

AI BASED PICTURE TRANSLATION APP

Submitted in partial fulfilment of requirements for the award of
Bachelor of Engineering
in
Electronics and Communication Engineering
by
SHAIK NAINAZ (39130420)



DEPERTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

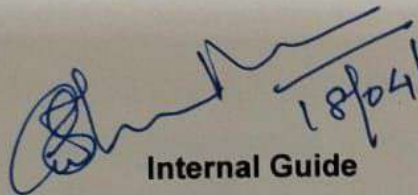
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

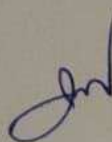
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Shaik Nainaz (39130420) who carried out the project entitled "AI BASED PICTURE TRANSLATION APP" under our supervision from Aug 2022 to April 2023.

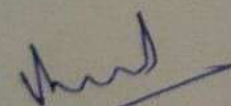

18/04/2023
Internal Guide

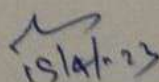
Dr.M.S. GODWIN PREMI, M.E., Ph.D.


Head of the Department

Dr.T. RAVI M.E., Ph.D.

Submitted for Viva voce Examination held on 19/04/2023


Internal Examiner

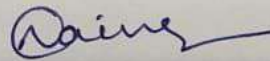

19/04/23
External Examiner

DECLARATION

I, **SHAIK NAINAZ (39130420)** hereby declare that the Project Report entitled "**AI BASED PICTURE TRANSLATION APP**" done by us under the guidance of **Dr. M.S. GODWIN PREMI, M.E, Ph.D.**, Professor, Department of Electronics and Communication Engineering is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

Chennai

18-04-2023



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. NANDHITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr.T. RAVI M.E., Ph.D.** Head of the Department, Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.M.S. GODWIN PREMI, M.E, Ph.D.**, Professor, Department of Electronics and Communication Engineering for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project

ABSTRACT

Trying to grasp and understand the universal nuances of multiple languages is a whole other game. To date, the only way to translate has been for 'human translators' to study many languages and attempt to interpret and bridge language boundaries for various purposes. The complexities of the characters may certainly put even the most modern technology to the test. Translation applications may be lifesavers when you are travelling and are surrounded by people who speak or read a language you do not understand. The system uses Object Detection from Google Cloud vision API to recognize the object and it is a part of Machine Learning (ML). Similarly, for Text, we are using Text Detection from Google Cloud vision API to recognize the Text, and it is also part of ML. It translates the text or object into a preferred language. For instance, if a user selects a computer image and Marathi as the preferred language, Google Translation API will translate the recognized image and display the message/output in Marathi. Another option is available here; the same user must pick the image on which text is accessible, and that text or message will be transformed into the specified language. It will be beneficial if you do not understand any other language text or message, as you will be able to convert it into your understandable language.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTARCT	v
	LIST OF FIGURES	vii
1	Introduction	1
	1.1 Picture Translation	1
	1.2 Picture Translation using AI App	2
	1.3 AI Technique for Translation	2
2	Literature Review	4
	2.1 Language Translation Platform	4
	2.2 An Android-based language translator	4
	2.3 Language Translator Application using Fire Base	5
	2.4 Sign Language Translator for Mobile Platforms	5
	2.5 English Text to Multilingual Speech Translator	6
	2.6 Literature Summary	6
	2.7 Problem Definition	7
	2.8 Objective of Project	7
3	Existing System	8
	3.1 Artificial Intelligence Based Language Translation Platform	8
	3.2 Block Diagram	9
4	Requirement Analysis	11
	4.1 Software Used for the Proposed System	12
	4.2 Explanation of Android Studio	12
5	Description of Proposed System	18
	5.1 Architecture Model of Proposed System	18
	5.2 Explanation of Data base	23
	5.3 Language used for proposed system	24
	5.4 Implementation Details	25
	5.5 Structure of the project	26
6	RESULTS AND DISCUSSION	33
7	SUMMARY AND CONCLUSION	34

7.1	Conclusion	
7.2	Future Work	34
REFERENCES		
APPENDIX		
A.	SOURCE CODE	
B.	SCREEN SHOTS	

LIST OF FIGURES

Fig No.	TITLE	Page No.
1.1	Picture Identification	2
3.1	Block Diagram for Language Translation	9
4.1	Architecture of Android System	1
4.2	Levels of API	15
4.3	Declaration of Application Manifest	16
5.1	Text Detection	18
5.2	Object Detection	19
5.3	File Provider Location	19
5.4	Match Image	20
5.5	Translator Editor Tab	21
5.6	Capture Image File	22
5.7	Object Detection Function	23
5.8	DFD (Data flow Datagram)	24
5.9	App Engine Model	27
5.10	Android Manifestation Tab	28
5.11	Organization of App Modules	29
5.12	Inspection Results	32
6.1	Process of how the app Works	33

CHAPTER 1

INTRODUCTION

1.1 Picture Translation

Trying to grasp and understand the universal nuances of multiple languages is a whole other game. To date, the only way to translate has been for 'human translators' to study many languages and attempt to interpret and bridge language boundaries for various purposes. The complexities of the characters may certainly put even the most modern technology to the test. Translation applications may be lifesavers when you are travelling and are surrounded by people who speak or read a language you do not understand. The users may either take a picture or select an image from their phone's gallery. The Google API will recognize the image and translate it into the language the user prefers to select. Here, the system can detect an object (object detection) such as a computer, furniture, and so on. It is the same as the object module, but here the text must be available for it to run efficiently. Because it can only recognize text that is present in the picture. Imagine you are a professor who gave a presentation with English slides. Now you are appointed to give another presentation to Italian audience. For better understanding, you might want to generate a presentation instantly in Italian text keeping everything same. Pic Translate does the exact same thing with a few clicks! The application is used to translate different words from different languages through image. You can translate text in photos you have already taken. As well as taking a new photo in the Translate app and translate text on the captured photo using Optical Character Recognition (OCR) extracting text from images so that it can be edited, copied, formatted, indexed, searched, or translated covering many languages.

1.2 Picture Translation using app

AI Camera is the best image translator app like Google Lens that has made it easy for frequent travellers to translate text, text from images, text boards, objects, and files, into any preferred language. It is an app that identifies objects by picture. Not only this, but the app also converts the image texts into users preferred language as it supports almost all the languages. Moreover, it has a dictionary of different words that supports a range of languages like Francis, Italian, Japanese, Netherlands, Chinese, Japanese, Korean, and many more language. I will try to describe the use and development of the AI based pic Translate app. Most of the existing apps focus on just translating the text written on the image and show it in another new window. This translation may help you

in understanding the text but the originality of the scene is lost and you might not be able to relate the picture with the text. Keeping the valuable experience of the user in mind and the advanced technology of AI, pic Translate is developed to preserve the background of the image while converting the text into the desired language. There are over 50 languages to interchangeably translate the text. These new images will enhance your presentation and will make it more revealing and expressive.

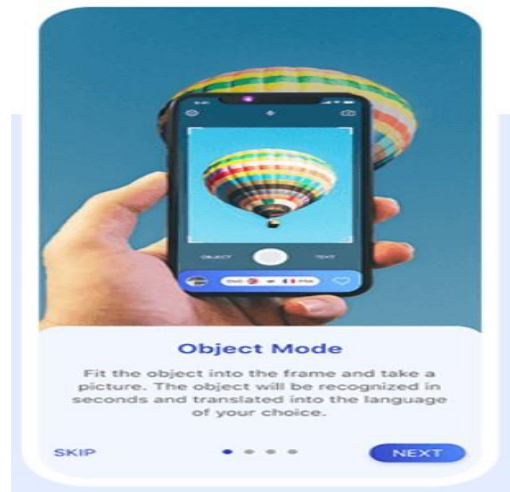


Fig:1.1: Picture Identification

1.3 AI technique for Translation

AI translation is a machine translation process based on complex deep learning algorithms. Using intelligent behaviour, it can reason and understand a source text and generate another in a different language. In order to understand the source and linguistically construct sentences in another language, a deep AI translator, or deep learning, employs a system of neural networks inspired by the workings of the human brain. In machine translation, artificial intelligence has helped algorithms to improve parsing behaviour, learn word usage, interpret according to context, and structure the sentence: A language translator is a mobile application that can be utilized for translating from one language to another. The problem of language difference has hindered effective information communication over the years. This traditional approach used for solving the problem of language differences has not been productive and favourable. The study develops an android language converter app in order to make

learning and language translation easy and facilitates stress-free communication. The system will also be able to evaluate language translation to determine their suitability for everyday conversation.

CHAPTER 2

LITERATURE REVIEW

This Chapter will talk about the techniques and methods used for Translating Image into text to users preferable and comfortable language.

2.1 Language Translation Platform

Manjur Kolhar and Abdalla Al ameen has said that the ETF English language translation framework can be used by faculty members who find it difficult to convey the meaning of English words or sentences. Clients can access this framework through their computers or laptops and handheld devices, such as mobile phones and tablets. To access the framework, it is always necessary to login to the PSAU website (www.psau.edu.sa). This website also allows clients to download their courses for the current semester through their individual accounts and dashboards. The PSAU (www.psau.edu.sa) server affords many services to students and faculty members. For example, students can see their attendance and grades through the blackboard, library service, and candidature information. Hence, the integration of the proposed ETF into this website is also anticipated. An English translation client (ETC) communicates with the PSAU server, which then communicates with the Google Cloud translation server upon receiving a translation request. A cloud translation application programming interface, which is accessible via the Google Cloud platform, is used; customers can integrate Google Translate into their own site or application. The use of Google Translate API is governed by terms and conditions: for example, timers are required to adhere to certain layout rules and branding. The following must be strictly observed by customers who employ the Cloud Translation API service in their websites.

2.2 An Android-based language translator

Roseline Oluwaseun Ogundokun has proposed Android Text translation which is based on a double objective of a mobile translator, text libraries of different languages were built using MATLAB, which includes the most frequently used terms in conversations in daily life. The user is asked to enter the text in the android code and click on the language-named button to convert to the text production. demonstrates the system architecture of the mobile translator. This is showed how the system proposed will function in developing this android application which can translate the most used languages in the world (i.e., English, Spanish, French, Hindi, Arabic, and Chinese) to any selected languages, the android application will be using the IBM Waston Language

translator, an API that uses machine learning and AI. IBM Watson employs rule-based AI and a broad range of other mechanisms (e.g., information retrieval systems) and the latest Neural network AI [60]. In Theory, the application was created using Apache Cordova and Node.js application platform operating on IBM Cloud Kubernetes software that uses Tesseract OCR to identify text in pictures, Watson Language Converter to interpret the approved text, and Watson Natural Language Comprehension to extract sentiment, the meaning of the book

2.3 Language Translator Application using Fire Base

Vaishnavi, HR Dhanush Datta, Varsha Vemuri, L Janaki proposed a system where they will be implementing translation with all the functionalities like scanning text, speech recognition and translates the text and includes the languages which are popular in our country as well as popular all over the world. The advantage of this application is it does not require internet connectivity. To develop the language translator in order to understand unknown languages to convert image to text with firebase optical character recognition, translate to user specific language to convert voice to text, translating to user specific language. User specific language as input text and translate to user interested language that support firebase. So here in the proposed system where we will be implementing translation with support all the functionalities like scanning text, speech recognition and translates the text and includes the languages which are popular in our country as well as popular all over the world. The advantage of this application is it does not require internet connectivity. Proposed an android application captures image using device camera process it and determines the corresponding gesture.

2.4 Sign Language Translator for Mobile Platforms

Mahesh M, Arvind Jayaprakash, Geetha M they proposed an android application that converts sign language to natural language and enable deaf and dumb community to talk over mobile phones. Developing Sign Recognition methods for mobile applications has challenges like need for light weight method with less CPU and memory utilization. The application captures image using device camera process it and determines the corresponding gesture. An initial phase of comparison using histogram matching is done to identify those gestures that are close to test sample and further only those samples are subjected to Oriented Fast and Rotated BRIEF (Binary Robust Independent Element Features) based comparison hence reducing the CPU time. The

user of the application can also add new gestures into the dataset. The application allows easy communication of deaf and dumb with society. Though there are many computer-based applications for sign language recognition, development in android platform is adequately less. This paper is focused on providing an applicative architecture of hand glove that records the gestures made by a speech and hearing disabled people, converts them into a meaningful text and transmits them to remote areas with help of Bluetooth

2.5 English Text to Multilingual Speech Translator

Vivek Hanumanth, Rubi Debnath, Disha Bhattacharjee, Deepti Tripathi, Sahadev their aim is at providing design and development solution of an Android application whose objective is to provide a solution to overcome the barrier of languages by implementing text to speech conversion in different languages. The Android application developed text to speech conversion to facilitate the translation of English language text into speech output in different languages. We have also proposed few improvements which can further advance this system to include more target audiences to make it more beneficial and useful. The proposed English Text to Multilingual Speech Translator using Android (T2MSTA) aims at aiding the people lacking the power of speech or non-native speakers like people who do not share a common language. Essentially, this approach uses computing power to build sophisticated data models to translate one source language into another. Statistical Machine Translation consists of Language Model (LM), Translation Model (TM) and Decoder.

2.6 Literature Summary

As observed in the literature survey most of the projects implemented till date are all concentrated on converting the text or the text which is present on the text. We found that using by using the technology of Artificial Intelligence and Java programming language we can convert the Image into users' comfortable language and by using the application language translator using android application we have been able to convert more than 160 languages through this app. Not only this, but the app also converts the image texts into users preferred language as it supports almost all the languages. Moreover, it has a dictionary of different words that supports a range of languages like Francis, Italian, Japanese, Netherlands, Chinese, Japanese, Korean, and many more languages.

2.7 Problem Definition

During the identification of image there may be many other images which are similar and some pictures were not identified exactly and in translating words there may be partial spelling mistakes.

2.8 Objective of the project

The users may either take a picture or select an image from their phone's gallery. The Google API will recognize the image and translate it into the language the user prefers to select. Here, the system can detect an object (object detection) such as a computer, furniture, and so on. It is the same as the object module, but here the text must be available for it to run efficiently. Because it can only recognize text that is present in the picture. Instead of translating it, again and again, the user can mark it as favourite and then maintain a complete list of favourites under this image to text converter app.

CHAPTER – 3

EXISTING SYSTEM

3.1 – Artificial Intelligence Based Language Translation Platform

The use of computer-based technologies by non-native Arabic-speaking teachers for teaching native Arabic-speaking students can result in higher learner engagement. In this study, a machine translation (MT) system is developed as a learning technology. The proposed system can be linked to a digital podium and projector to reduce multitasking. A total of 25 students from Prince Sattam bin Abdulaziz University, Saudi Arabia participated in our experiment and survey related to the use of the proposed technology-enhanced MT system. An important reason for using this framework is to exploit the high service bandwidth (up to several bandwidths) made available for interactive translation services. The framework is deployed by linking it to a video camera, digital podium, and projector in each classroom for the translation operation. Accordingly, After employing the system, the effect on the understanding of the students of the technical aspects of the subject that was taught using English was evaluated. The results indicated that the use of the developed technology for translation during classroom sessions was beneficial. Furthermore, the engagement of students in the classes improved their performance and learning outcomes. The students also commented that the proposed framework is useful from two perspectives: vocabulary improvement and subject comprehension. Many of the students indicated that working on assignments and homework using the framework was useful because words difficult to understand were translated the use of an instant MT service is proposed. The use of PowerPoint presentations (PPTs) is included with the proposed service because Prince Sattam bin Abdulaziz University (PSAU) and its affiliated colleges use digital podiums and projectors in the classrooms. Most of the universities in Saudi Arabia under the Ministry of Education employ a similar setup, in which blackboards are equipped with projectors and digital podiums; thus, the implementation of the proposed method using this hardware is feasible. When students struggle to understand any English word presented in the PPT, the proposed framework can aid students to understand it by immediately providing a translation, allowing the students to learn and the faculty members to teach in a typical classroom environment. Moreover, the use of multimedia equipment has been observed to increase the attentiveness of students. Numerous MT systems are available; however, they cannot be linked to PPTs because they are third-

party solutions. It was found that teachers who multitasked in the use of the medium of instruction scored lower in their evaluations than those who were not involved in multitasking. Hence, the proposed service has been developed in such a way that it can be easily linked to a PPT to display the instant translation on the same PPT slide being presented by the teacher. Different from earlier MT systems based on human intervention, recent technological advancements have enabled the translation of source documents into various languages using advanced machine learning techniques

3.2 Block Diagram

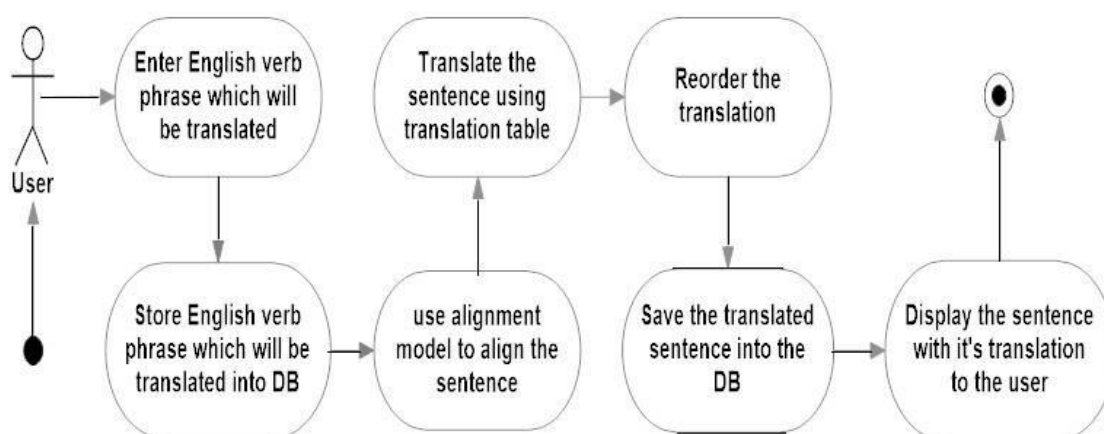


Fig:3.1: Block diagram for Language Translation

The proposed model algorithms for the model learning, training and the model translation are System administrator will support the dictionary and feed it with vocabularies and translated sentences or parallel corpora, which used as a reference for the translation model. He will enter this data through the user interface (UI) and Database Management System (DBMS) tool. Prepare the parallel corpora for the training process by normalizing words, tokenization, true casing, and cleaning. Create the language model for the source language to help in translation process. 4. Train the word alignment model with parallel corpus data. Save the alignment model files for references and to help create the reordering model. Generate translation table with maximum likelihood estimation. Save the translation table. Build phrases table for translated phrases with their scores. Save the phrases table. Build reordering model. Save the reordering table. Build the translation model. Save the translation model. Tune the translation model and minimize the error rate for translation to improve translation quality and add model upgrade.

Binaries the phrase table and reordering table. By performing all the previous steps, the translation model would be trained for parallel corpus.

The user source statement translation algorithm

User enters the English verb phrase statement, which will be translated through the UI and DBMS tool.

Store the English verb phrase to the DB.

Align the sentence using alignment model.

Translate the sentence using the translation table.

Reorder the translated sentence.

Save the translated sentence into the DB.

Display the translated sentence to the user.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Software Used for the Proposed System

The Software that is used for our Proposed System is Android Studio which is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc. It contains a Linux-based Operating System, middleware, and key mobile applications. It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in various devices such as mobiles, tablets, televisions etc. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. OHA a consortium of 84 companies such as google, Samsung, AKM, synaptic, KDDI, Garmin, Teleca, Ebay, Intel etc. It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services, and deploy handsets using the Android Platform. After learning what is android, let us see the features of android. The important features of android are given below: It is open-source, anyone can customize the Android Platform, there are a lot of mobile applications that can be chosen by the consumer, it provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc. The Android software development kit (SDK) includes a comprehensive set of development tools. The Android SDK Platform Tools are a separately downloadable subset of the full SDK, consisting of command-line tools such as `adb` and `fast boot`.

The Android Debug Bridge (ADB) is a tool to run commands on a connected Android device. Fast boot is a protocol used for flashing filesystems. Code written in C/C++ can be compiled to ARM, or x86 native code (or their 64-bit variants) using the Android Native Development Kit (NDK). The Android 3.1 platform (also backported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple

protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode. Obstacles to development include the fact that Android does not use established Java standards, that is, Java SE and ME. This prevents compatibility between Java applications written for those platforms and those written for the Android platform. Android reuses the Java language syntax and semantics, but it does not provide the full class libraries and APIs bundled with Java SE or ME. Obstacles to development include the fact that Android does not use established Java standards, that is, Java SE and ME. This prevents compatibility between Java applications written for those platforms and those written for the Android platform. Android reuses the Java language syntax and semantics, but it does not provide the full class libraries and APIs bundled with Java SE or ME. Obstacles to development include the fact that Android does not use established Java standards, that is, Java SE and ME. This prevents compatibility between Java applications written for those platforms and those written for the Android platform. Android reuses the Java language syntax and semantics, but it does not provide the full class libraries and APIs bundled with Java SE or ME.

4.2 Explanation of Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as: A flexible Gradle-based build system, A fast and feature-rich emulator, A unified environment where you can develop for all Android devices Instant Run to push changes to your running app without building a new APK Code templates and GitHub integration to help you build common app features and import sample code Extensive testing tools and frameworks, Lint tools to catch performance, usability, version compatibility, and other problems, C++ and NDK support Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine. Let us see the Android Architecture System:

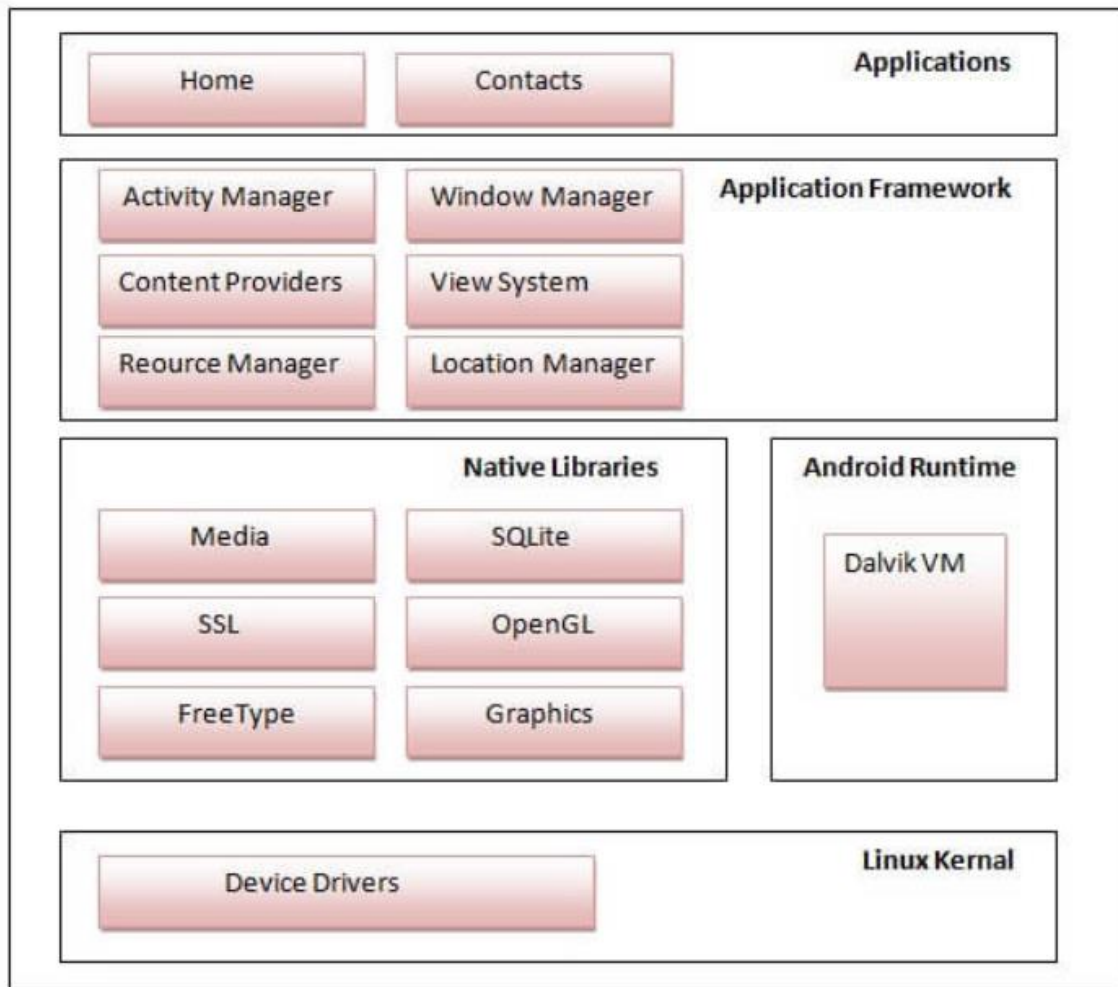


Fig:4.1: Architecture of Android System

Android architecture or Android software stack is categorized into five parts:

Linux kernel: It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

Native Libraries: On the Top of Linux Kernel there are Native Libraries such as Web Kit, OpenGL, Free Type, SQLite, Media, Run Time Library (Lib) Etc. The Web Kit library is responsible for browser support, SQLite is for database, Free Type for font support, Media for playing and recording audio and video formats. **Android Runtime:** In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application.

DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

Application Framework: On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

Applications: On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using Linux kernel. API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.

The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of: A core set of packages and classes, A set of XML elements and attributes for declaring a manifest file, A set of XML elements and attributes for declaring and accessing resources, A set of Intents, A set of permissions that applications can request, as well as permission enforcements included in the system. Each successive version of the Android platform can include updates to the Android application framework API that it delivers.

Updates to the framework API are designed so that the new API remains compatible with earlier versions of the API. That is, most changes in the API are additive and introduce new or replacement functionality. As parts of the API are upgraded, the older replaced parts are deprecated but are not removed, so that existing applications can still use them. In a very small number of cases, parts of the API may be modified or removed, although typically such changes are only needed to ensure API robustness and application or system security. All other API parts from earlier revisions are carried forward without modification. The framework API that an Android platform deliver is specified using an integer identifier called "API Level". Each Android platform version supports exactly one API Level, although support is implicit for all earlier API Levels (down to API Level 1). The initial release of the Android platform provided API Level 1 and subsequent releases have incremented the API Level.

The following table specifies the API Level supported by each version of the Android platform.

Platform Version	API Level
Android 2.3.3	10
Android 2.3	9
Android 2.2	8
Android 2.1	7
Android 2.0.1	6
Android 2.0	5
Android 1.6	4
Android 1.5	3
Android 1.1	2
Android 1.0	1

Fig:4.2: Levels of API

The API Level identifier serves a key role in ensuring the best possible experience for users and application developers:

It lets the Android platform describe the maximum framework API revision that it supports:

It lets applications describe the framework API revision that they require

It lets the system negotiate the installation of applications on the user's device, such that version-incompatible applications are not installed.

Each Android platform version stores its API Level identifier internally, in the Android system itself.

Applications can use a manifest element provided by the framework API — `<uses-sdk>` — to describe the minimum and maximum API Levels under which they are able to run, as well as the preferred API Level that they are designed to support. The element offers three key attributes.

`android:minSdkVersion` Specifies the minimum API Level on which the application can run. The default value is "1".

android:targetSdkVersion — Specifies the API Level on which the application is designed to run. In some cases, this allows the application to use manifest elements or behaviours defined in the target API Level, rather than being restricted to using only those defined for the minimum API Level.

android:maxSdkVersion — Specifies the maximum API Level on which the application can run. **Important:** Please read the <uses-sdk> documentation before using this attribute.

When the user attempts to install an application, or when revalidating an application after a system update, the Android system first checks the <uses-sdk> attributes in the application's manifest and compares the values against its own internal API Level. The system allows the installation to begin only if these conditions are met:

If an android:minSdkVersion attribute is declared, its value must be less than or equal to the system's API Level integer. If not declared, the system assumes that the application requires API Level 1.

If an android: maxSdkVersion attribute is declared, its value must be equal to or greater than the system's API Level integer. If not declared, the system assumes that the application has no maximum API Level. Please read the <uses-sdk> documentation for more information about how the system handles this attribute.

When declared in an application's manifest, a <uses-sdk> element might look like this:

```
<manifest>
  <uses-sdk android:minSdkVersion="5" />
  ...
</manifest>
```

Fig:4.3: Declaration of application Manifest

The principal reason that an application would declare an API Level in android:minSdkVersion is to tell the Android system that it is using APIs that were *introduced* in the API Level specified. If the application were to be somehow installed on a platform with a lower API Level, then it would crash at run-time when it tried to access APIs that do not exist. The system prevents such an outcome by not allowing the application to be installed if the lowest API Level it requires is higher than that of the platform version on the target device.

For example, the `android.appwidget` package was introduced with API Level 3. If an application uses that API, it must declare a `android:minSdkVersion` attribute with a value of "3". The application will then be installable on platforms such as Android 1.5 (API Level 3) and Android 1.6 (API Level 4), but not on the Android 1.1 (API Level 2) and Android 1.0 platforms (API Level 1).

Application forward compatibility, applications are generally forward-compatible with new versions of the Android platform. Because almost all changes to the framework API are additive, an Android application developed using any given version of the API (as specified by its API Level) is forward-compatible with later versions of the Android platform and higher API levels. The application should be able to run on all later versions of the Android platform, except in isolated cases where the application uses a part of the API that is later removed for some reason.

Forward compatibility is important because many Android-powered devices receive over-the-air (OTA) system updates. The user may install your application and use it successfully, then later receive an OTA update to a new version of the Android platform. Once the update is installed, your application will run in a new run-time version of the environment, but one that has the API and system capabilities that your application depends on.

In some cases, changes *below* the API, such those in the underlying system itself, may affect your application when it is run in the new environment. For that reason, it's important for you, as the application developer, to understand how the application will look and behave in each system environment. To help you test your application on various versions of the Android platform, the Android SDK includes multiple platforms that you can download. Each platform includes a compatible system image that you can run in an AVD, to test your application.

CHAPTER 5

Description of Proposed System

5.1 Architecture Model of Proposed System

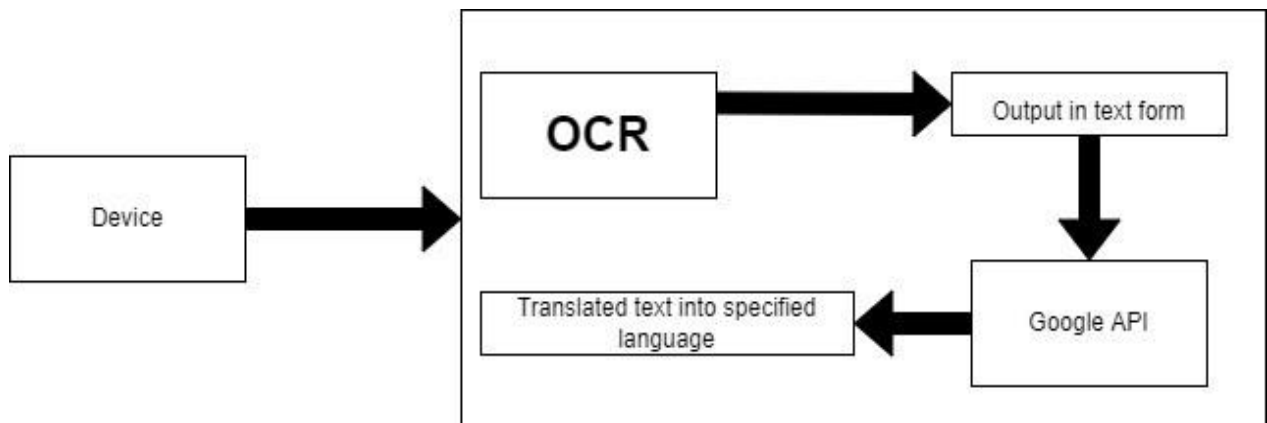
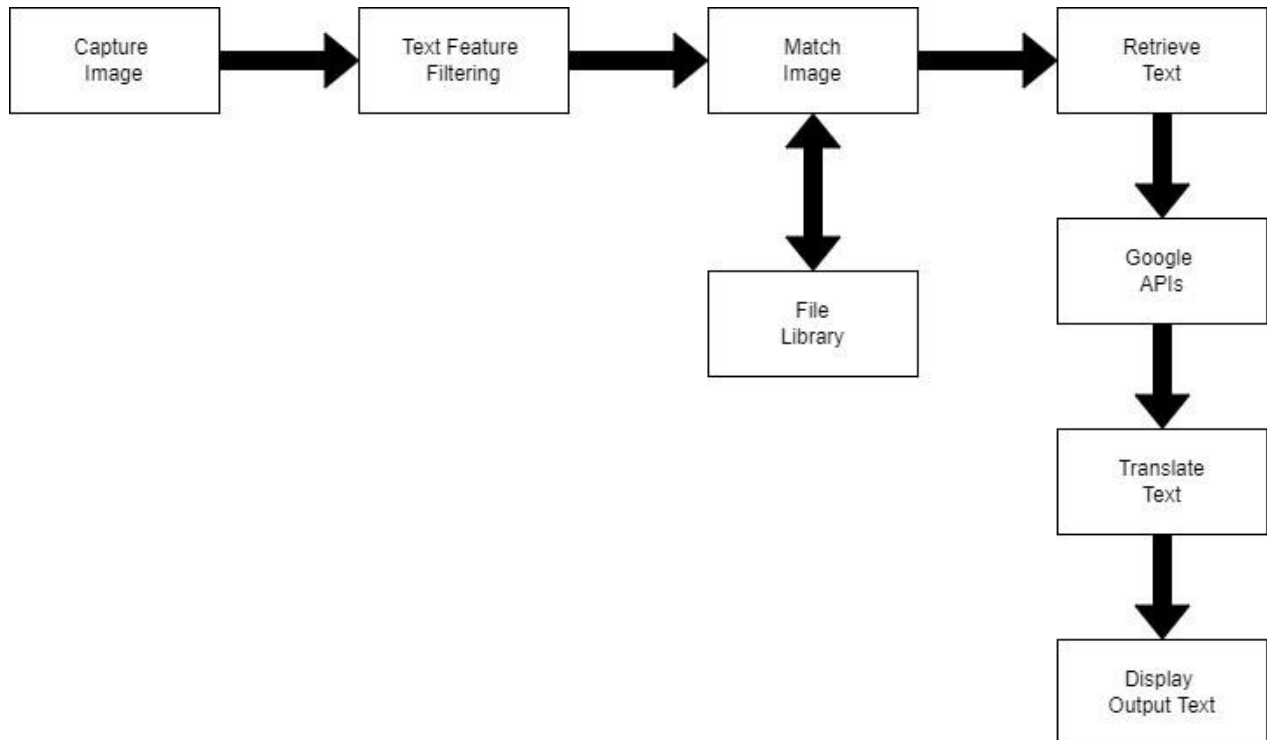


Fig:5.1: Text Detection

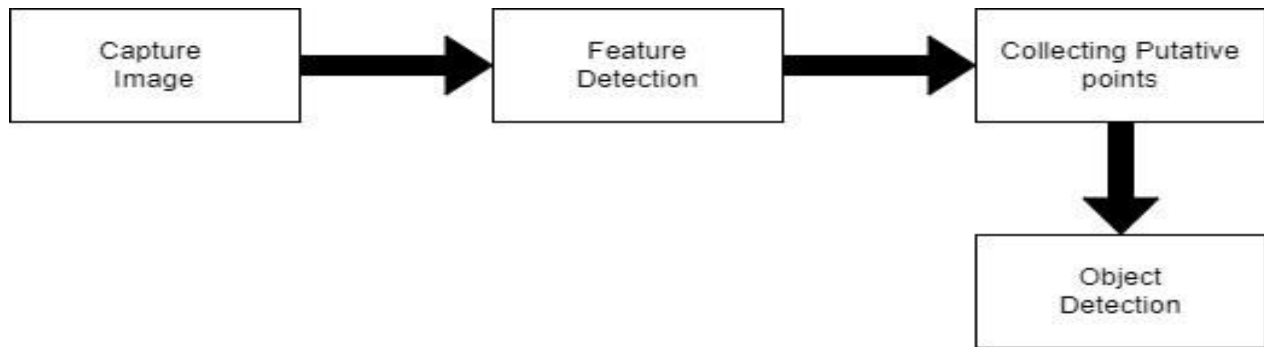


Fig:5.2: Object Detection

Capture Image: Camera is a very common feature that lots of apps provide. Most of the required image capture feature in own application. e.g. profile image, creating a post with an image, every food or social media app required camera feature in own application.

Basically, Image capturing is very simple task, even if you are using custom. In this article, we are going to see how to capture image from camera using FileProvider in your Android app. Google introduces File Provider in version 22.1.0. FileProvider class is a derived class from Content Provider. In other words, you can say, FileProvider provide facilitates to secure file sharing in android via `content://Uri` instead of a `file:///Uri`. Define list of file paths for FileProvider in xml file and save it in `res/xml` folder. Since we want to save only in public picture directory, we will just add path of it to xml.

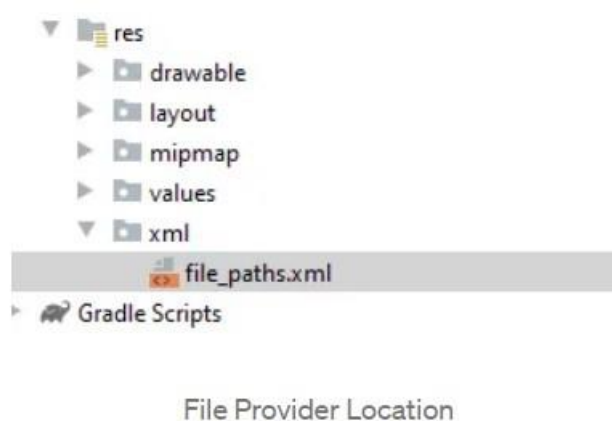


Fig:5.3: File Provider Location

Text Feature Filtering: The purpose of a `<uses-feature>` declaration is to inform any external entity of the set of hardware and software features your application depends

on. The element offers a required attribute that lets you specify whether your application requires and can't function without the declared feature or prefers to have the feature but can function without it. Because feature support can vary across Android devices, the `<uses-feature>` element serves an important role in letting an application describe the device-variable features that it uses. The set of available features that your application declares corresponds to the set of feature constants made available by the Android Package Manager. Feature constants are listed in the Features reference section in this document. You must specify each feature in a separate `<uses-feature>` element, so if your application requires multiple features, it declares multiple `<uses-feature>` elements. Starts an asynchronous filtering operation. Calling this method cancels all previous non-executed filtering requests and posts a new filtering request that will be executed later.

Match Image: The image capture use case is designed for capturing high-resolution, high-quality photos and provides auto-white-balance, auto-exposure, and auto-focus (3A) functionality, in addition to simple manual camera controls. The caller is responsible for deciding how to use the captured picture, including the following options

- `takePicture (Executor, OnImageCapturedCallback)`: This method provides an in-memory buffer of the captured image
- `takePicture (OutputFileOptions), Executor, (OnImageSavedCallback)`: This method saves the captured image to the provided file location.

There are two types of customizable executors on which Image Capture runs, the call back executor, and the IO executor. The call back executor is the parameter of the `takePicture` methods



```
Kotlin  Java

ImageCapture imageCapture =
    new ImageCapture.Builder()
        .setTargetRotation(view.getDisplay().getRotation())
        .build();

cameraProvider.bindToLifecycle(lifecycleOwner, cameraSelector, imageCapture, imageAnalysis, preview);
```

Fig:5.4: Match Image

Google API: To use the Tuning Fork library in your own app, enable the Android Performance Parameters API in the Google Cloud Console. You need a Google developer account to create a Google Cloud Console project. You can use an existing Google Cloud Console project (for example, the one you created for the demo app). Follow the steps below to add the proper API key restrictions to your existing project.

Translate Text: In your projects file navigator, expand the newly created values directory, right-click the strings.xml file to open the Translation Editor. In the Translation Editor, click the globe-icon to add a new language file, in our example we will add 'de' (German) and 'es' (Spanish). We also want to add the region locale 'de-at' for Austria, which can't be created with the editor. As a result, we need to create the directory 'values-de-rat' in the resources folder and create a strings.xml.

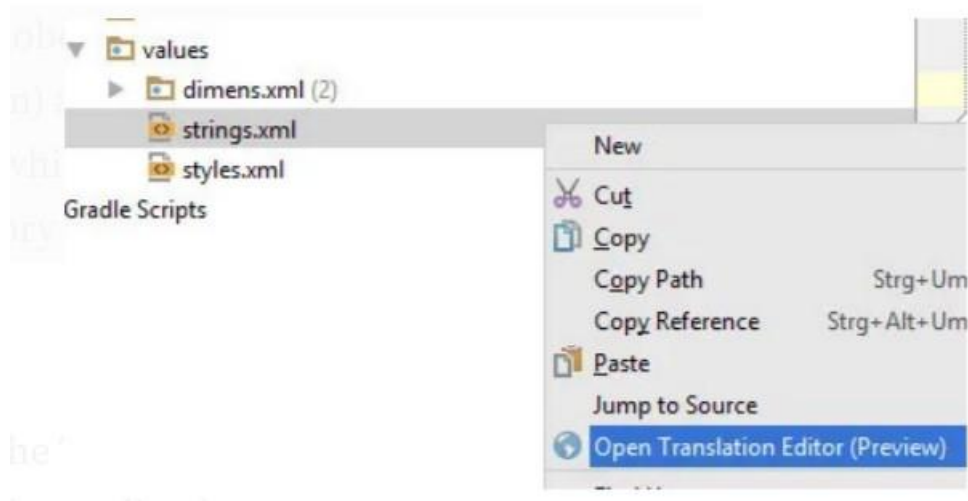


Fig:5.5: Translator Editor Tab

Display Output Text: Text View is a simple widget that is seen in every android application. This widget is used to display simple text within the android application. We can add custom styling to the text that we have to show. In this article, we will look at How to create a simple Text View in an android application.

OCR (Optical character recognition): **OCR** is the process of reading printed or handwritten text and converting them into machine-encoded text. OCR is mainly used in the field of artificial intelligence, pattern recognition, and computer vision.

Capture Image: *Object detection* is the machine learning task of identifying the presence and location of multiple classes of objects within an image. An object

detection model is trained on a dataset that contains a set of known objects. The trained model receives image frames as input and attempts to categorize items in the images from the set of known classes it was trained to recognize. For each image frame, the object detection model outputs a list of the objects it detects, the location of a bounding box for each object, and a score that indicates the confidence of the object being correctly classified.

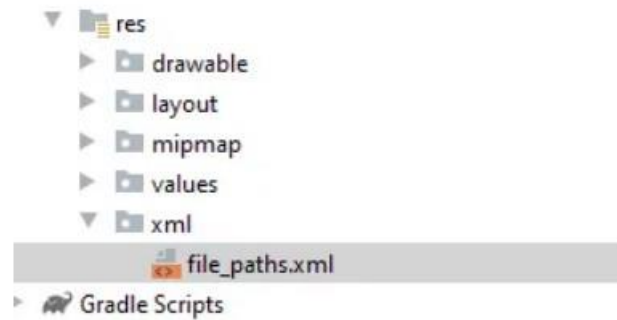


Fig:5.6: Capture Image file

Feature Detection: When you pass an image to ML Kit, it detects up to five objects in the image along with the position of each object in the image. When detecting objects in video streams, each object has a unique ID that you can use to track the object from frame to frame. You can use a custom image classification model to classify the objects that are detected. Please refer to Custom models with ML Kit for guidance on model compatibility requirements, where to find pre-trained models, and how to train your own models. There are two ways to integrate a custom model. You can bundle the model by putting it inside your app's asset folder, or you can dynamically download it from Firebase.

Object Detection: In `STREAM_MODE` (default), the object detector runs with low latency, but might produce incomplete results (such as unspecified bounding boxes or category labels) on the first few invocations of the detector. Also, in `STREAM_MODE`, the detector assigns tracking IDs to objects, which you can use to track objects across frames. Use this mode when you want to track objects, or when low latency is important, such as when processing video streams in real time. In `SINGLE_IMAGE_MODE`, the object detector returns the result after the object's bounding box is determined. If you also enable classification, it returns the result after the bounding box and category label are both available. Therefore, detection latency is potentially higher. Also,

in SINGLE_IMAGE_MODE, tracking IDs are not assigned. Use this mode if latency isn't critical and you don't want to deal with partial results. The object detection and tracking API is optimized for these two cores use they are Live detection and tracking of the most prominent object in the camera viewfinder, The detection of multiple objects from a static image.



Fig:5.7: Object Detection Function

5.2 Explanation of Data Base

The Database Inspector lets you inspect, query, and modify your app's databases while your app is running. This is especially useful for database debugging. The Database Inspector works with plain SQLite and with libraries built on top of SQLite, such as Room. There are many ways to store your data, such as in an online database, in a local SQLite database, or even in a text file. It is up to you to decide what is the best solution for your application.

This lesson shows you how to create an SQLite virtual table that can provide robust full-text searching. The table is populated with data from a text file that contains a word and definition pair on each line in the file.

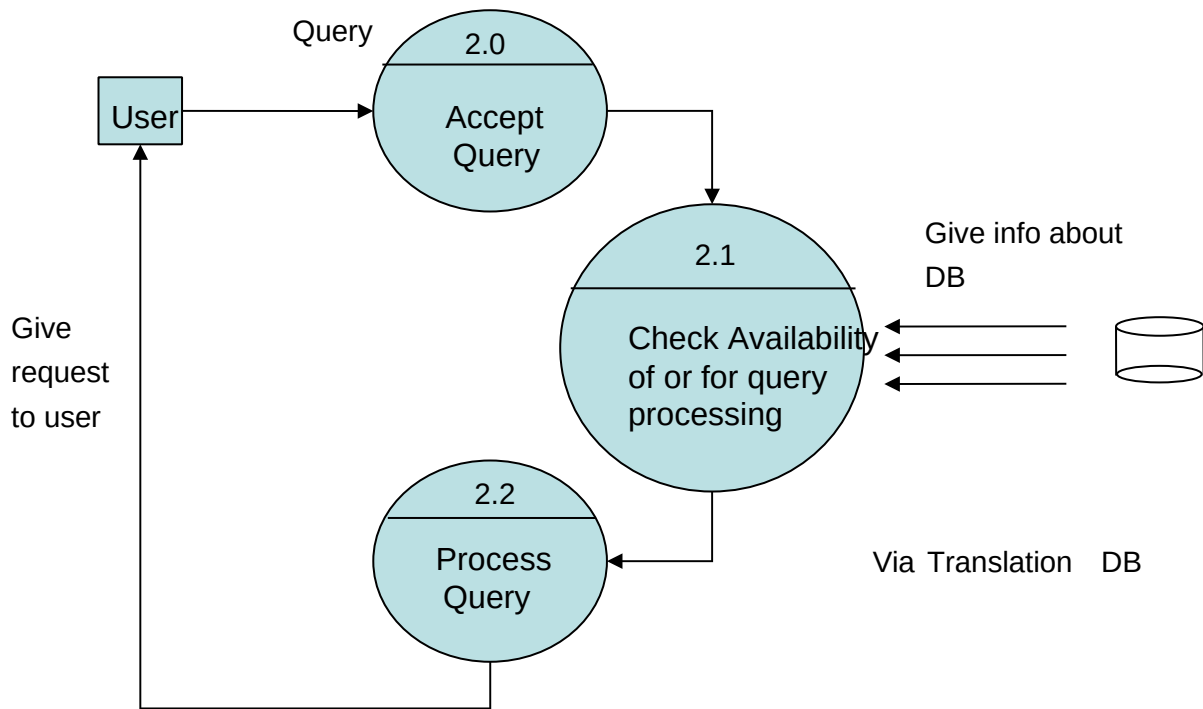


Fig:5.8: DFD (Database Flow Diagram Prediction)

5.3 Language used for Proposed System

Android is very popular OS for mobile phones today. Millions of apps are there that power different types, models and manufactures mobile phones today. Java the popular programming language is used to develop Android applications.

There are solid reasons why java is used for android development. The main objective behind Android development was to create a platform-independent application environment that can run on every device. As we know java already have this quality so java was chosen for android development. Android applications run on a special virtual machine called the Dalvik VM that is a direct inspiration from java virtual machine called JVM. Android application can run on any device where special Dalvik VM is implemented. These way android applications are compiled and run in optimum performance environment with the feature of platform independence.

The good approach towards software development is the object-oriented approach. Java is based on the oops concept. Android relies heavily on Java fundamentals like classes and objects and its other useful features of oops. Java has an extensive set of libraries. It is easy to take advantage of these libraries. Android SDK has many standard Java libraries included. These provide functionalities for data structure, math functions,

graphics implantation, and networking functions and much more. These java libraries help us to do everything else we could want. This way java helps develop Android applications fast and inefficient manner. Android is made to run on different platforms i.e., hardware platforms. Thus, architectural neutrality is desired and necessary. Android code is written once and to execute need to compile and optimise native code for better performance on various devices. Java has platform independent feature so it is used for android development. Java is very popular language due to its awesome features and performance. The community of Developers those have proficiency is big. Thus, android developers to choose java as there is already a good base of java programmers are available that can help in creating, improving android applications plus with many libraries and tools of java make developers life easier. Large java developer base enables to develop a lot of android apps fast so it is based on java. Developers those do not use java have to deal with serious problems like memory leads and bad pointer usage. Sometimes these problems harm at the highest level like the crash of application or crash of OS. Android easily implement and fix common problems with other programming languages with the help of Java. These are some problems those never occur when you program with java. Java is machine independent and runs only in JVM space so it protects you from these problems. Java is the best and the first language for native software development. That is really the only option for native applications, native applications are the heart of android. Java's important core features also motivated android developers to use it for development, these features include easy learning, platform-independent, secure, object-orientation.

5.4 Implementation Details

The Project application is loaded in Android Studio. We used Android Studio for Design and coding of project.

Hardware Requirement:

1. Laptop or PC

➤ i3 Processor Based Computer or higher

➤ 1GB RAM

- 5 GB Hard Disk

2. Android Phone or Tablet

- 1.2 Quad core Processor or higher

- 1 GB RAM

❖ Software Requirement:

1. Laptop or PC

- Windows 7 or higher.

- Java

- Android Studio

2. Android Phone or Tablet

- Android v5.0 or Higher

5.5 Structure of the Project

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

Android app modules: A module is a collection of source files and build settings that let you divide your project into discrete units of functionality. Your project can have one or many modules, and one module can use another module as a dependency. You can independently build, test, and debug each module. Additional modules are useful when creating code libraries within your own project or when you want to create different sets of code and resources for different device types, such as phones and wearables, but keep all the files scoped within the same project and share some code. To add a new module to your project, click File > New > New Module.

Library modules: An Android library is structurally the same as an Android app module. It includes everything needed to build an app, including source code, resource files, and

an Android manifest. However, instead of compiling into an APK that runs on a device, an Android library compiles into an Android Archive (AAR) file that you can use as a dependency for an Android app module. Unlike JAR files, AAR files offer the following functionality for Android apps: AAR files can contain Android resources and a manifest file, which lets you bundle in shared resources like layouts and drawable in addition to Kotlin or Java classes and methods, AAR files can contain C/C++ libraries for use by the app module's C/C++ code.

By default, Android Studio displays your project files in the Android project view. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

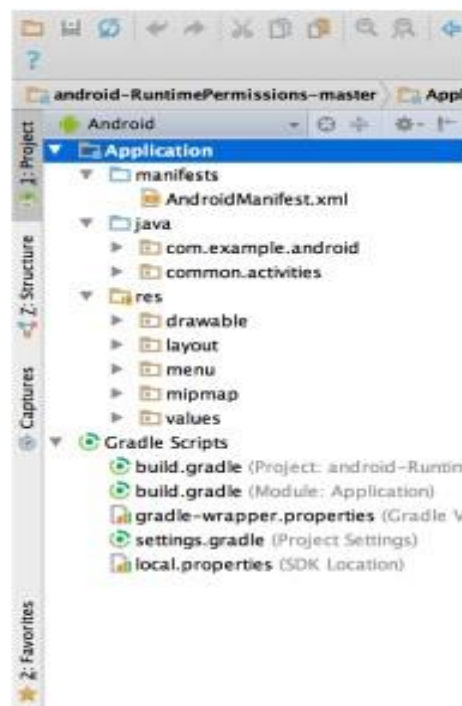


Fig:5.9: App Engine Modules

manifests: Contains the AndroidManifest.xml file. The Android project structure on disk differs from this flattened representation.

java: Contains the Java source code files, including JUnit test code. RestAPI.java [Android Side] -This File consist of all the functionalities which are used to communicate with the database in the form of functions. -This file has a URL String which is globally

declared and holds a URL which has the web services. -When you call a function from the Restapi the URL is hit and the function is called in the server side. -All the functions are not used directly on the main Thread but on a Asynchronous Task and using Another class JSONParse we parse the Json object to get it in the String Format.

res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images. You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element. You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element. The User Interface.

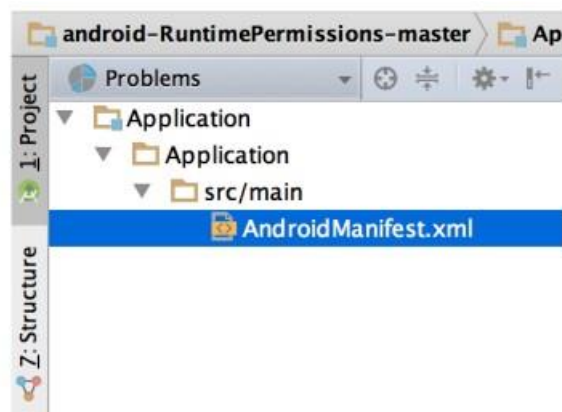


Fig:5.10: Android Manifestation Tab

TOOL BAR: The toolbar lets you carry out a wide range of actions, including running your app and launching Android tools.

NAVIGATION BAR: The navigation bar helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.

EDITOR WINDOW: The editor window is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.

TOOL WINDOW BAR: The tool window bar runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows. The tool windows give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.

STATUS BAR: The status bar displays the status of your project and the IDE itself, as well as any warnings or messages. You can organize the main window to give yourself more screen space by hiding or moving toolbars and tool windows. You can also use keyboard shortcuts to access most IDE features. At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

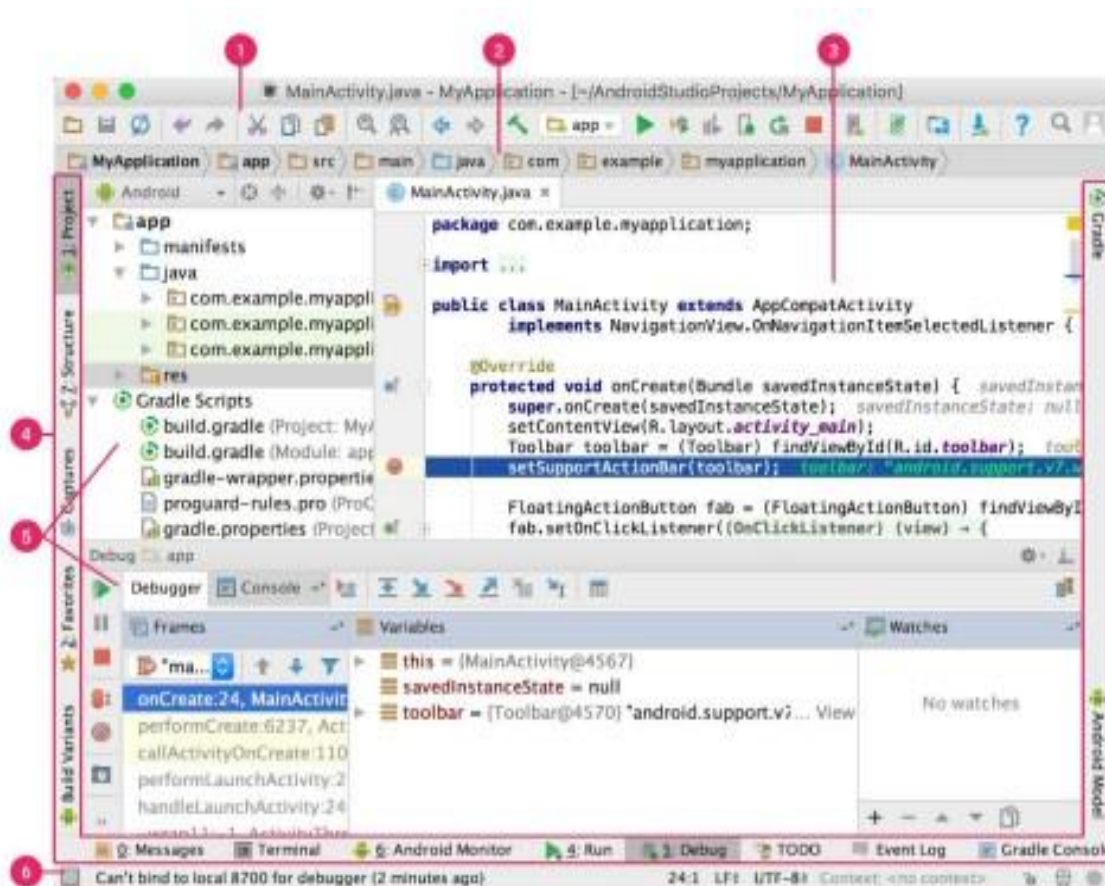


Fig:5.11: Organisation of the App modules

Tool Windows

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most used tool windows are pinned to the tool window bar at the edges of the application window. To expand or collapse a tool window, click the tool's name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows. To return to the current default tool window layout, click **Window > Restore Default Layout** or customize your default layout by clicking **Window > Store Current Layout as Default**. To show or hide the entire tool window bar, click the window icon  in the bottom left-hand corner of the Android Studio window. To locate a specific tool window, hover over the window icon and select the tool window from the menu.

Navigation Here are some tips to help you move around Android Studio. Switch between your recently accessed files using the **Recent Files** action. Press **Control+E** (**Command+E** on a Mac) to bring up the **Recent Files** action. By default, the last accessed file is selected. You can also access any tool window through the left column in this action. ad View the structure of the current file using the *File Structure* action. Bring up the **File Structure** action by pressing **Control+F12** (**Command+F12** on a Mac). Using this action, you can quickly navigate to any part of your current file. Search for and navigate to a specific class in your project using the *Navigate to Class* action. Bring up the action by pressing **Control+N** (**Command+O** on a Mac). **Navigate to Class** supports sophisticated expressions, including camel humps, paths, line navigate to, middle name matching, and many more. If you call it twice in a row, it shows you the results out of the project classes.

Navigate to a file or folder using the *Navigate to File* action. Bring up the **Navigate to File** action by pressing **Control+Shift+N** (**Command+Shift+O** on a Mac). To search for folders rather than files, add a **/** at the end of your expression. Navigate to a method or field by name using the *Navigate to Symbol* action. Bring up the **Navigate to Symbol** action by pressing **Control+Shift+Alt+N** (**Command+Shift+Alt+O** on a Mac). Find all the pieces of code referencing the class, method, field, parameter, or statement at the current cursor position by pressing **Alt+F7**. Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and

independently from the command line. You can use the features of the build system to do the following:

Customize, configure, and extend the build process, create multiple APKs for your app, with different features using the same project and modules, Reuse code and resources across source sets, by employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build Gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Multiple APK Support

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and allowing them to share test APK, java, dx, and ProGuard settings. Debug and Profile Tools android Studio assists you in debugging and improving the performance of your code, including inline debugging and performance analysis tools. Inline debugging use inline debugging to enhance your code walk-throughs in the debugger view with inline verification of references, expressions, and variable values. Inline debug information includes: Inline variable values, referring objects that reference a selected object, Method returns values, Lambda and operator expressions, Tooltip values.

Code inspections

Whenever you compile your program, Android Studio automatically runs configured Lint and other IDE inspections to help you easily identify and correct problems with the structural quality of your code. The Lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.



Fig:5.12: Inspection Results

WebAPI [REST FUL API/ASP.net with C# web service]: IServiceAPI.cs is an interface class which has all the functions which are shown in the RestAPI.java.ServiceAPI.cs: is a C# class file which has the real coding- all the queries.Handler.ashx:is the handler class where JSON is imported and values of IServiceAPI & ServiceAPI is passed. Web.config the connection string of the database is passed. DBConfig.cst this file consists of the connection object or connection string which u used by ServiceAPI.cs Working: RestApi file communicates with the WebAPI and WEBAPI with the database and JSON is used to hold and pass the data. The WebAPI is uploaded on the server and by calling the handler class from any browser and passing the ANDROID as query string will give u the RESTAPI.java which u copy paste in the android part and use it. E.g., <http://abc.hostoise.com/Handler1.ashx> Here ABC is the subdomain Hostoise.com is the domain And Handler1.ashx is the file from the WEBAPI which we call it.

CHAPTER 6

RESULTS AND DISCUSSION

AI based picture translation app discusses on how the application works by the automatic function of Auto Machine learning Languages

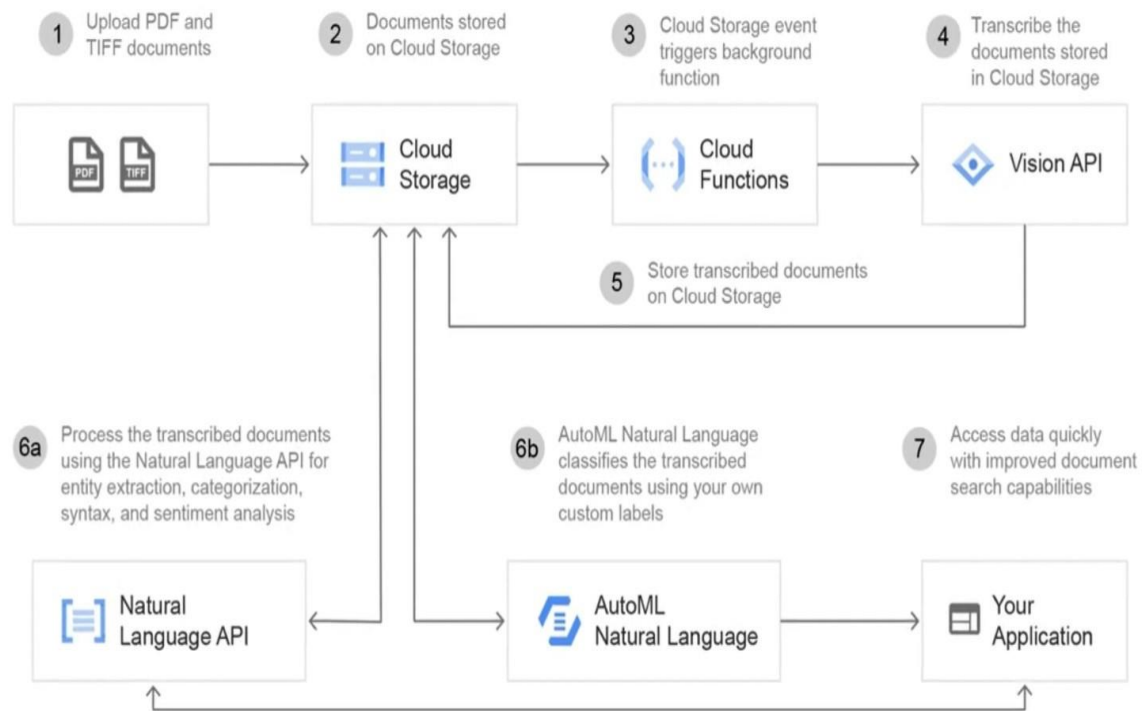


Fig:6.1: Process of how the App works

Here Firstly we need to upload or capture the picture of an real time application which will automatically gets proceeded to the cloud storage environment which we have created in library class and such that cloud storage event triggers the background application function which further proceeds to the vision of API which Transcribe the documents stored in Cloud Storage.

This further cloud storage processes the transcribed documents using the Natural Language API for entity Extraction, categorization, Syntax and sentimental analysis and then the AutoML Natural Language classifies the transcribed documents using your own custom labels The final process completes and the application which we have created access data quickly which improves the document search capabilities.

CHAPTER 7

SUMMARY AND CONCLUSIONS

7.1 Conclusion

This was our project of System Design about AI-Based Picture Translation App developed in Java programming language. The Development of this system takes a lot of efforts from us. We think this system gave a lot of satisfaction to all of us.

Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fruitful to us.

7.2 Future Work

To make this process work, Several companies and Google were using a machine learning technology known as generative adversarial networks, otherwise known as GAN models — the same technology that powers the Magic Eraser feature to remove objects from photos taken on the Google Pixel smartphones. This advancement will allow Google to now blend the translated text into even very complex images, making the translation feel natural and seamless, the company says. It should seem as if you're looking at the item or object itself with translated text, not an overlay obscuring the image.

The feature is another development that seems to point to Google's plans to further invest in the creation of new AR glasses, as an ability to translate text in the real world could be a key selling point for such a device. The company noted that every month, people use Google to translate text and images over a billion times in more than 100 languages. It also this year began testing AR prototypes in public settings with a handful of employees and trusted testers, it said.

While there's obvious demand for better translation, it's not clear if users will prefer to use their smartphone for translations rather than special eyewear. After all, Google's first entry into the smart glasses space, Google Glass, ultimately failed as a consumer product.

References

1. Artificial Intelligence Based Language Translation Platform Manju Kolhar* and Abdalla AL ameen-
https://www.researchgate.net/publication/350555260_Artificial_Intelligence_Based_Language_Translation_Platform
2. An android-based language translator application- International E- Conference on Data Analytics, Intelligent Systems, and Information Security & ICDIIS 2020 11-12 December 2020, Pollachi, India
3. Edge Connect: Generative Image Inpainting with Adversarial Edge Learning-K amyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, Mehran Ebrahimi Faculty of Science, University of Ontario Institute of Technology, Canada
4. Language Translator Application- International Journal for Research in Applied Science & Engineering Technology (IJRASET); July 2022 Language Translator Application | IJRASET Publication - Academia.edu
5. SIGN LANGUAGE TRANSLATOR FOR MOBILE PLATFORMS- 2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)
6. Hierarchical Recurrent Deep Fusion Using Adaptive Clip Summarization for Sign Language Translation- IEEE Transactions on Image Processing; Hierarchical Recurrent Deep Fusion Using Adaptive Clip Summarization for Sign Language Translation | IEEE Journals & Magazine | IEEE Xplore
7. English Text to Multilingual Speech Translator Using Android-International Journal of Inventive Engineering and Sciences (IJIES)- (PDF) English Text to Multilingual Speech Translator Using Android (researchgate.net)
8. A Natural Language Processing System for English to Igbo Language Translation in Android- (PDF) A Natural Language Processing System for English to Igbo Language Translation in Android (researchgate.net)
9. Design and Development of Android Based Mobile Application for Especially Abled People- Design and Development of Android Based Mobile Application for Especially Abled People | SpringerLink

APPENDIX

A. SOURCE CODE

```
package com.scanlibrary;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DialogFragment;
import android.content.DialogInterface;
import android.os.Bundle;

@SuppressLint("ValidFragment")
public class SingleButtonDialogFragment extends DialogFragment {

    protected int positiveButtonTitle;

    protected String message;

    protected String title;

    protected boolean isCancelable;

    public SingleButtonDialogFragment(int positiveButtonTitle,

                                     String message, String title, boolean isCancelable) {

        this.positiveButtonTitle = positiveButtonTitle;

        this.message = message;

        this.title = title;

        this.isCancelable = isCancelable;

    }

    @Override

    public Dialog onCreateDialog(Bundle savedInstanceState) {
```

```

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity())

        .setTitle(title)

        .setCancelable(isCancelable)

        .setMessage(message)

        .setPositiveButton(positiveButtonTitle, new
DialogInterface.OnClickListener() {

            @Override

            public void onClick(DialogInterface dialog, int which) {

                }

            });

        return builder.create();
    }
}

package com.scanlibrary;

import android.annotation.SuppressLint;
import android.content.ContentUriis;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.provider.DocumentsContract;

```

```
import android.provider.MediaStore;
```

```
/**
```

```
 * Created by Aki on 1/7/2017.
```

```
 */
```

```
public class PathUtils {
```

```
    /*
```

```
     * Gets the file path of the given Uri.
```

```
    */
```

```
    @SuppressWarnings("NewApi")
```

```
    public static String getPath(Context context, Uri uri) {
```

```
        final boolean needToCheckUri = Build.VERSION.SDK_INT >= 19;
```

```
        String selection = null;
```

```
        String[] selectionArgs = null;
```

```
        // Uri is different in versions after KITKAT (Android 4.4), we need` to
```

```
        // deal with different Uris.
```

```
        if (needToCheckUri &&
```

```
DocumentsContract.isDocumentUri(context.getApplicationContext(), uri)) {
```

```
            if (isExternalStorageDocument(uri)) {
```

```
                final String docId = DocumentsContract.getDocumentId(uri);
```

```
                final String[] split = docId.split(":");
```

```
                if (Build.VERSION.SDK_INT >= 30){
```

```
                    return Environment.getExternalStorageDirectory() + "/" + split[1];
```

```
                }else{
```

```

        return
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS)

        + "/" + split[1];

    }

} else if (isDownloadsDocument(uri)) {

    final String id = DocumentsContract.getDocumentId(uri);

    uri = ContentUris.withAppendedId(

        Uri.parse("content://downloads/public_downloads"), Long.valueOf(id));

} else if (isMediaDocument(uri)) {

    final String docId = DocumentsContract.getDocumentId(uri);

    final String[] split = docId.split(":");

    final String type = split[0];

    if ("image".equals(type)) {

        uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;

    } else if ("video".equals(type)) {

        uri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;

    } else if ("audio".equals(type)) {

        uri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;

    }

    selection = "_id=?";

    selectionArgs = new String[]{ split[1] };

}

}

if ("content".equalsIgnoreCase(uri.getScheme())) {

```

```

String[] projection = { MediaStore.Images.Media.DATA };

Cursor cursor = null;

try {

    cursor = context.getContentResolver().query(uri, projection, selection,
selectionArgs, null);

    int column_index =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

    if (cursor.moveToFirst()) {

        return cursor.getString(column_index);

    }

} catch (Exception e) {

}

} else if ("file".equalsIgnoreCase(uri.getScheme())) {

    return uri.getPath();

}

return null;

}

```

```

/**

```

```

 * @param uri The Uri to check.

```

```

 * @return Whether the Uri authority is ExternalStorageProvider.

```

```

 */

```

```

public static boolean isExternalStorageDocument(Uri uri) {

```

```

    return "com.android.externalstorage.documents".equals(uri.getAuthority());

```

```
}
```

```
/**
```

```
 * @param uri The Uri to check.
```

```
 * @return Whether the Uri authority is DownloadsProvider.
```

```
*/
```

```
public static boolean isDownloadsDocument(Uri uri) {
```

```
    return "com.android.providers.downloads.documents".equals(uri.getAuthority());
```

```
}
```

```
/**
```

```
 * @param uri The Uri to check.
```

```
 * @return Whether the Uri authority is MediaProvider.
```

```
*/
```

```
public static boolean isMediaDocument(Uri uri) {
```

```
    return "com.android.providers.media.documents".equals(uri.getAuthority());
```

```
}
```

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

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    package="com.example.translate_objecttext">
```

```
    <uses-permission android:name="android.permission.INTERNET" />
```

```
    <uses-permission
```

```
        android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<application
```

```
    android:allowBackup="true"
```

```
    android:hardwareAccelerated="true"
```

```
    android:icon="@drawable/icon"
```

```
    android:label="@string/app_name"
```

```
    android:largeHeap="true"
```

```
    android:requestLegacyExternalStorage="true"
```

```
    android:preserveLegacyExternalStorage="true"
```

```
    android:roundIcon="@drawable/icon"
```

```
    android:supportsRtl="true"
```

```
    android:theme="@style/Theme.Translate_ObjectText"
```

```
    android:usesCleartextTraffic="true">
```

```
<meta-data
```

```
    android:name="com.google.mlkit.vision.DEPENDENCIES"
```

```
    android:value="ocr" />
```

```
<activity android:name="com.scanlibrary.ScanActivity" />
```

```
<activity
```

```
    android:name=".ObjectTextCurved"
```

```
    android:exported="true">
```

```
<intent-filter>

    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<provider

    android:name="androidx.core.content.FileProvider"

    android:authorities="${applicationId}.provider"

    android:exported="false"

    android:grantUriPermissions="true">

    <meta-data

        android:name="android.support.FILE_PROVIDER_PATHS"

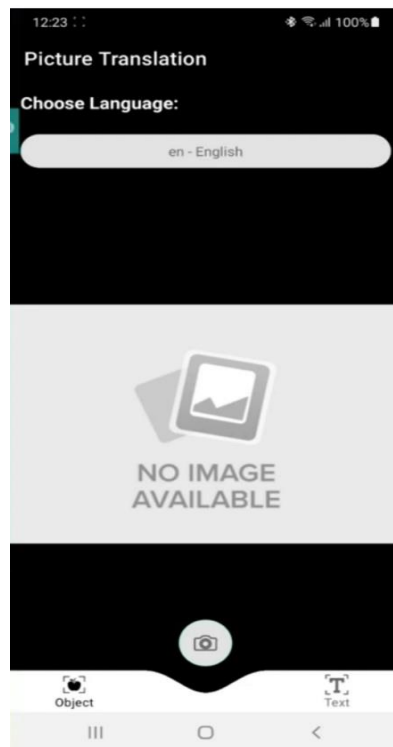
        android:resource="@xml/provider_paths" />

    </provider>

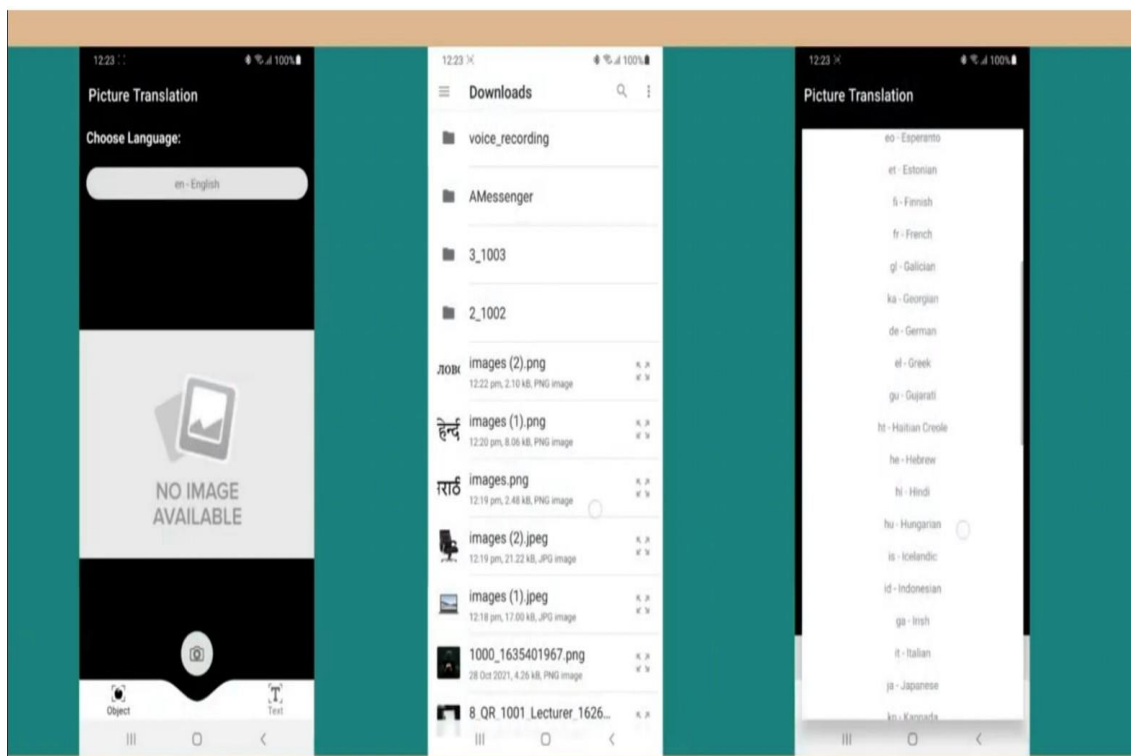
</application>

</manifest>
```

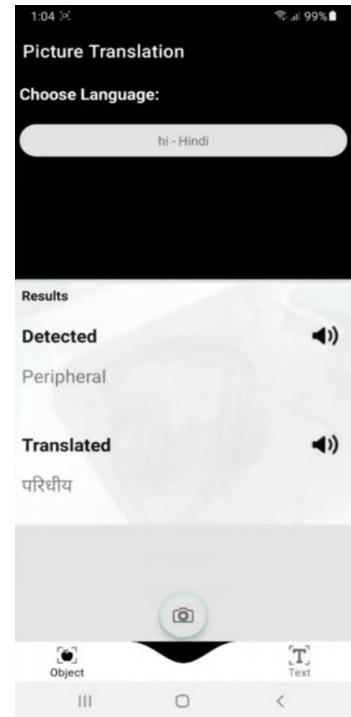
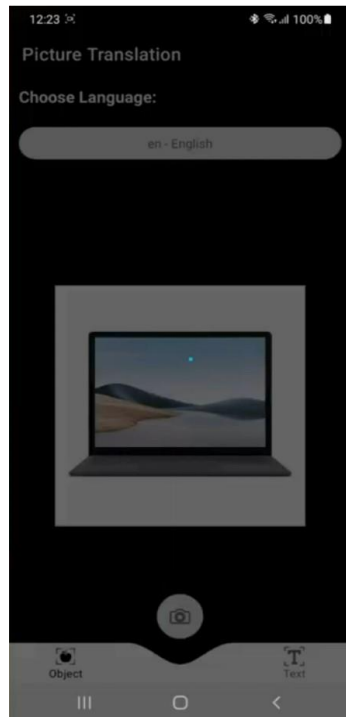
B. SCREEN SHOTS



The translation app allows you to translate various languages into text or speech



To detect object, we use google object detection from google cloud vision API, which is a part of ML.



Similarly for text recognition we use text detection from google cloud vision as a part of ML.

