**B.E. (CIS) PROJECT REPORT by**

**Ifrah Ishtiaq**

**Department of Computer and Information Systems Engineering**

**NED University of Engg. & Tech., Karachi-75270**

# B.E. (CIS) PROJECT REPORT

**Project Group:**

Ifrah Ishtiaq CS-17132

Ameema Arif CS-17122

Mahrukh Khan CS-17003

Syeda Sara Akif CS-17002

**BATCH:** 2017-18

**Project Advisor(s):**

Ms. Fakhra Aftab (Internal Advisor)

**September** 2021

**Department of Computer and Information Systems Engineering**

**NED University of Engg. & Tech., Karachi-75270**

# ABSTRACT

*The abstract of the report will be just after the Internal Title Page. The abstract should be brief and should be about half a page but certainly not more than one page. 12-point Italic Times New Roman Font with should be used.*

# ACKNOWLEDGEMENTS

The acknowledgement page (if any) should be after the abstract page.

# CONTENTS

**S.No. Page No.**

1. [Introduction 1](#_Toc82564234)

[1.1 Project Background 1](#_Toc82564235)

[1.2 Problem Statement 1](#_Toc82564236)

[1.3 About the Project 2](#_Toc82564237)

[1.4 Related Work 2](#_Toc82564238)

[1.5 Goals and Objectives 3](#_Toc82564239)

1. [Speech Recognition and Translation 5](#_Toc82564241)

[2.1 What is Speech Recognition? 5](#_Toc82564242)

[2.2 Working of Speech Recognition 5](#_Toc82564243)

[2.2.1 Conventional ASR Models 5](#_Toc82564244)

[2.2.2 End-to-end ASR Models 6](#_Toc82564245)

[2.3 Approaches and Types of ASR 7](#_Toc82564246)

[2.4 Performance Evaluation of ASR 8](#_Toc82564247)

[2.4.1 Accuracy and its Improvement 8](#_Toc82564248)

[2.4.2 Latency and its Improvement 9](#_Toc82564249)

[2.5 What is Translation and Real Time Translation? 10](#_Toc82564250)

[2.5.1 Translation 10](#_Toc82564251)

[2.5.2 Real Time Translation 10](#_Toc82564252)

[2.6 Need for Translation: 10](#_Toc82564253)

[2.7 Background of RTT 11](#_Toc82564254)

[2.8 Working of RTT 12](#_Toc82564255)

[2.9 Future of RTT 13](#_Toc82564256)

1. [Face Tracking and Subtitles Display in Real Time 15](#_Toc82564258)

[3.1 Face Detection and Tracking 15](#_Toc82564259)

[3.1.1 Face Detection: 15](#_Toc82564260)

[3.1.2 How Face Detection works 15](#_Toc82564261)

[3.1.3 Face Tracking 16](#_Toc82564262)

[3.1.4 How Face Tracking works 16](#_Toc82564263)

[3.1.5 Firebase ML Kit: 16](#_Toc82564264)

[3.1.6 What is firebase Face detection? 17](#_Toc82564265)

[3.2 Subtitles Display in Real Time 17](#_Toc82564266)

[3.2.1 Fundamentals to Live Subtitles 18](#_Toc82564267)

[3.2.1.1 Open Captions 18](#_Toc82564268)

[3.2.1.2 Subtitles in Same language 18](#_Toc82564269)

[3.2.1.3 Translated Subtitles 19](#_Toc82564270)

[3.2.1.4 Subtitling Constraints 19](#_Toc82564271)

[3.2.2 Approaches 20](#_Toc82564272)

[3.2.2.1 Using Android Studio Only 20](#_Toc82564273)

[3.2.2.2 Using Python with Java 20](#_Toc82564274)

[3.2.2..3 Using Python Only 20](#_Toc82564275)

[3.2.2.4 Using SRT files in Python 21](#_Toc82564276)

[3.2.2.5 Using OpenCV in Python and Java 21](#_Toc82564277)

[3.2.2.6 Using Python and OpenCV in Java 21](#_Toc82564278)

[3.2.2.7 Using OpenCV in Java 21](#_Toc82564279)

[3.2.2.8 Using OpenCV in Python 22](#_Toc82564280)

1. [Android App Development and Integration 23](#_Toc82564282)

[4.1 App Activties 23](#_Toc82564283)

[4.1.1 Main Activity 24](#_Toc82564284)

[4.1.2 Camera Activity 26](#_Toc82564285)

[4.1.3 Gallery Activity 29](#_Toc82564286)

[4.1.4 Settings Activity 30](#_Toc82564287)

[4.1.5 Help Activity 32](#_Toc82564288)

1. [Testing 33](#_Toc82564290)

[5.1 Requirements 33](#_Toc82564291)

[5.2 Application Testing 33](#_Toc82564296)

1. [Achievement (Project Outcome) 34](#_Toc82564298)

6.1 Making Real-time Videos

6.2 Speech Recognition

6.3 Speech to Text

6.4 Text Translation

6.5 Subtitles Display

1. [Future Enhancement 38](#_Toc82564300)

7.1 Reduce delays

7.2 Save Videos Along with Subtitles

7.3 Adding more languages

7.4 Face detection and Speech Detection

7.5 Using Machine Learning

1. [Conclusion 40](#_Toc82564302)

9. References 41

# CHAPTER 1

**Introduction**

The first chapter of the report will be the introduction chapter. The format shown on the first page of this first chapter must be followed in all chapters. The text should be double-line spaced. The start of every new paragraph should be indented with one tab space. The font should be Regular 12-point Times New Roman and fully justified. The sections should be numbered according to the number of Chapter.

* 1. **Section 1**

There can be several sections within every chapter. The start of every section should be indented. The number of sections will depend upon the text. The sections should be numbered according to the number of chapter. For example for chapter 1, the section should be 1.1, 1.2, 1.3 and so on.

All the figures and tables within the text should also be numbered according to chapter number. For example for Chapter number 1 the numbering will be Fig. 1.1, Fig 1.2, Table 1.1, Table 1.2 and so on. When referring to the Figures and Tables in the text, respective numbers should be used.

The references of books, journals or articles used for the preparation of report should appear within the text. The references should use a citation number. For example while referring an article, the number should be used. [1] Similarly while referring a book, the citation number should be used. [2] The details of references should appear in the references section at the end of the report.

* + 1. **Subsection within a section**

There can be subsections within the sections. The format of the subsections should be the same as shown in this example. The subsections should also be numbered according to the number of chapter. For example for chapter 1, section 1.1, subsections should be numbered as 1.1.1, 1.1.2 and so on.

The format of the second page of every chapter will be like this page.

# CHAPTER 2

**Name of Chapter 2**

The format of the first page of the second chapter will be like this page.

# CHAPTER 3

**Methodology**

Research and practical work on generating Live Subtitles is being carried out from a long time but still there is a lot of room for improvement, accuracy and innovation in this domain. Speech recognition technique is being used for live subtitling since 2005, or may be before that. Live subtitling technology is still not very popular world-wide as it is hard to develop. However, Countries like U.S and U.K are using this technology efficiently from a long time.

Live subtitles are introduced as a ceaseless progression of text as individuals talk. Live subtitles usually are not very accurate as there is not enough time for correction between speech recognition and its conversion into subtitles. Live subtitles will ordinarily slack the sound by a few seconds because of the inborn postponement in translating, encoding, and displaying the captions. Live captions also has some typographic blunders or mishearing of the verbally expressed words, with no time available for correction before transmission.

Since, live subtitles are not very accurate there is still a lot of improvement needed in this domain. Some technologies can be developed that can understand different accents and provide efficient hearing to reduce typographic blunders. The lag between the speech and display of subtitles is still a big challenge that open doors to a wide research domain which can devise some efficient solution for this. Besides these challenges live subtitling is still a very useful technique which can make the life of common people easier but still it is not used widely. Through our App Live Subtitles we want to take this technology to the common people, so that they can use it in an easy and convenient way.

**3.1 Fundamentals to Live Subtitles**

Subtitling is the need of this modern era. There’s a common saying that more screens means more subtitles. That is said because now everyone tries to increase the reach of their content and to make something understandable internationally language barrier should be removed and this job is done efficiently through subtitling.

**3.1.1 Open Captions**

Open captions are part of the video and cannot be turned off and on [1]. Means these captions/subtitles will be displayed throughout the video as they are recorded as part of the video. Since the main objective of live subtitles is to make communication easier for people with hearing impairment and reduce the language barrier so open captions will fulfill these requirements.

**3.1.2 Subtitles in Same language**

Usually referred to as SLS(Same Language Subtitles). By same language we mean that the subtitles for a video would be generated in the language used in the video. This type of subtitles are useful for people with hearing impairments as they don’t have to be dependent on the sign language translation and can easily read what someone is saying. This also keeps them connected to people who doesn’t know sign language. However it cannot be said that they are only useful for people with hearing impairment, because many times due to the difference of accent and pronunciation people are unable to understand what somebody has said although they can hear them perfectly.

**3.1.3 Translated Subtitles**

Translated subtitles mean that the subtitles would be generated in the language selected by the user, regardless of the language used in the video. This type of subtitling is very useful for increasing the reach of content and to reduce the language barrier. In the past, videos that were meant to be reached by people internationally had to be dubbed into different languages in order to make them understandable internationally. Dubbing is a time taking process as first it requires content translation from a proficient speaker of both the languages(The language used in the video and the language in which it is to be translated), then it has to be dubbed in that language again by some proficient speakers of that language. So this process needs a lot of resources and time. On the other hand, subtitling only needs translation of the content and its placement in the video. Nowadays, such artificially intelligent softwares are available which only needs the content and they place it in the video by themselves and some softwares can even provide automatic translation of that content.

**3.1.4 Subtitling Constraints**

Subtitling constraints mean the limitations in the process of subtitling. Translated subtitles are not usually the exact translation of the content spoken in the video. Translated subtitles usually follow word to word translation technique. There can be some phrases, narrations, proverbs or poetry spoken in the video which when translated for subtitling doesn’t provides the exact meaning of it, however it hints at the overall meaning of it. This usually happens because nowadays people mostly make use of language translator softwares which employs word to word language translation.

**3.2 Methodology**

The front-end of this app is built on Android Studio 4.1.2. Android Studio is one of the most popular platform used to build android applications, as it is open-source and free to use. We kept the front-end of our application simple in order to make it user friendly. We have used the Android version 4.1 Jelly Bean API 16 so that our app can run on approximately 99.8% of the devices. Since, everyone doesn’t have the latest android version because some people don’t find it useful to upgrade their android version quickly or some people don’t know much about android versions. So, building an app using the latest android version can reduce its reachability.

An Android activity is one screen of the Android app's user interface. In that way an Android activity is very similar to windows in a desktop application. An Android app may contain one or more activities, meaning one or more screens. Every activity also have an XML file which can be used to design the activity and a Java file which is used to define the functionality. The Android app starts by showing the main activity, and from there the app may make it possible to open additional activities [2]. Every activity in an android app goes through some states during its life in the app this process is called Activity Life Cycle and it is illustrated in Fig. 3.1

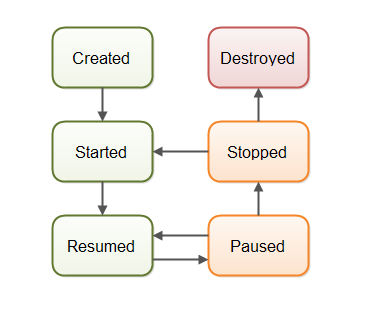


Fig. 3.0.1 Android Activity Life Cycle

Our Android app is consist of 5 activities namely:

* Main Activity
* Camera Activity
* Gallery Activity
* Settings Activity
* Help Activity

**3.2.1 Main Activity**

Main Activity is the one which will appear right after when user opens the application. Its function is to provide a link to other activities.

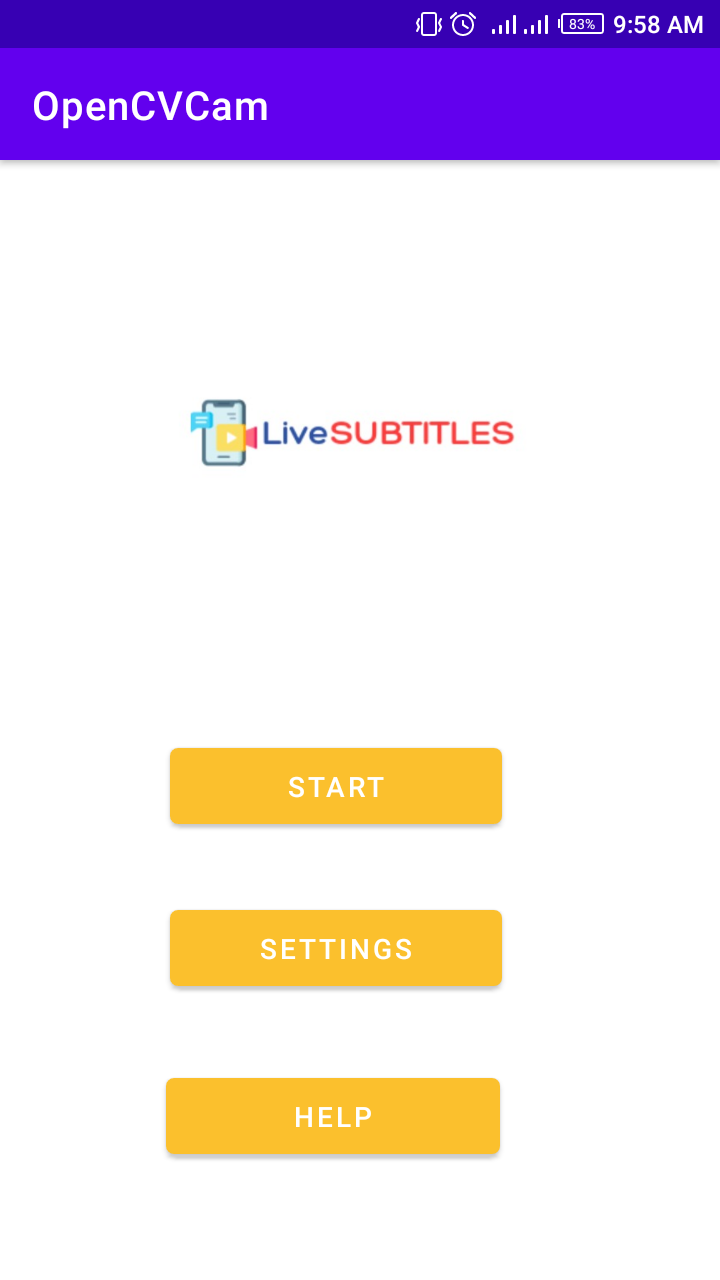


Fig. 3.2 Main Activity

**Front-End**

In Fig. 3.2 we can see the main activity. It has three buttons which can take the user to other activities. This activity is designed using a Relative layout within a Frame layout which has three Buttons and one ImageView for displaying the app logo. These things can either be coded or can be designed by using drag and drop functionality of Android Studio.

**Back-End**

It has a simple functionality to open the camera, settings or help activity when their respective button is clicked. Each of these buttons is first defined using findViewById function and then has a onClick method to take user to that specific activity every time when the button is clicked.

**3.2.2 Camera Activity**

When the Start button is clicked Camera Activity is launched. It provides many functions and the basic functionality offered by the app is provided in this activity.

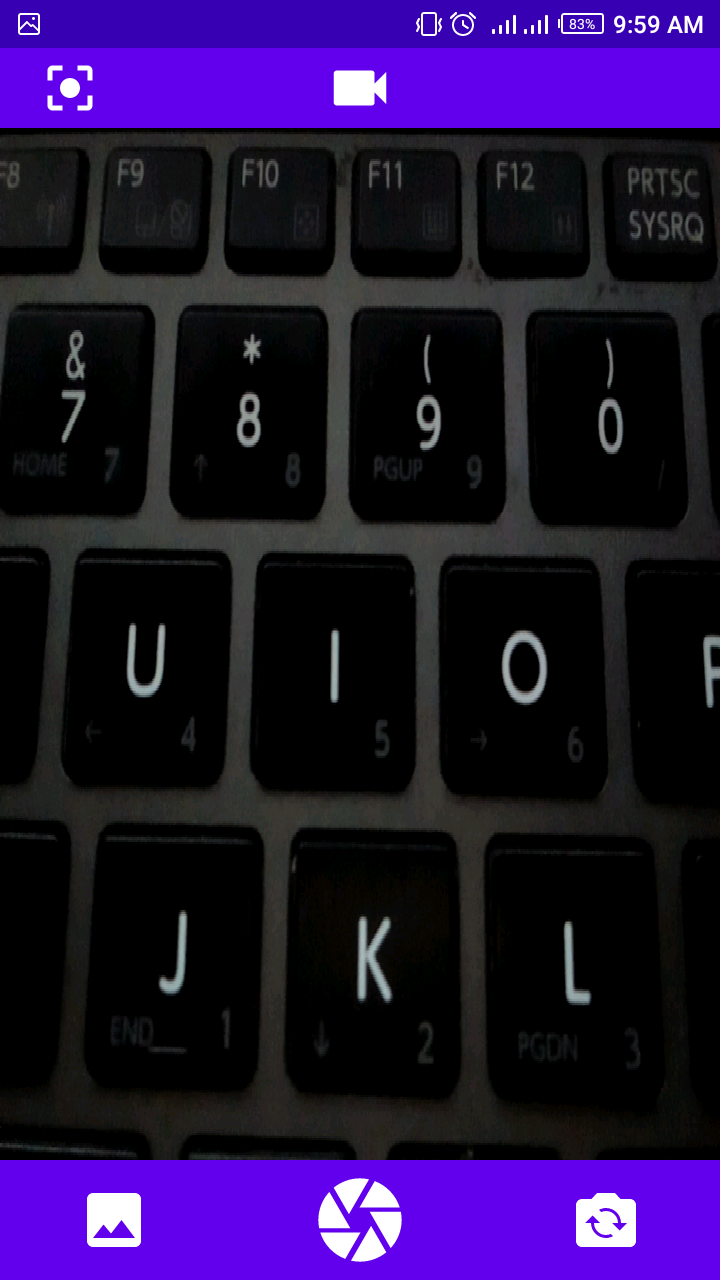


Fig. 3.3 Camera Activity

**Front-End**

We can see in Fig. 3.3 that there are some icons on the top and bottom bar of this activity, these icons provide different functionalities. Camera activity is designed using Frame layout which is consist of a JavaCameraView and two linear layouts one for top bar and one for bottom bar. Top linear layout contains two ImageView’s one for video camera button and the other for change image resolution button. Bottom linear layout contains three ImageView’s one to view gallery, one for camera button and one for flip camera button. The media is played in JavaCameraView.

**Back-End**

The camera function is implemented the OpenCV module android-sdk version 3.4.13 so first its status is checked whether it is connected or not. Then all required permissions are checked like permission to use camera, microphone, storage etc. Then the flip camera button is defined and its method is implemented to swap camera between back and front camera whenever this button is pressed. Method for gallery button is implemented to open gallery activity. By default Picture mode is set that means when the camera activity opens picture mode is enabled so user can take pictures from either back or front camera. This is an additional feature provided in the app. Video mode is enabled when the user will click on the Video camera button. In video mode the camera button will be replaced by a circle button.

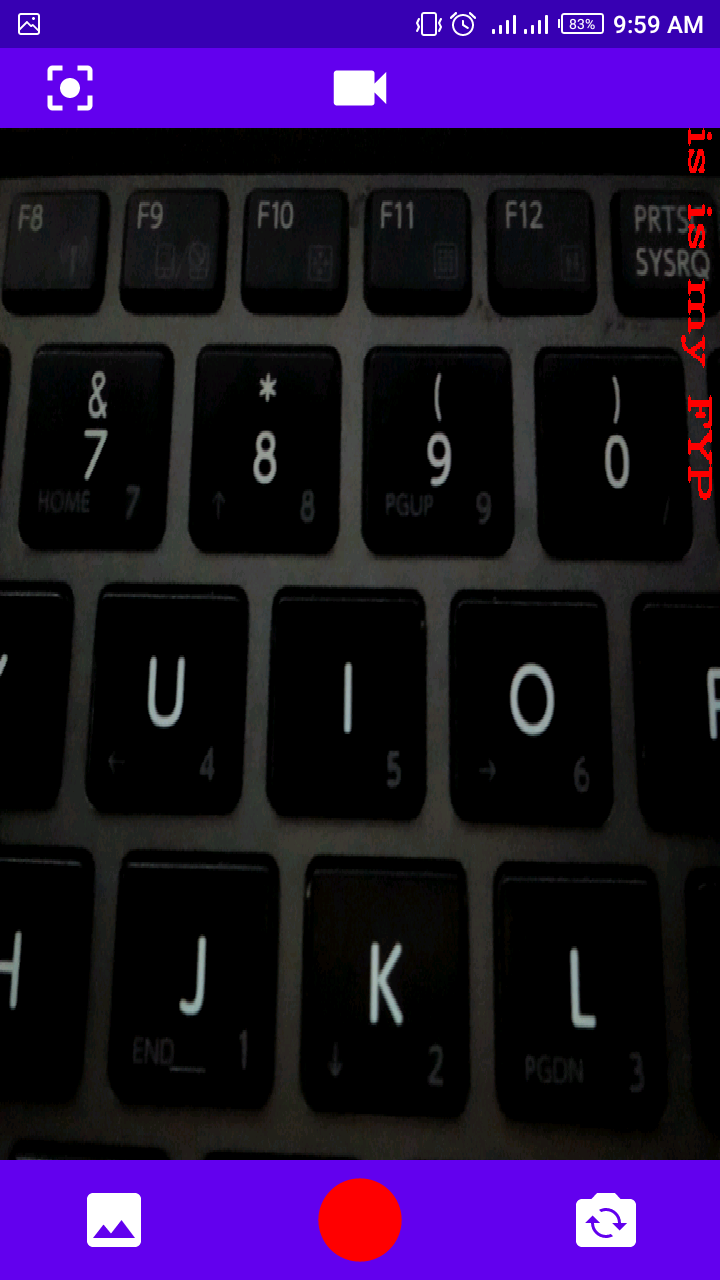


Fig. 3.4 Capturing Video

As it can be seen in Fig. 3.4 that when the user start capturing video the circle turns red and when the user stops capturing video the circle button turns white. All this is implemented in the onTouch method of video camera button. Functionalities to save either picture or video in user’s external storage are implemented in the take picture button onTouch method. Then the following methods are implemented:

* onResume( )
* onPause( )
* onDestroy( )
* onCameraViewStarted( )
* onCameraViewStopped( )

The speech recognizer class will also be used in this Java file to recognize speech, convert it into text and display that text as subtitles over the video. The text that can be seen in Fig. 3.4 on the screen is overlaid on the screen using the ImgProc.putText() function of OpenCV in the onCameraFrame method. Some of the approaches to do speech to text conversion and displaying that text as subtitles on the video will be discussed in next section.

**3.2.3 Gallery Activity**

This activity will appear when user clicks the gallery icon placed on the bottom left corner of the camera activity.

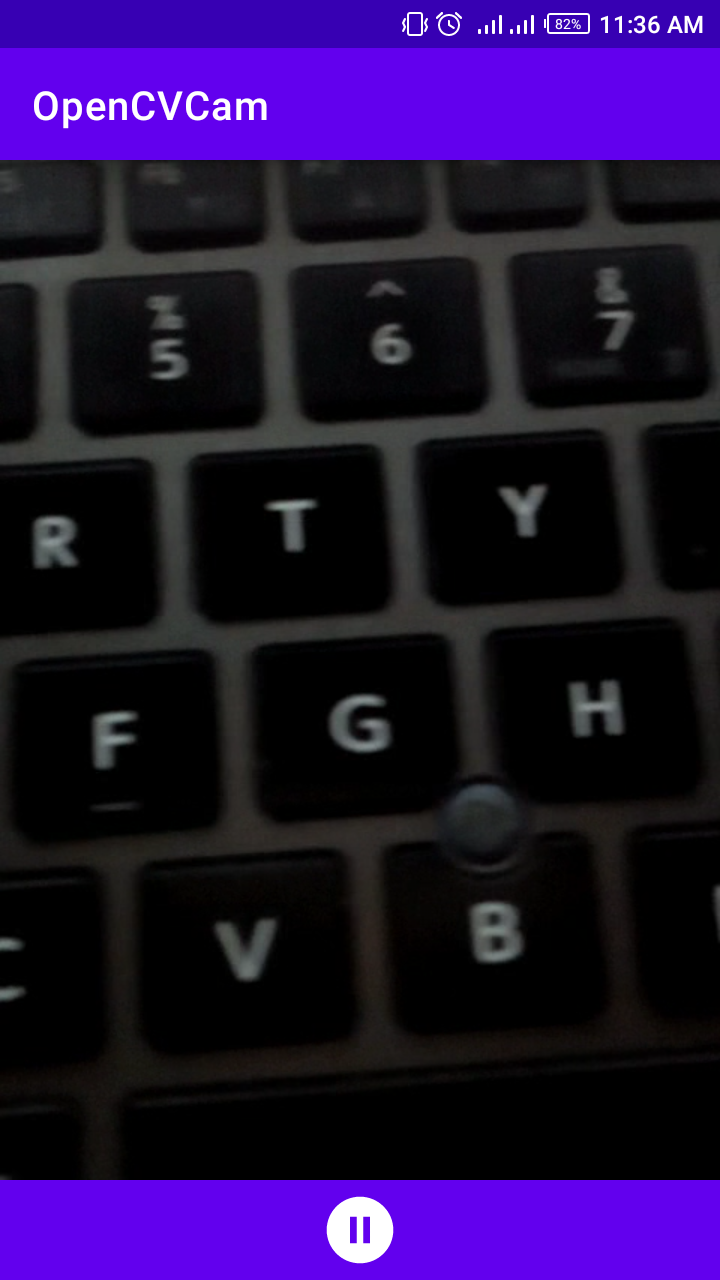


Fig. 3.5 Gallery Activity

**Front-End**

This is a simple activity which shows captured media in the order of oldest to latest. For video on the bottom it has a bar which has an icon in the center which keep changing into Play, Pause or Replay buttons according to the situation.

**Back-End**

It fetches media files from the user’s external storage using the createFileArray method and allows user to swipe media file right and left with viewPager class.

**3.2.4 Settings Activity**

This activity will appear when user clicks the settings button on the main activity. This activity let the user to choose the language in which he wants to view the subtitles.

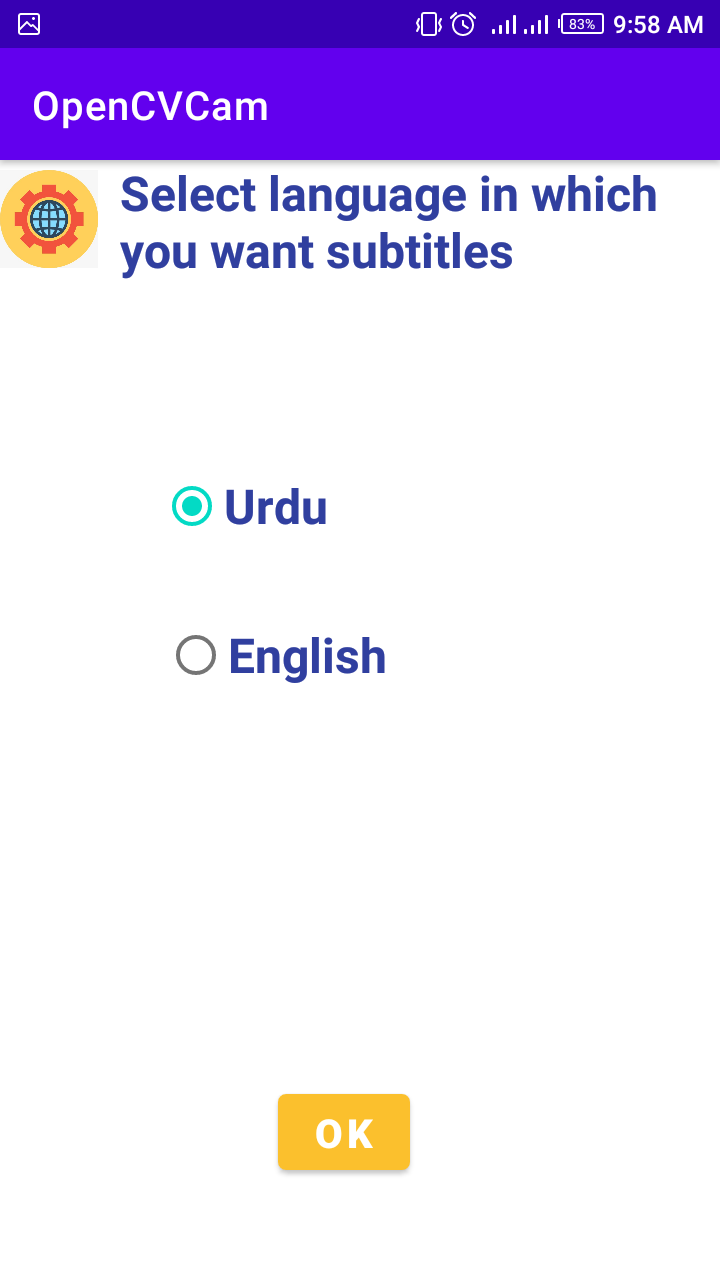


Fig. 3.6 Settings Activity

**Front-End**

Activity in Fig. 3.6 is designed using a Relative layout. It has an ImageView to display the settings symbol, a TextView for description, a button and a RadioGroup. The RadioGroup has two RadioButton’s one for English and one for Urdu, only one radio button can be selected at once.

**Back-End**

It includes the method to pass the input from radio buttons to the camera activity’s speech recognition class. It also includes the onClick method for Ok button to take the user back to the main activity every time when it is clicked.

**3.2.5 Help Activity**

This activity will appear when user clicks the help button on the main activity. This activity will provide user some guidelines on how to use this app.



Fig. 3.7 Help Activity

**Front-End**

This activity is designed using Relative layout which is consist of an ImageView for displaying help symbol, two TextView’s one for heading and one for guidelines and a button to take the user back to the main activity.

**Back-End**

This a simple activity. It includes onClick method for back button to return to the main activity whenever this button is pressed.

**3.3 Approaches**

In this section we will discuss some approaches that are used for speech to text conversion and then putting that text as subtitles on the video.

**3.3.1 Using Android Studio Only**

Doing everything in android studio using Java alone and not using any external module or software.

**Result:** Couldn't find any function in Java to extract audio from time to time to convert it to text and put it on the screen as subtitles.

**3.3.2 Using Python with Java**

Making video using Java in android studio and passing that video to Python function as a parameter to do the further processing. This is done by using Chaquopy which is a platform to execute Python scripts in android studio.

**Result:** Couldn't find any function in Java to pass frames one by one to Python function for processing.

**3.3.3 Using Python Only**

When the button is clicked calling Python function to do everything from capturing video, extracting audio, converting speech to text to displaying that text in the form of subtitles.

**Result**: This approach build successfully but App crashes on run time.

**3**.**3.4 Using SRT files in Python**

Write the recognized audio in an SRT (SubRip Text) file time to time and read that file to put subtitles on the video, using Python. Note that SRT files are subtitles files that contain Block no. , starting and ending time for the subtitle to be displayed on the screen and the subtitle to be displayed within that time.

**Result:** Reading and writing the SRT file simultaneously causes a lot of delay and runtime errors.

**3.3.5 Using OpenCV in Python and Java**

Make video in android studio using OpenCV module and pass that video as a parameter to a Python function to display subtitles using OpenCV. Note that we are distributing the tasks between Java and Python because there are some functions of OpenCV that are available in Python but not in Java.

**Result:** App keeps crashing when we start recording video.

**3.3.6 Using Python and OpenCV in Java**

Making video in android studio using OpenCV get the recognized audio from python function using speech recognition module and put the text returned from python function on the screen using OpenCV putText function in android studio.

**Result:** App keeps crashing when we start recording video.

**3.3.7 Using OpenCV in Java**

Making video in android studio using OpenCV module and putting a hardcoded text on it using OpenCV putText function.

**Result:** App working fine and displaying the overlayed text on the screen when we start recording video.

**3.3.8 Using OpenCV in Python**

Making video using OpenCV in Python and recognizing audio using speech recognition module and putting that recognized text on screen using OpenCV.

**Result:** Things working fine but slow.

# REFERENCES

1. "Captioning Basics", *Cryrop.org*, 2021. [Online]. Available: https://cryrop.org/documents/Teacher-Resources/Technology%20Integration/ADA%20Resources/Designing%20Accessible%20Multimedia/4.3%20Captioning%20Basics.pdf.
2. J. Jenkov, "Android Activity", *Tutorials.jenkov.com*, 2021. [Online]. Available: http://tutorials.jenkov.com/android/activity.html#:~:text=An%20Android%20activity%20is%20one,meaning%20one%20or%20more%20screens. [Accessed: 10- Sep- 2021].

**6.1 Making Real-time Videos**

**6.2 Speech Recognition**

**6.3 Speech to Text**

**6.4 Text Translation**

**6.5 Subtitles Display**

**7.1 Reduce delays**

**7.2 Save Videos Along with Subtitles**

**7.3 Adding more languages**

**7.4 Face detection and Speech Detection**

**7.5 Using Machine Learning**