchronous translation as the audio from the source is lagging to translate simultaneously.
- Latest UI/UX trends are not used.

6.3 Future Enhancements

- We will add an online meet create feature in our application.
- We would try to integrate it further to have synchronous translation as the audio from the online video lecture is translated simultaneously without any buffer directly to animated hands so as they have no interruptions during the learning. We would test it further with institutions by serving this app to the deaf community near us and try to fix the problems according to them.
- We will add some latest UI/UX trends in our application in order to enhance the user experience.
- Add an option to record the session for future reference
- We will create a Movable screen option to move the animated hands screen in any direction with respect to the video lecture screen for visual convenience.

6.4 Conclusion

Our app was built on the fact that the online platforms like Microsoft Teams ,Zoom have been given the title of"online classroom" where the study lectures can take place from the comfort of a home. Although,through ages these platforms were not designed to suit the deaf community.Due to pandemic the whole education has shifted on platforms like these , it has been difficult for the deaf to be able to attend these. Therefore,we made a framework where a deaf student can attend any online lecture or any webinar without having to worry about the translation.We would make it more interactive so they don't feel disconnected from the real process. Our app has achieved some of the ideas that wanted to achieve the above goal.We will enhance some things which will make it much more attractive and effective as well.This app has been made by keeping in mind the fact why would users use this and how beneficial can he/she be made in order to get a proper chance to enjoy E-Learning.

Appendix A

Module 1

<u>SPLASH ACTIVITY:</u>

```
package com.appcodewithprachi.speech2isl;

import android.app.Activity;
import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

public class SplashScreenLogo extends Activity {

Handler handler;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.splash_screen_logo);

handler=new Handler();

handler.postDelayed(new Runnable() {

@Override

public void run() {

Intent intent=new Intent(SplashScreenLogo.this, LoginActivity.class);

startActivity(intent);

finish();

}

},3000);

}

}
```

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent"

android:background="#CF4A72"
```

```
tools:context=".SplashScreenLogo">

<TextView

android:id="@+id/textView"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginBottom="308dp"

android:backgroundTint="?android:attr/colorFocusedHighlight"

android:fontFamily="@font/dancing_script"

android:text="Recognize Everything Freely"

android:textColor="@color/white"

android:textSize="25sp"

android:textStyle="bold|italic"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent" />

<ImageView

android:id="@+id/imageView3"

android:layout_width="263dp"

android:layout_height="223dp"

android:src="@drawable/logoisl"

app:layout_constraintBottom_toTopOf="@+id/textView"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.497"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.908" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

LOGIN ACTIVITY

```
package com.appcodewithprachi.speech2isl;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
public class LoginActivity extends AppCompatActivity {
Button login;
EditText et;
SharedPreferences sp;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
login = findViewById(R.id.submit);
et = findViewById(R.id.name);
login = (Button) findViewById(R.id.submit);
sp = getSharedPreferences("login",MODE_PRIVATE);
if(sp.getBoolean("logged",false)){
goToTabbedActivity();
}
login.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

goToTabbedActivity();

sp.edit().putBoolean("logged",true).apply();

}

});

}

public void goToTabbedActivity(){

Intent i = new Intent(this,TabbedActivity.class);

startActivity(i);

finish();

}

}
```

xml code:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent"

android:background="#E75480"

tools:context=".LoginActivity">

<TextView

android:id="@+id/textView"

android:layout_width="337dp"

android:layout_height="166dp"

android:layout_marginTop="36dp"

android:background="#69263A"
```

```
android:fontFamily="@font/architects_daughter"

android:paddingLeft="20dp"

android:paddingTop="10dp"

android:text="@string/text_view"

android:textColor="?android:attr/windowBackground"

android:textSize="30sp"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent" />

<EditText

android:id="@+id/name"

android:layout_width="350dp"

android:layout_height="55dp"

android:background="#69263A"

android:textColorHint="@color/white"

android:layout_centerHorizontal="true"

android:layout_marginHorizontal="10dp"

android:layout_marginBottom="76dp"

android:autofillHints="username"

android:fontFamily="@font/architects_daughter"

android:hint="Username"

android:inputType="textPersonName"

android:paddingStart="50dp"

android:paddingLeft="30dp"

android:textColor="@color/colorPrimaryDark"

app:layout_constraintBottom_toTopOf="@+id/email_id"

app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.500"
app:layout_constraintStart_toStartOf="parent"
tools:ignore="HardcodedText,MissingConstraints" />
<EditText
android:id="@+id/email_id"
android:layout_width="350dp"
android:layout_height="55dp"
android:background="#69263A"
android:layout_below="@+id/name"
android:layout_centerHorizontal="true"
android:textColorHint="@color/white"
android:layout_marginHorizontal="10dp"
android:layout_marginBottom="68dp"
android:autofillHints="Email Id"
android:fontFamily="@font/architects_daughter"
android:hint="Email Id"
android:inputType="textEmailAddress"
android:paddingStart="50dp"
android:paddingLeft="30dp"
android:textColor="@color/colorPrimaryDark"
app:layout_constraintBottom_toTopOf="@+id/password"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.500"
app:layout_constraintStart_toStartOf="parent"
tools:ignore="HardcodedText,MissingConstraints" />
<EditText
android:id="@+id/password"
```

```
android:layout_width="350dp"

android:layout_height="55dp"

android:layout_below="@+id/email_id"

android:background="#69263A"

android:layout_centerHorizontal="true"

android:textColorHint="@color/white"

android:layout_marginHorizontal="10dp"

android:layout_marginBottom="156dp"

android:autofillHints="Password"

android:fontFamily="@font/architects_daughter"

android:hint="Password"

android:inputType="textPassword"

android:paddingStart="50dp"

android:paddingLeft="30dp"

android:textColor="@color/colorPrimaryDark"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.500"

app:layout_constraintStart_toStartOf="parent"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/submit"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/password"

android:fontFamily="@font/architects_daughter"

android:text="@string/submit"
```

```
android:textColor="@android:color/white"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.500"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/password"

android:background="#69263A"

app:layout_constraintVertical_bias="0.406" />

<androidx.constraintlayout.widget.Barrier

android:id="@+id/barrier"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

app:barrierDirection="top" />

<ImageView

android:id="@+id/user_icon"

android:layout_width="40dp"

android:layout_height="55dp"

android:layout_marginStart="28dp"

android:layout_marginLeft="28dp"

android:paddingLeft="10dp"

app:layout_constraintBottom_toTopOf="@+id/email_id"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/name"

app:layout_constraintVertical_bias="0.0"

app:srcCompat="@drawable/username_icon" />

<ImageView

android:id="@+id/email_icon"
```

```
android:layout_width="40dp"

android:layout_height="55dp"

android:layout_marginStart="28dp"

android:layout_marginLeft="28dp"

android:paddingLeft="10dp"

app:layout_constraintBottom_toTopOf="@+id/password"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/email_id"

app:layout_constraintVertical_bias="0.0"

app:srcCompat="@drawable/email_id_icon" />

<ImageView

android:id="@+id/password_icon"

android:layout_width="40dp"

android:layout_height="55dp"

android:layout_marginStart="28dp"

android:layout_marginLeft="28dp"

android:paddingLeft="10dp"

app:layout_constraintBottom_toTopOf="@+id/submit"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/password"

app:layout_constraintVertical_bias="0.0"

app:srcCompat="@drawable/password_icon" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

TABBED ACTIVITY:

```
package com.appcodewithprachi.speech2isl;

import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;
import android.os.Bundle;
import com.google.android.material.tabs.TabItem;
import com.google.android.material.tabs.TabLayout;
public class TabbedActivity extends AppCompatActivity {
TabLayout tabLayout;
TabItem tabItem1,tabItem2, tabItem3;
ViewPager viewPager;
PageAdapter pageAdapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_tabbed);
TabLayout tabLayout = (TabLayout) findViewById(R.id.tablayout1);
TabItem tabItem1 = (TabItem) findViewById(R.id.tab1);
TabItem tabItem2 = (TabItem) findViewById(R.id.tab2)
TabItem tabItem3 = (TabItem) findViewById(R.id.tab3);
final ViewPager viewPager = (ViewPager) findViewById(R.id.vpager);
pageAdapter=new PageAdapter(getSupportFragmentManager(),tabLayout.getTabCount());
viewPager.setAdapter(pageAdapter);
tabLayout.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
@Override
public void onTabSelected(TabLayout.Tab tab) {
viewPager.setCurrentItem(tab.getPosition());
if(tab.getPosition()==0 || tab.getPosition()==1 || tab.getPosition()==2)
pageAdapter.notifyDataSetChanged();
}
@Override
public void onTabUnselected(TabLayout.Tab tab) {
}
@Override
public void onTabReselected(TabLayout.Tab tab) {
}
});
viewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tabLayout));
//listen for scroll or page change
}
}
```

PAGEADAPTER ACTIVITY JAVA

```
package com.appcodewithprachi.speech2isl;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;
public class PageAdapter extends FragmentPagerAdapter
{
int tabcount;
public PageAdapter(@NonNull FragmentManager fm, int behavior) {
super(fm, behavior);
tabcount=behavior;
}
@NonNull
@Override
public Fragment getItem(int position)
{
switch (position)
{
case 0 : return new QuotesFragment();
case 1 : return new DirectionFragment();
case 2 : return new ConverterrFragment();
default: return null;
}
}
@Override
public int getCount() {
return tabcount;
}
}
```

WELCOME ACTIVITY:

```
package com.appcodewithprachi.speech2isl;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
/**
    ●  A simple {@link Fragment} subclass.
    ●  Use the {@link QuotesFragment#newInstance} factory method to
    ●  create an instance of this fragment.
    ●  /
public class QuotesFragment extends Fragment {
```

```
// TODO: Rename parameter arguments, choose names that match
**// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
// TODO: Rename and change types of parameters
**private String mParam1;
private String mParam2;
public QuotesFragment() {
// Required empty public constructor
}
/**
** Use this factory method to create a new instance of*
** this fragment using the provided parameters.*
```

```
** @param param1 Parameter 1.*
** @param param2 Parameter 2.*
** @return A new instance of fragment QuotesFragment.*
*/
**// TODO: Rename and change types and number of parameters
**public static QuotesFragment newInstance(String param1, String param2) {
QuotesFragment fragment = new QuotesFragment();
Bundle args = new Bundle();
args.putString(ARG_PARAM1, param1);
args.putString(ARG_PARAM2, param2);
fragment.setArguments(args);
return fragment;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
if (getArguments() != null) {
mParam1 = getArguments().getString(ARG_PARAM1);
mParam2 = getArguments().getString(ARG_PARAM2);
}
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
// Inflate the layout for this fragment
return inflater.inflate(R.layout.fragment_quotes, container, false);
}
}
```

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/constraintLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#CF4A72"
tools:context=".QuotesFragment">
<!-- TODO: Update blank fragment layout -->
<TextView
android:id="@+id/quote"
```

```
android:layout_width="260dp"

android:layout_height="wrap_content"

android:gravity="center"

android:layout_marginBottom="152dp"

android:fontFamily="@font/architects_daughter"

android:text="Without your involvement you can’t succeed. With your involvement you can’t fail.

-By APJ Abdul Kalam"

android:textColor="@color/white"

android:textSize="25sp"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.528"

app:layout_constraintStart_toStartOf="parent" />

<ImageView

android:id="@+id/education_quote"

android:layout_width="292dp"

android:layout_height="276dp"

android:layout_marginBottom="24dp"

android:src="@drawable/education"

app:layout_constraintBottom_toTopOf="@+id/quote"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.554"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="1.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

// TODO: Rename parameter arguments, choose names that match
```

```
**// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
// TODO: Rename and change types of parameters
**private String mParam1;
private String mParam2;
public DirectionFragment() {
// Required empty public constructor
}
/**
** Use this factory method to create a new instance of*
** this fragment using the provided parameters.*
```

```
** @param param1 Parameter 1.*
** @param param2 Parameter 2.*
** @return A new instance of fragment DirectionFragment.*
*/
**// TODO: Rename and change types and number of parameters
**public static DirectionFragment newInstance(String param1, String param2) {
DirectionFragment fragment = new DirectionFragment();
Bundle args = new Bundle();
args.putString(ARG_PARAM1, param1);
args.putString(ARG_PARAM2, param2);
fragment.setArguments(args);
return fragment;
}
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
if (getArguments() != null) {
mParam1 = getArguments().getString(ARG_PARAM1);
mParam2 = getArguments().getString(ARG_PARAM2);
}
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
// Inflate the layout for this fragment
View v = inflater.inflate(R.layout.fragment_direction, container, false);
button = v.findViewById(R.id.imageButtongotonavbar);
button.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
startActivity(new Intent(getActivity(),NavigationActivity.class));
}
});
return v;
}
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/constraintLayout"
```

```
android:layout_width="match_parent"

android:layout_height="match_parent"

android:background="#CF4A72"

<TextView

android:id="@+id/section_label"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginStart="@dimen/activity_horizontal_margin"

android:layout_marginTop="@dimen/activity_vertical_margin"

android:layout_marginEnd="@dimen/activity_horizontal_margin"

android:layout_marginBottom="@dimen/activity_vertical_margin"

app:layout_constraintLeft_toLeftOf="parent"

app:layout_constraintTop_toTopOf="@+id/constraintLayout"

tools:layout_constraintLeft_creator="1"

tools:layout_constraintTop_creator="1" />

<TextView

android:id="@+id/split_screen"

android:layout_width="340dp"

android:layout_height="wrap_content"

android:layout_marginTop="16dp"

android:background="#FF759F"

android:fontFamily="@font/architects_daughter"

android:text="@string/how_to_use_split_screen_mode_on_an_android_device"

android:textColor="@color/white"

android:textSize="20sp"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.42"
```

```
app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent" />

<TextView

android:id="@+id/step1"

android:layout_width="340dp"

android:layout_height="wrap_content"

android:layout_marginTop="12dp"

android:background="#54E864"

android:fontFamily="@font/architects_daughter"

android:text="@string/step1"

android:textColor="@color/white"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.42"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/split_screen" />

<TextView

android:id="@+id/step2"

android:layout_width="340dp"

android:layout_height="wrap_content"

android:layout_marginTop="84dp"

android:background="#54E864"

android:fontFamily="@font/architects_daughter"

android:text="@string/step3"

android:textColor="@color/white"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.42"

app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="@+id/step1" />
<TextView
android:id="@+id/step4"
android:layout_width="340dp"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"
android:background="#54E864"
android:fontFamily="@font/architects_daughter"
android:text="@string/step4"
android:textColor="@color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.422"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/step2" />
<TextView
android:id="@+id/step5"
android:layout_width="340dp"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"
android:background="#54E864"
android:fontFamily="@font/architects_daughter"
android:text="@string/step55"
android:textColor="@color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.422"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/step4" />
```

```
<TextView

android:id="@+id/step6"

android:layout_width="340dp"

android:layout_height="wrap_content"

android:layout_marginTop="8dp"

android:background="#54E864"

android:fontFamily="@font/architects_daughter"
```

NAVIGATION ACTIVITY

```
package com.appcodewithprachi.speech2isl;

import android.os.Bundle;

import android.view.Menu;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.drawerlayout.widget.DrawerLayout;

import androidx.navigation.NavController;

import androidx.navigation.Navigation;

import androidx.navigation.ui.AppBarConfiguration;

import androidx.navigation.ui.NavigationUI;

import com.google.android.gms.auth.api.signin.GoogleSignInClient;

import com.google.android.material.navigation.NavigationView;

public class NavigationActivity extends AppCompatActivity {

private AppBarConfiguration mAppBarConfiguration;

GoogleSignInClient mGoogleSignInClient;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_navigation);

Toolbar toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

DrawerLayout drawer = findViewById(R.id.drawer_layout);

NavigationView navigationView = findViewById(R.id.nav_view);

// Passing each menu ID as a set of Ids because each

// menu should be considered as top level destinations.

mAppBarConfiguration = new AppBarConfiguration.Builder(

R.id.nav_home, R.id.nav_books, R.id.nav_makenotes)

.setDrawerLayout(drawer)

.build();

NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);

NavigationUI.setupActionBarWithNavController(this, navController, mAppBarConfiguration);

NavigationUI.setupWithNavController(navigationView, navController);

}

//public void OpenSettings(View view) {

//  Intent intent = new Intent(this,SettingsActivity.class);

//startActivity(intent);

//}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.navigation, menu);

return true;

}

@Override

public boolean onSupportNavigateUp() {
```

```
NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
return NavigationUI.navigateUp(navController, mAppBarConfiguration)
|| super.onSupportNavigateUp();
}
}
```

XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:openDrawer="start">
<include
layout="@layout/app_bar_main"
android:layout_width="match_parent"
android:layout_height="match_parent" />
<com.google.android.material.navigation.NavigationView
android:id="@+id/nav_view"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_gravity="start"
android:fitsSystemWindows="true"
app:headerLayout="@layout/nav_header_main"
app:menu="@menu/activity_main_drawer" />
```

```
</androidx.drawerlayout.widget.DrawerLayout>
```

HOME FRAGMENT JAVA

```
package com.appcodewithprachi.speech2isl.ui.home;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;
import com.appcodewithprachi.speech2isl.R;
public class HomeFragment extends Fragment {
private HomeViewModel homeViewModel;
public View onCreateView(@NonNull LayoutInflater inflater,
ViewGroup container, Bundle savedInstanceState) {
homeViewModel =
ViewModelProviders.of(this).get(HomeViewModel.class);
View root = inflater.inflate(R.layout.fragment_home, container, false);
final TextView textView = root.findViewById(R.id.text_home);
homeViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
@Override
public void onChanged(@Nullable String s) {
textView.setText(s);
}
});
return root;
}
}
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#F55887"
tools:context=".ui.home.HomeFragment">
<TextView
android:id="@+id/text_home"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:textAlignment="center"
android:textSize="20sp"
android:text="Home"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MAKENOTES FRAGMENT

```
package com.appcodewithprachi.speech2isl.ui.makenotes;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;
import com.appcodewithprachi.speech2isl.R;
public class MakenotesFragment extends Fragment {
private MakenotesViewModel makenotesViewModel;
public View onCreateView(@NonNull LayoutInflater inflater,
ViewGroup container, Bundle savedInstanceState) {
makenotesViewModel =
ViewModelProviders.of(this).get(MakenotesViewModel.class);
View root = inflater.inflate(R.layout.fragment_makenotes, container, false);
final TextView textView = root.findViewById(R.id.text_makenotes);
makenotesViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
@Override
public void onChanged(@Nullable String s) {
textView.setText(s);
}
});
return root;
}
}
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#F55887"
tools:context=".ui.makenotes.MakenotesFragment">
<TextView
```

```
android:id="@+id/text_makenotes"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginStart="8dp"

android:layout_marginTop="8dp"

android:layout_marginEnd="8dp"

android:text="Make Notes"

android:textAlignment="center"

android:textSize="20sp"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.068" />

<ImageView

android:id="@+id/imageNotes"

android:layout_width="318dp"

android:layout_height="230dp"

android:layout_marginTop="32dp"

android:src="@drawable/notesimages"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.494"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/text_makenotes" />

<TextView

android:id="@+id/textView4"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Nothing to show here right now...... "

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.498"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageNotes"
app:layout_constraintVertical_bias="0.233" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Nav_header_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="@dimen/nav_header_height"
android:background="#199C28"
android:gravity="bottom"
android:orientation="vertical"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:theme="@style/ThemeOverlay.AppCompat.Dark">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"
android:fontFamily="@font/architects_daughter"
android:paddingTop="@dimen/nav_header_vertical_spacing"
android:text="S2ISL"
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView" />
<ImageView
```

```
android:id="@+id/imageView"
android:layout_width="98dp"
android:layout_height="73dp"
android:layout_marginTop="24dp"
android:background="@color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/logoisl" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

App_bar_mai

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".NavigationActivity">
<com.google.android.material.appbar.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">
<androidx.appcompat.widget.Toolbar
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="#199C28"
app:popupTheme="@style/AppTheme.PopupOverlay" />
</com.google.android.material.appbar.AppBarLayout>
<include layout="@layout/content_main" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Activity_main_drawer

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
tools:showIn="navigation_view">
<group android:checkableBehavior="single">
<item
android:id="@+id/nav_home"
android:icon="@drawable/ic_baseline_home_24"
android:title="@string/menu_home" />
<item
android:id="@+id/nav_books"
android:icon="@drawable/ic_baseline_notes_24"
android:title="Books" />
<item
android:id="@+id/nav_makenotes"
android:icon="@drawable/ic_baseline_library_books_24"
android:title="Make Notes" />
</group>
</menu>
```

Navigation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto">
<item
android:id="@+id/action_profile"
android:orderInCategory="100"
android:title="Profile"
app:showAsAction="never" />
<item
android:id="@+id/action_settings"
android:orderInCategory="100"
```

```
android:title="@string/action_settings"
app:showAsAction="never" />
</menu>
```

Module 2

CONVERTER JAVA

```
package com.appcodewithprachi.speech2isl;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.MediaController;
import android.widget.VideoView;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import java.util.ArrayList;
import static android.app.Activity.RESULT_OK;
public class ConverterrFragment extends Fragment {
//ImageButton speechButton;
EditText speechText;
private static final int RECOGNIZER_RESULT = 1;
// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
// TODO: Rename and change types of parameters
private String mParam1;

private String mParam2;

public ConverterrFragment() {

// Required empty public constructor

}

/**

** Use this factory method to create a new instance of*

** this fragment using the provided parameters.*

** @param param1 Parameter 1.*

** @param param2 Parameter 2.*

** @return A new instance of fragment ConverterrFragment.*

*/

Uri video = Uri.parse("android.resource://" + getContext().getPackageName() + "/" + R.raw.onetoninesignlangvideo);

signlangvideo.setVideoURI(video);

//signlangvideo.start();

// Inflate the layout for this fragment

ImageButton speechButton = (ImageButton) root.findViewById(R.id.imageButtonspeech);

EditText speechText = (EditText) root.findViewById(R.id.text);

speechButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent speechIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

speechIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

speechIntent.putExtra(RecognizerIntent.EXTRA_PROMPT,"Speech to ISL");

startActivityForResult(speechIntent, RECOGNIZER_RESULT);

}

});

return root;

}

// from here the text will be transfered to a deep learning model which translate text to Indian sign language
```

```
public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
if (requestCode == RECOGNIZER_RESULT && resultCode == RESULT_OK) {
ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
speechText.setText(matches.get(0).toString());
}
super.onActivityResult(requestCode, resultCode, data);
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#CF4A72"
tools:context=".ConverterrFragment">
<!-- TODO: Update blank fragment layout -->
<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="@font/architects_daughter"
android:text="Translator"
android:textColor="@color/white"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.036" />
<EditText
android:id="@+id/text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:layout_marginTop="16dp"
android:fontFamily="@font/architects_daughter"
android:hint="Speech to Indian Sign Language Translation"
android:textColor="@color/white"
android:textColorHint="@color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageButtonspeech" />
<VideoView
android:id="@+id/signlanguageview"
android:layout_width="355dp"
android:layout_height="364dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/text" />
<ImageButton
android:id="@+id/imageButtonspeech"
android:layout_width="226dp"
android:layout_height="185dp"
android:layout_marginTop="52dp"
android:src="@drawable/imagemicrophone"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Module 3

```
import os, sys
from keras.models import Model
from keras.layers import Input, LSTM, GRU, Dense, Embedding
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
import numpy as np
import matplotlib.pyplot as plt
BATCH_SIZE = 64
EPOCHS = 20
LSTM_NODES =256
NUM_SENTENCES = 50
MAX_SENTENCE_LENGTH = 10
MAX_NUM_WORDS = 1000
EMBEDDING_SIZE = 100
```

Data Preparation and Processing

```
input_sentences = []
output_sentences = []
output_sentences_inputs = []
count = 0
for line in open(r'/content/numerical.txt',encoding='latin-1'):
count += 1
if count > NUM_SENTENCES:
break
if '\n' not in line:
continue
input_sentence, output = line.rstrip().split('\n')
output_sentence = output + ' <eos>'
output_sentence_input = '<sos> ' + output
input_sentences.append(input_sentence)
output_sentences.append(output_sentence)
output_sentences_inputs.append(output_sentence_input)
print("num samples input:", len(input_sentences))
print("num samples output:", len(output_sentences))
print("num samples output input:", len(output_sentences_inputs))
```

Tokenization and Padding

```
input_tokenizer = Tokenizer(num_words=MAX_NUM_WORDS)
input_tokenizer.fit_on_texts(input_sentences)
input_integer_seq = input_tokenizer.texts_to_sequences(input_sentences)
word2idx_inputs = input_tokenizer.word_index
print('Total unique words in the input: %s' % len(word2idx_inputs))
max_input_len = max((len(sen),default=None for sen in input_integer_seq)
print("Length of longest sentence in input: %g" % max_input_len)
output_tokenizer = Tokenizer(num_words=MAX_NUM_WORDS, filters=")
output_tokenizer.fit_on_texts(output_sentences + output_sentences_inputs)
output_integer_seq = output_tokenizer.texts_to_sequences(output_sentences)
output_input_integer_seq = output_tokenizer.texts_to_sequences(output_sentences_inputs)
word2idx_outputs = output_tokenizer.word_index
print('Total unique words in the output: %s' % len(word2idx_outputs))
num_words_output = len(word2idx_outputs) + 1
max_out_len = max(len(sen) for sen in output_integer_seq)
print("Length of longest sentence in the output: %g" % max_out_len)
```

Word Embeddings

```
encoder_input_sequences = pad_sequences(input_integer_seq, maxlen=max_input_len)
print("encoder_input_sequences.shape:", encoder_input_sequences.shape)
print("encoder_input_sequences[172]:", encoder_input_sequences[172])
decoder_input_sequences = pad_sequences(output_input_integer_seq, maxlen=max_out_len, padding='post')
print("decoder_input_sequences.shape:", decoder_input_sequences.shape)
print("decoder_input_sequences[172]:", decoder_input_sequences[172])
```

TF-IDF Vectorizer

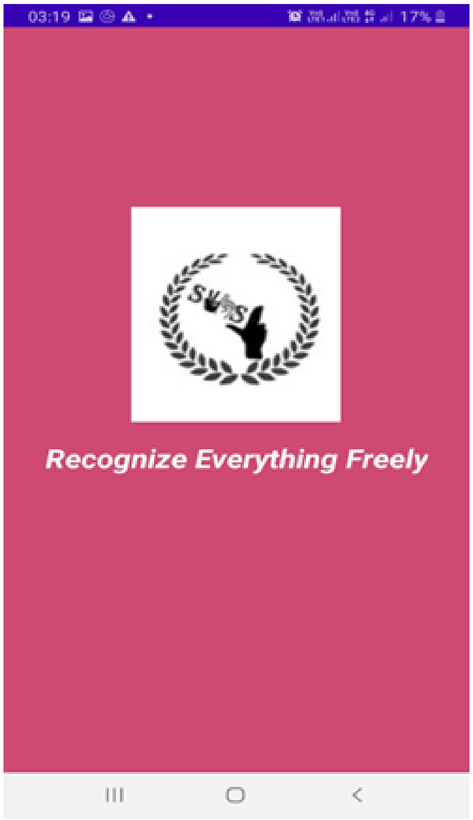
```
from sklearn.feature_extraction.text import TfidfVectorizer
sentences = [
'This is the study material for this topic .',
'you have to learn from the material given',
'you can make notes from the study material',
'There will be an exam held of the topic  '
]
vectorizer = TfidfVectorizer(norm = False, smooth_idf = False)
sentence_vectors = vectorizer.fit_transform(sentences)
print(sentence_vectors.toarray())
decoder_targets_one_hot = np.zeros((
len(input_sentences),
max_out_len,
num_words_output
),
dtype='float32'
)
```

Creating the model

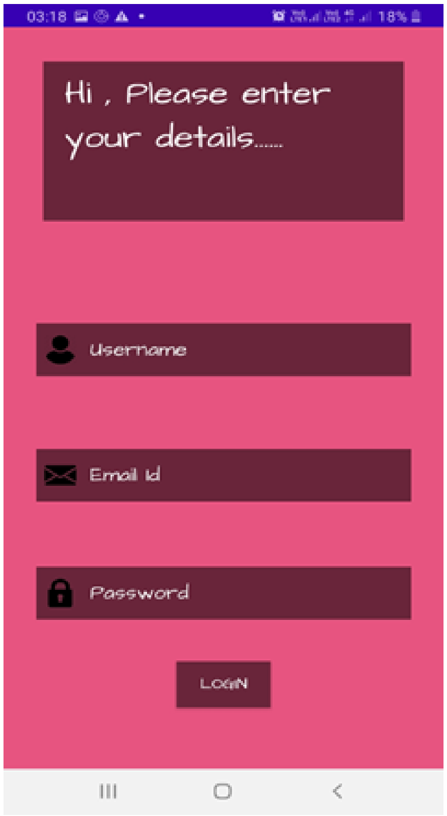
```
decoder_targets_one_hot.shape
for i, d in enumerate(decoder_output_sequences):
for t, word in enumerate(d):
decoder_targets_one_hot[i, t, word] = 1
encoder_inputs_placeholder = Input(shape=(max_input_len,))
x = embedding_layer(encoder_inputs_placeholder)
encoder = LSTM(LSTM_NODES, return_state=True)
encoder_outputs, h, c = encoder(x)
encoder_states = [h, c]
decoder_inputs_placeholder = Input(shape=(max_out_len,))
decoder_embedding = Embedding(num_words_output, LSTM_NODES)
decoder_inputs_x = decoder_embedding(decoder_inputs_placeholder)
decoder_lstm = LSTM(LSTM_NODES, return_sequences=True, return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs_x, initial_state=encoder_states)
decoder_dense = Dense(num_words_output, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
decoder_inputs_placeholder = Input(shape=(max_out_len,))
decoder_embedding = Embedding(num_words_output, LSTM_NODES)
decoder_inputs_x = decoder_embedding(decoder_inputs_placeholder)
decoder_lstm = LSTM(LSTM_NODES, return_sequences=True, return_state=True)
```

```
decoder_outputs, _, _ = decoder_lstm(decoder_inputs_x, initial_state=encoder_states)

decoder_dense = Dense(num_words_output, activation='softmax')

decoder_outputs = decoder_dense(decoder_outputs)
```

Appendix B



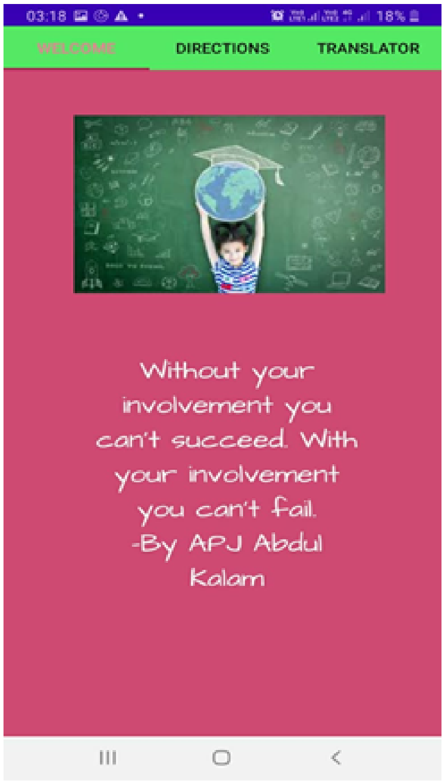
Recognize Everything Freely



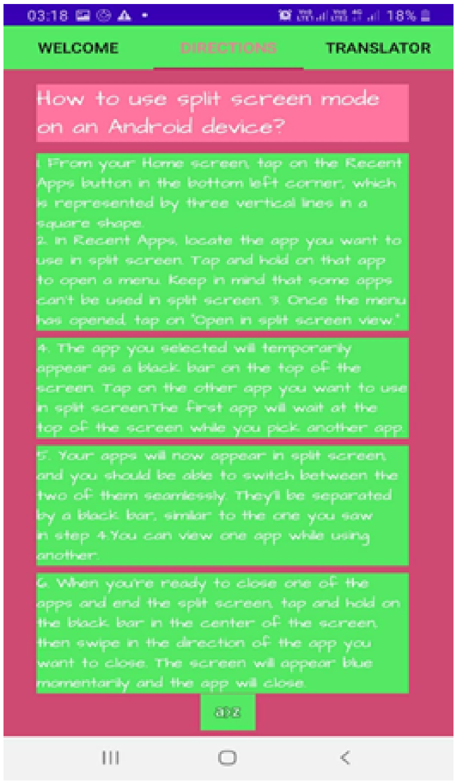
Hi , Please enter your details......

Fig.1

Fig.2



Without your involvement you can't succeed. With your involvement you can't fail.
-By APJ Abdul Kalam



How to use split screen mode on an Android device?

1. From your Home screen, tap on the Recent Apps button in the bottom left corner, which is represented by three vertical lines in a square shape.
2. In Recent Apps, locate the app you want to use in split screen. Tap and hold on that app to open a menu. Keep in mind that some apps can't be used in split screen. 3. Once the menu has opened, tap on 'Open in split screen view.'

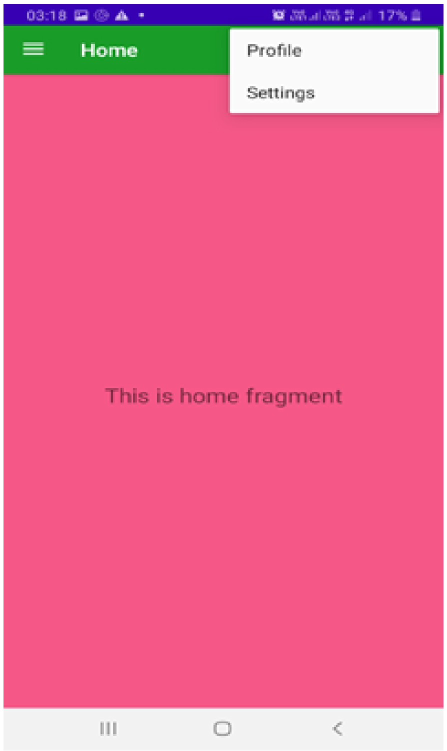
4. The app you selected will temporarily appear as a black bar on the top of the screen. Tap on the other app you want to use in split screenThe first app will wait at the top of the screen while you pick another app.

5. Your apps will now appear in split screen, and you should be able to switch between the two of them seamlessly. They'll be separated by a black bar, similar to the one you saw in step 4.You can view one app while using another.

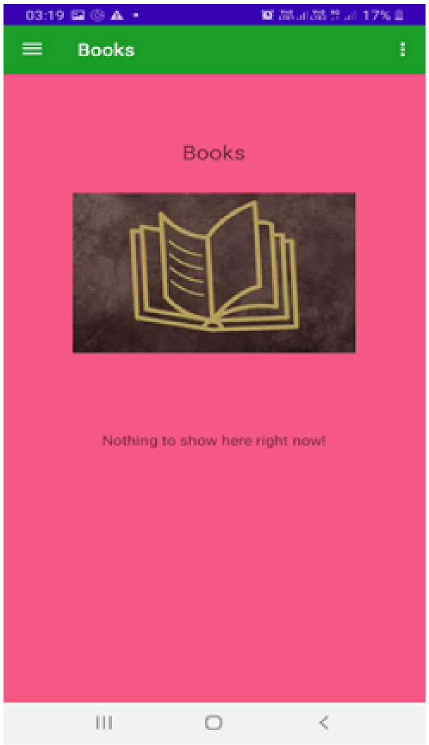
6. When you're ready to close one of the apps and end the split screen, tap and hold on the black bar in the center of the screen, then swipe in the direction of the app you want to close. The screen will appear blue momentarily and the app will close.

Fig.3

Fig.4



This is home fragment

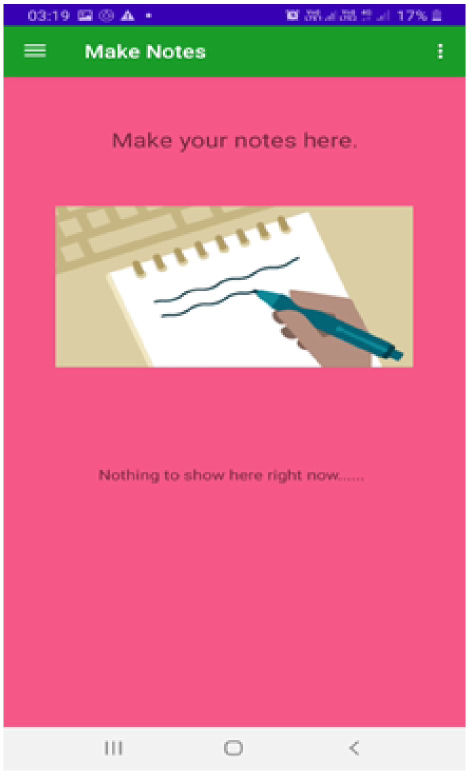


Books

Nothing to show here right now!

Fig.5

Fig.6



Make your notes here.

Nothing to show here right now......

Fig.7

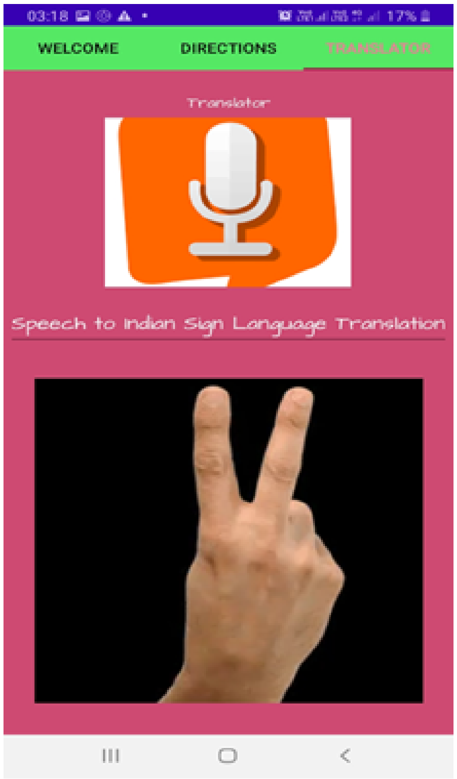


Fig.8

REFERENCES

1. https://www.researchgate.net/publication/221365167_V2S_Voice_to_Sign_Language_Translation_System_for_Malaysian_Deaf_People
2. https://www.govtech.com/magazines/gt/Speech-Recognition-vs-Speaker-Recognition-.html
3. http://www.typoday.in/2016/spk_papers/Nakul_Singal_TypographyDay-2016.pdf
4. https://link.springer.com/chapter/10.1007/978-3-642-05036-7_82
5. https://www.sciencedirect.com/science/article/pii/S1110866516300111
6. http://islrtc.nic.in/sites/default/files/List_of_books_Available_in_Library_(updated)_for_website-converted.pdf
7. http://www.typoday.in/2016/spk_papers/Nakul_Singal_TypographyDay-2016.pdf
8. https://www.researchgate.net/publication/342735487_Introduction_of_Indian_Sign_Language_in_Inclusive_Education
9. http://docshare01.docshare.tips/files/24292/242924330.pdf
10. https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa
11. https://towardsdatascience.com/language-translation-with-rnns-d84d43b40571

12. https://www.researchgate.net/publication/319367587_Machine_translation_using_deep_learning_An_overview
13. https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm
14. https://www.researchgate.net/publication/283983876_Accessible_Options_for_Deaf_People_in_e-Learning_Platforms_Technology_Solutions_for_Sign_Language_Translation
15. https://www.researchgate.net/publication/236634112_Speech_to_Sign_Language_Interpreter_System_SSLIS
16. https://www.researchgate.net/publication/320816463_E-Learning_through_Single_Hand_and_Two_Hand_Sign_Language
17. https://www.researchgate.net/publication/4204715_An_e-Learning_System_for_the_Deaf_people
18. https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/
19. https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd
20. https://www.analyticsvidhya.com/blog/2019/07/learn-build-first-speech-to-text-model-python/
21. https://www.researchgate.net/publication/283184719_Continuous_Indian_Sign_Language_Gesture_Recognition_and_Sentence_Formation/link/57f5d7df08ae91deaa5c8a89/download
22. https://www.researchgate.net/publication/283184719_Continuous_Indian_Sign_Language_Gesture_Recognition_and_Sentence_Formation/link/57f5d7df08ae91deaa5c8a89/download
23. https://www.kdnuggets.com/2020/05/guide-choose-right-machine-learning-algorithm.html
24. https://www.kdnuggets.com/2018/12/six-steps-master-machine-learning-data-preparation.html
25. https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html
26. https://towardsdatascience.com/do-you-know-how-to-choose-the-right-machine-learning-algorithm-among-7-different-types-295d0b0c7f60