Text Recognition, Object detection and Language Translation App

submitted in partial fulfilment of the requirements

of the degree of

B.E. Information Technology

by

Arti Jha - 32

Amruta Khot - 43

Chaitra Naik - 60

Under the guidance of

Prof. Sejal D'mello

Department of Information Technology

Atharva College of Engineering, Malad(W)

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

University of Mumbai

2021–2022

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

AET'S

ATHARVA COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**Text Recognition, Object detection and Language Translation App**" is a bonafide work of "**Arti Jha" (32), "Amruta Khot" (43) and "Chaitra Naik" (60)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Information Technology

Supervisor/Guide **External Examiner**

College Seal

Internal Examiner **Head of Department** **Principal**

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

Project Report Approval for B.E.

This project report entitled ***Text Recognition, Object detection and Language Translation App*** by ***Arti Jha, Amruta Khot*** and ***Chaitra Naik*** is approved for the degree of ***B.E. in Information Technology.***

Examiners

1.....

2.....

Date:

Place:

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Arti Jha (32)

Amruta Khot (43)

Chaitra Naik (60)

Date:

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

Abstract

This study presents a comprehensive review of OCR, Translation, and Object Detection Research from a single image. With the fast advancement of deep learning, more powerful tools that can learn semantic, high-level, and deeper features have been proposed to solve the issues that plague traditional systems. The rise of high-powered desktop computers has aided OCR reading technology by permitting the creation of more sophisticated recognition software that can read a range of common printed typefaces and handwritten texts. However, implementing an OCR that works in all feasible scenarios and produces extremely accurate results remains a difficult process. Object detection is also the difficult problem of detecting various items in photographs. Object identification using deep learning is a popular use of the technology, which is distinguished by its superior feature learning and representation capabilities when compared to standard object detection approaches. The major focus of this review paper is on text recognition, object detection, and translation from an image-based input application employing OCR and the YOLO technique.

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

Contents

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

List of Figures

List of Tables

List of Abbreviations

Sr No.	Abbreviation	Expanded form
1	OCR	Optical Character Recognition
2	YOLO	You Only Look Once

Chapter 1

Introduction

This project involves three different features which are OCR, object detection and language translation and all these features are embedded together.

1.1 Description

Many applications based on OCR, language translation, and object identification have been seen. However, the majority of applications do not provide all of these functionalities. All of these characteristics have been included into this system. On the app's main screen, the user will be presented with three alternatives. Text recognition, object detection, and language translation are the three possibilities. Any essential option can be selected by the user. There are two alternative possibilities for text recognition. The user may either choose a picture from the gallery or click on a live image. After the user has provided input, the user must pick a language for translation and click submit. Object detection is the app's second feature. The user is given two alternatives here as well. The user can either choose a picture for the gallery or click on a live image. After the object has been discovered, the user may translate the object's name into whatever language they like. Only language translation is the last option, which requires the user to compose a paragraph or sentence and pick a language for translation.

1.2 Problem Formulation

To overcome from the language barrier among countries and also states within the country, Suppose you are planning to visit some place or say country where people speak different language from whatever you speak, and let's say you actually went to visit that place and you saw a banner but you can't figure out what was written over there, and say you need to know what was written over there in order to take next step, this is where you might get in trouble right?, but with our project you need not to be worried about all of these thing, just on taking picture of that banner, you can actually translated whatever written there in banner in your preferred language. Now let’s suppose you are in some random country where people speak some random languages, and say you want to know what a bird is called in their language. The cool way to do this is just click the picture of the bird and know what that is called in any

random language. So the main thing is, you can detect the object and translate that object name in any preferred language.

1.3 Motivation

During Business process, translation of pictures is also a difficult problem faced by international and domestic companies. Also, to overcome the problem of language barrier among countries and also states within the country, the proposed system will perform various features in an application. Text recognition has gained a lot of prominence in recent years as it has entered into a large arena of applications such as in automatic reading of licence plate, signboards. Object detection based on deep learning has achieved very good performance. Object detection is a profound computer vision technique that focuses on identifying and labelling objects within images, videos, and even live footage. However, there are many problems with images in real world shooting such as noise, blurring and rotating jitter etc. These problems have an impact on object detection. Issues of images captured in the real scene include the instability of the camera due to this captured image being blurred.

1.4 Proposed Solution

The three components of our project are OCR and language translation, object detection and language translation, and merely language translation. As a result, the user must first choose one of the aforementioned possibilities. We used OCR for text recognition, which was imported via tesseract. Tesseract employs a two-step process known as adaptive recognition. Character recognition is the first phase, and there are three sub-steps in this step as well. Image pre-processing is the initial stage. Images are preprocessed in this stage to increase the likelihood of successful recognition. Unwanted distortions are reduced, and certain visual characteristics are accentuated in this stage. The next two stages rely heavily on this phase. The actual recognition of character is the second sub step. It is based on the feature extraction idea. When the initial input is too vast to handle, just a subset of features is chosen. The features that are not picked are redundant, but the ones that are selected are critical. The performance is improved by using the smaller set of data instead of the initial huge one. The final sub step is picture post-processing. It is another high-accuracy error correcting approach. The tesseract's second step is to fill in any missing letters with letters that fit the word or phrase context. The text will be submitted to the language translation library, which will be imported using "googletrans" after it has been identified.

The YOLO method is used to detect objects in the second section of the app. The YOLO algorithm's first step is to partition the entire image into grids. There are seven vectors connected with each of the grid cells. Probability of the class, bounding box x, bounding box y, bounding box width, bounding box height, and classes are the vectors. As a result, anytime we come across an object grid cell at that moment, we check for the centroid first. Even if parts of two separate objects are present in a single grid, the centroid of whatever item is present in that grid is linked with that picture. If each grid is 4x4, for example, the real size of the grid becomes 4x4x7, where 4x4 is the grid size and each grid has 7 vectors. As a result, the train and test datasets are created. The picture is present in the train dataset, whereas vectors are present in the test dataset. We've made projections based on this. If the user desires to translate the object name into another language after it has been identified, the user must first pick a language for translation, which will be handled by the “googletrans” library.

The app's last feature is language translation, which requires the user to compose a sentence or a paragraph in any desired language and then pick a language for translation via the “googletrans” library.

1.4 Scope

- This project includes image extraction, recognition and language translation along with object detection in one system so the user doesn’t have to download different applications to use these features.
- All the tools used for this project are free of cost.
- This project uses the latest version of YOLO algorithm which is YOLOv4 so the user won’t face image localization problems.

Chapter 2

Review of literature

Sr No	Title	Authors	Publications	Approach
1.	Document Segmentation and Language Translation using Tesseract-OCR	Sahil Thakare, Ajay Kamble, Vishal Thengne, Mrs.U.R.Kamble	2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)	This paper presents details about translation in terms of a web application that accepts image document as an input using Python tesseract library and does its exact translation using Google Translator.
2.	Numerical Simulation Technology Study on Automatic Translation of Foreign Language Images Based on Tesseract-OCR	Gaohe Li1 Xinhao Li2 Bo Xu3	2019 International Conference on Robots & Intelligent System (ICRIS)	The author uses Tesseract-OCR automatic image recognition technology and OpenCV software library to effectively solve the problem of automatic image recognition and translation in foreign language materials.
3.	Object Detection Based on YOLO Network	Chengji Liu1, Yufan Tao1, Jiawei Liang1, Kai Li1, Yihang Chen1	2018 IEEE 4th Information Technology and Mechatronics Engineering Conference	In this paper, they simulated different degenerative processes of images for analysis and research. Firstly, they established the

			(ITOEC 2018)	models of degraded images. Then, they used mathematical models to train the network to adapt to the complex real-world environment.
4.	YOLO-LITE: A Real-Time Object Detection Algorithm Optimised for Non-GPU Computers	Rachel Huang, Jonathan Pedoeem, Cuixian Chen	2018 IEEE International Conference on Big Data (Big Data)	This paper focuses on YOLO-LITE, a real-time object detection model developed to run on portable devices such as a laptop or cellphone lacking a Graphics Processing Unit (GPU).

Table 2.1 Review of Literature

Chapter 3

System Analysis

3.1 Functional Requirements

ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application shall be free to be downloaded.

DEP: None

ID: FR2

TITLE: Mobile application - Run

DESC: Given that a user has downloaded the mobile application, then the user should be able to run the mobile application.

DEP: FR1

ID: FR3

TITLE: Run – Welcome Screen

DESC: Given that a user run the mobile application, then the first page that is shown shall be the Welcome screen.

Application shall display “OCR” menu.

Application shall display “Object detection” menu.

Application shall display “Language Translation” menu.

DEP: FR2

ID: FR4

TITLE: Welcome Screen - OCR

DESC: Given that a user has selected the OCR menu on Welcome Screen, then the user shall be prompted to choose input and output language.

DEP: FR3

ID: FR5

TITLE: Camera or Gallery

DESC: Given that a user has selected the Camera or Gallery menu, camera shall be activated and user should take a photograph of the text or should select image of text from Gallery.
DEP: FR4

ID: FR6

TITLE: Translate

DESC: Given that a user has selected the Translate option, then the application shall display translated text.

DEP: FR5

ID: FR7

TITLE: Welcome Screen – Object Detection

DESC: Given that a user has selected the Object Detection menu on Welcome Screen, then the user shall be prompted to choose Camera or Gallery.

DEP: FR3

ID: FR8

TITLE: Camera or Gallery

DESC: Given that a user has selected the Camera or Gallery menu, camera shall be activated and user should take a photograph of the object or should select image of object from Gallery. Then a user needs to adjust the image. Once that is done then the application shall display the detected object name.

DEP: FR7

ID: FR9

TITLE: Welcome Screen – Language Translation

DESC: Given that a user has selected the Language Translation menu on Welcome Screen, then the user shall be prompted to choose input and output language.

DEP: FR3

ID: FR10

TITLE: Type in input text

DESC: Given that the user has typed text in the section provided.

DEP: FR9

ID: FR11

TITLE: Translate

DESC: Given that a user has selected the Translate option, then the application shall display translated text.

DEP: FR10

3.2 Non-Functional Requirements

Feasibility

Desc:This project includes image extraction, recognition and language translation along with object detection in one system so the user doesn't have to download for the different applications. All the tools used for the project are free of cost. This project uses the latest version of YOLO algorithm which is YOLO V4 so the user won't face image localization problems.

Performance

Desc:It is tough to make a clear comparison between different object detection methods. So, we can't give a straight decision on the best model. For many real-life applications, we make choices to create an equilibrium of accuracy with speed. Therefore, we need to be aware of other characteristics that have a significant impact on performance. We have implemented a yolo algorithm for object detection. So the accuracy performance of YOLO algorithms is compared with faster RCNN and As per the result which we obtained, we can say that YOLO v4 takes less time compared to Faster R-CNN, and other YOLO algorithms.

Speed

Desc:YOLO v4 which takes less running time when examined with YOLO v3 and RCNN.We have thoroughly examined the characteristics of two procedures for speed of object detection, which are faster RCNN, And YOLO v4. The speed performance of YOLO algorithms is compared with faster RCNN . As per the result, we can say that YOLO v4 takes less time compared to Faster R-CNN, and other YOLO algorithms.

3.3 Specific Requirements

Hardware Requirements	Software Requirements
Android device	Anaconda (Jupyter Notebook)
	Android Studio
	Flutter
	Android OS (minimum Android API 16 : 4.1 Jellybean)
	Minimum RAM 500 MB.

Table 3.3 Specific Requirements

3.4 Use Case diagram and description

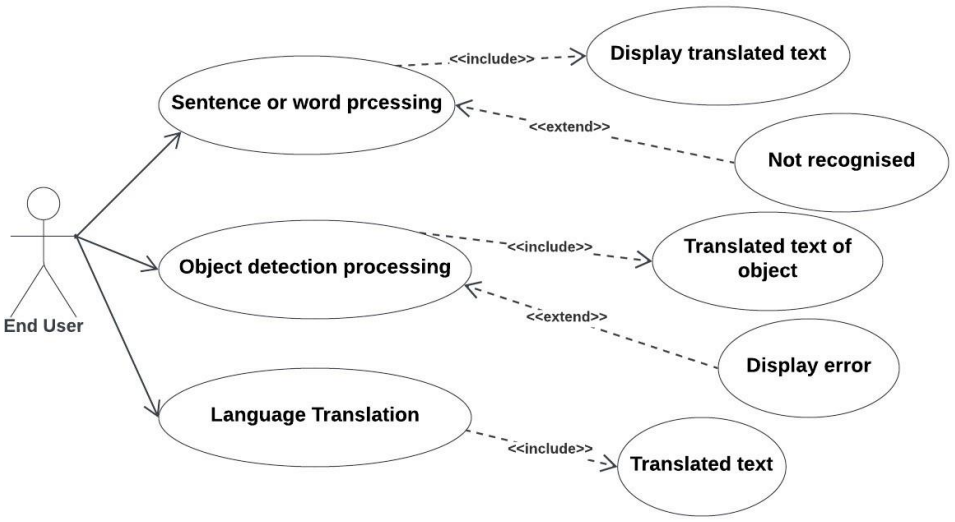

Figure 3.4 Use Case Diagram

Chapter 4

Analysis Modelling

4.1 Activity Diagram

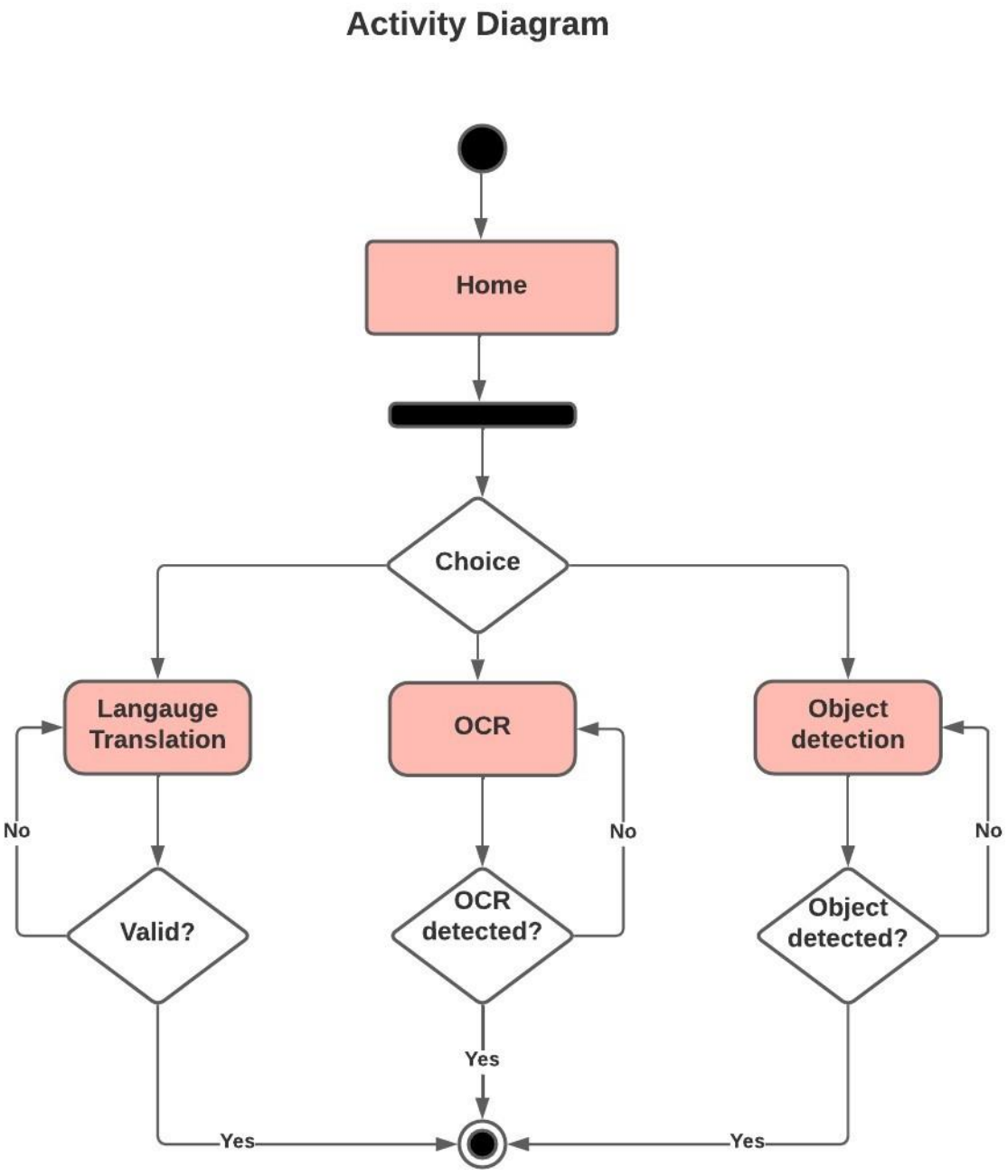

Figure 4.1 Activity Diagram

4.2 Functional Modelling

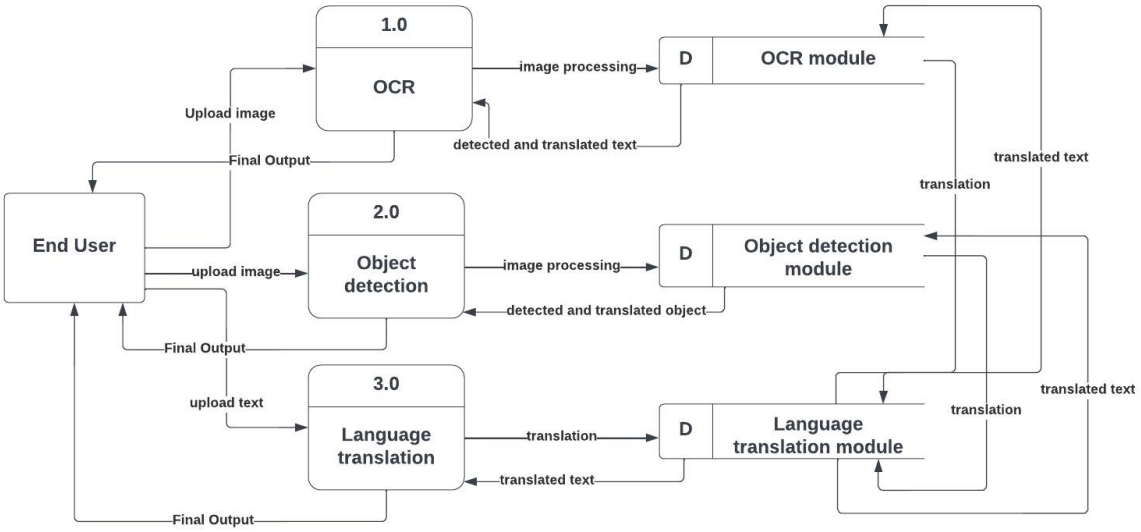

Fig 4.2 Data Flow Diagram

4.3 TimeLine Chart

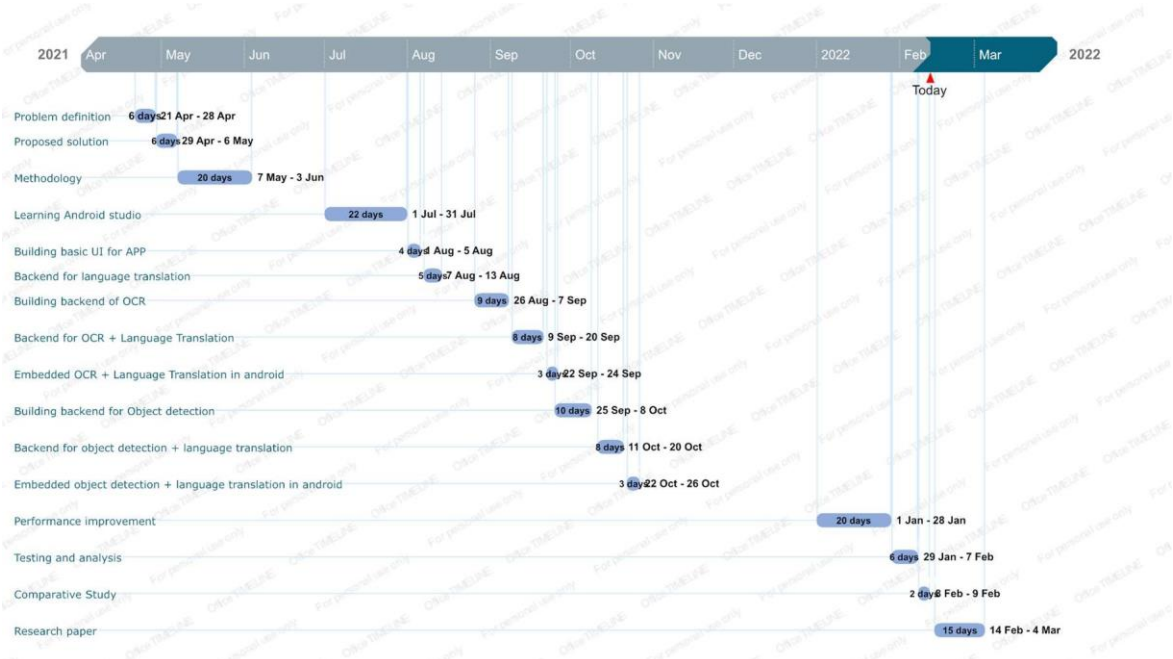

Figure 4.3 TimeLine Chart

Chapter 5

Design

5.1 Architectural Design

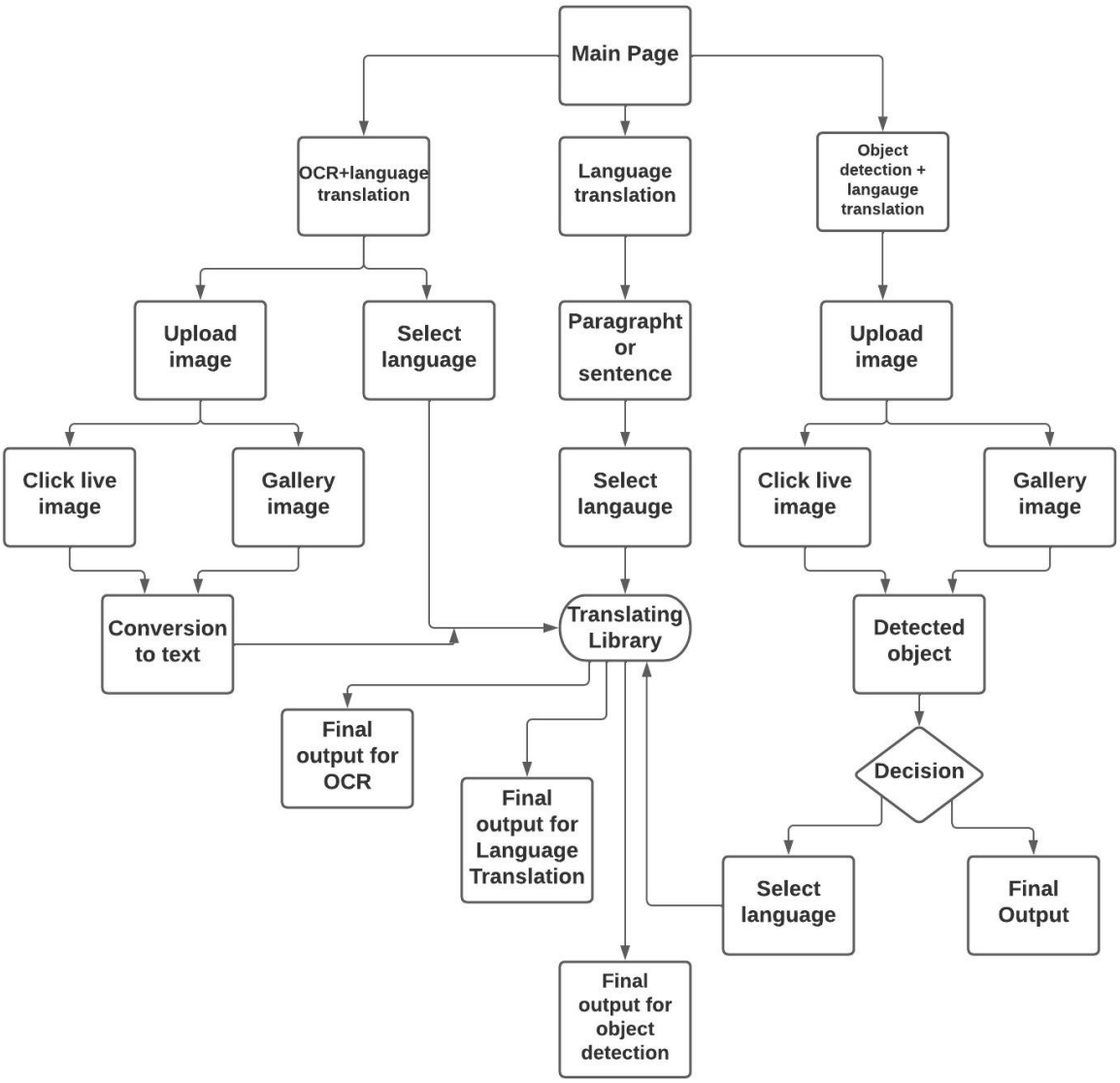

Figure 5.1 Block Diagram

Many applications based on OCR, language translation, and object identification have been seen. However, the majority of applications do not provide all of these functionalities. All of these characteristics have been included into this system. On the app's main screen, the user will be presented with three alternatives. Text recognition, object detection, and language translation are the three possibilities. Any essential option can be selected by the user. There

are two alternative possibilities for text recognition. The user may either choose a picture from the gallery or click on a live image. After the user has provided input, the user must pick a language for translation and click submit.

Object detection is the app's second feature. The user is given two alternatives here as well. The user can either choose a picture for the gallery or click on a live image. After the object has been discovered, the user may translate the object's name into whatever language they like. Only language translation is the last option, which requires the user to compose a paragraph or sentence and pick a language for translation.

5.2 User Interface Design

Figure 5.2.1 Main Page

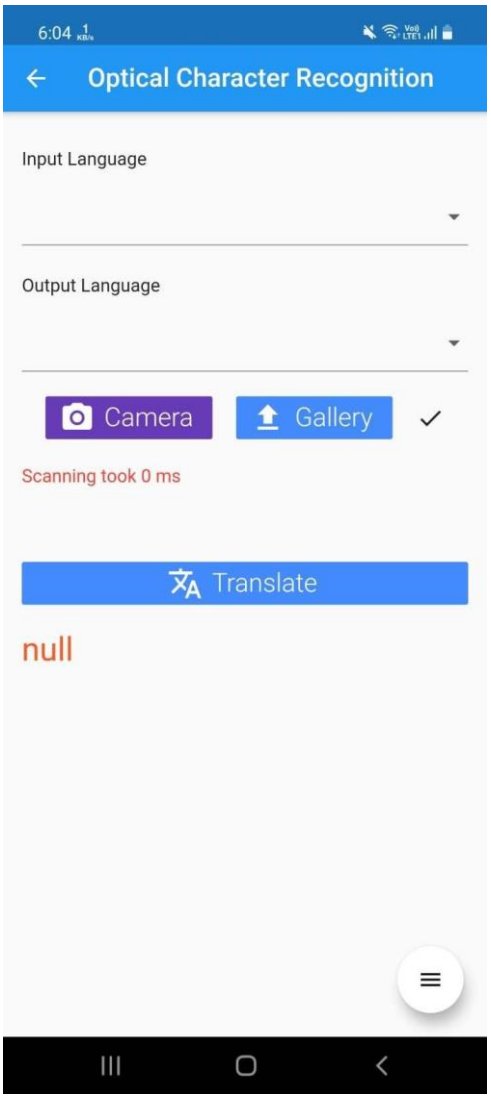

Figure 5.2.2 OCR

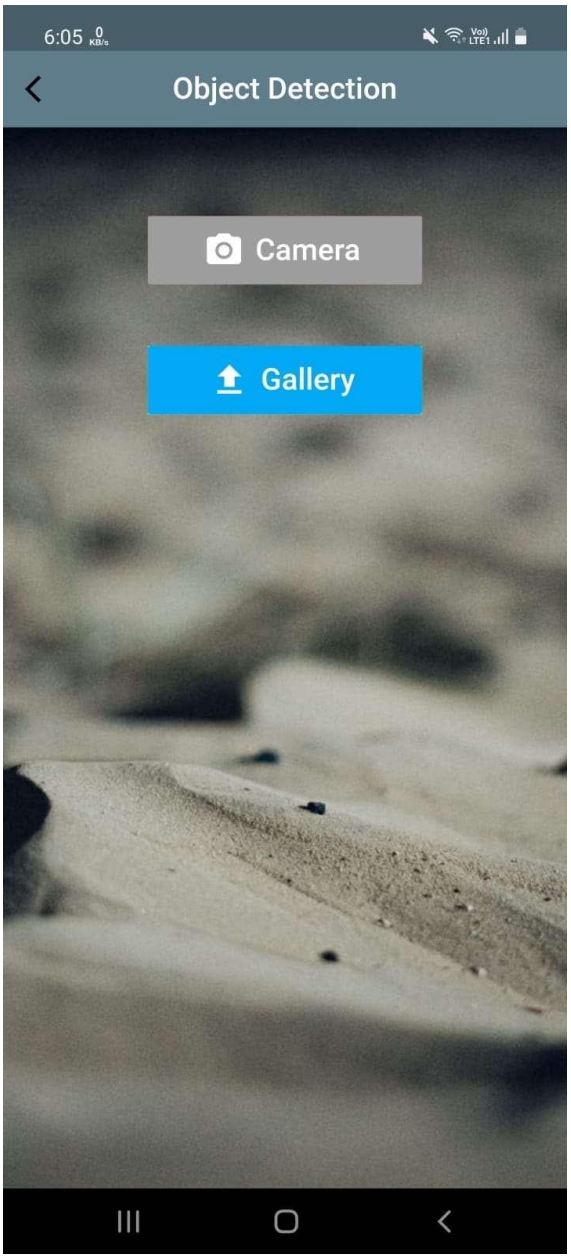

Figure 5.2.3 Object detection

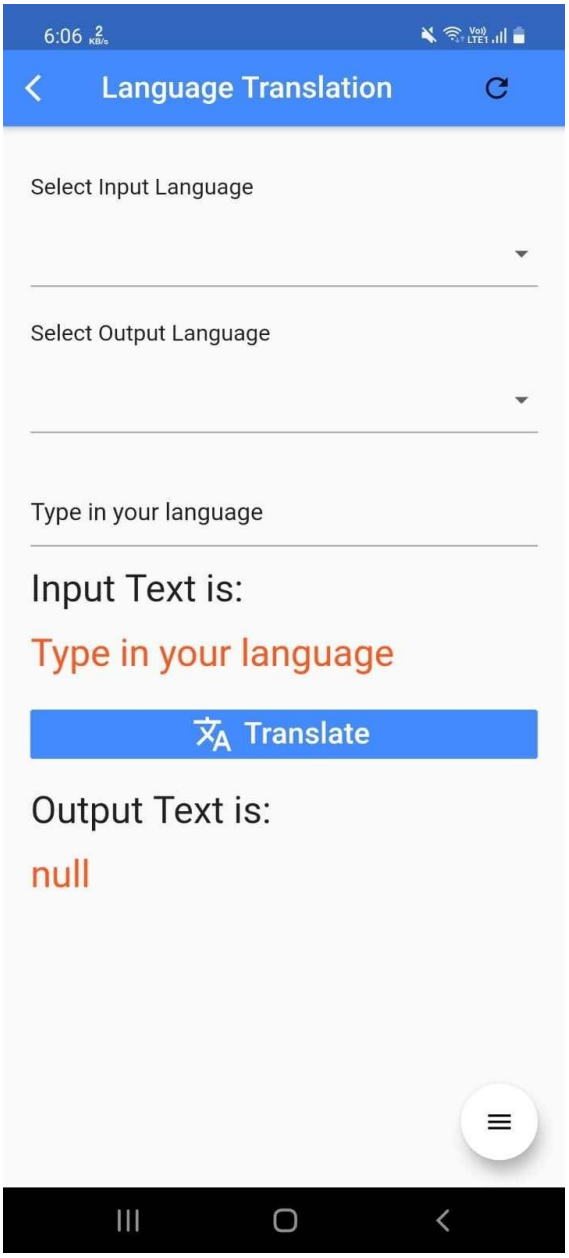

Figure 5.5.4 Language Translation

Chapter 6

Implementation

6.1 Algorithms

Optical Character Recognition

Optical character recognition (OCR) converts scanned images into text, allowing you to convert paper documents into editable, searchable digital files. This can significantly reduce the amount of physical space needed to store documents while also vastly improving document operations. It eliminates the need to manually process documents or rekey information, which can lead to errors, and it decreases the risk of lost or misfiled papers. It also has the potential to lower the costs of manual handling.

To convert a scanned image into text, OCR analyses the patterns of light and dark that make up the letters and numerals. Because OCR systems must recognise characters in a variety of fonts, rules are used to assist the system in matching what it sees in the image to the appropriate letters or numbers.

It's critical to scan the clearest version of the page possible for OCR to perform properly. Errors can be caused by blurred text or markings on the copy.

Character after character, OCR systems recognise the text, but the results are so quick that they appear to be immediate. You can check for mistakes as you go or at the conclusion of the procedure, and some tools will detect them automatically.

YOLO

The YOLO algorithm works by dividing the image into N grids, each having an equal dimensional region of SxS. Each of these N grids is responsible for the detection and localization of the object it contains.

Correspondingly, these grids predict B bounding box coordinates relative to their cell coordinates, along with the object label and probability of the object being present in the cell. This process greatly lowers the computation as both detection and recognition are handled by cells from the image, but It brings forth a lot of duplicate predictions due to multiple cells predicting the same object with different bounding box predictions.

YOLO makes use of Non Maximal Suppression to deal with this issue. In Non Maximal Suppression, YOLO suppresses all bounding boxes that have lower probability scores. YOLO achieves this by first looking at the probability scores associated with each decision and taking the largest one. Following this, it suppresses the bounding boxes having the largest Intersection over Union with the current high probability bounding box. This step is repeated till the final bounding boxes are obtained.

6.2 Working of the project

OCR

```
import './camera_or_gallery.dart';
import './translatorOnly.dart';
import 'package:flutter/material.dart';
import 'package:flutter_speed_dial/flutter_speed_dial.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
import 'dart:async';
import 'dart:convert';
import 'dart:io';
import 'dart:typed_data';
import 'package:flutter/services.dart';
import 'package:image_picker/image_picker.dart';
import 'package:image_cropper/image_cropper.dart';
import 'package:translator/translator.dart';
import 'package:http/http.dart' as http;
import 'package:dio/dio.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class OCR extends StatefulWidget {
 String uid, url;
 OCR({this.uid, this.url});
 @override
 _OCRState createState() => _OCRState(uid: uid, url: url);
}

class _OCRState extends State<OCR> {
 String uid, url;
 _OCRState({this.uid, this.url});
 bool fordec = false;
 String outputText = "Nothing";
 String languagein = 'eng';
 String selectLan;
```

```
String languageout = 'eng';
String inputLanguage = "";
String outputLanguage = "";
Translation output;
final List<String> lang = ["Nepali", "Hindi", "Punjabi"];
File imPth;
bool _scanning = false;
var _extractText;
int _scanTime = 0;

void _uploadFile(filePath, lang) async {
  // Get base file name
getImage(ImageSource source) async {
  File img = await ImagePicker.pickImage(source: source);
  if (img != null) {
    File cropped = await ImageCropper.cropImage(
        sourcePath: img.path,
        aspectRatio: CropAspectRatio(ratioX: 1, ratioY: 1),
        compressQuality: 100,
        maxHeight: 700,
        maxWidth: 700,
        compressFormat: ImageCompressFormat.jpg,
        androidUiSettings: AndroidUiSettings(
            toolbarColor: Colors.deepOrange,
            toolbarTitle: "Adjust Image",
            statusBarColor: Colors.deepOrange.shade900,
            backgroundColor: Colors.white));
    return cropped;
  }
  setState(() {
    imPth = img;
  });
}
```

Object detection

```
import './translatorOnly.dart';
import 'package:flutter/material.dart';
import 'package:translator/translator.dart';
import '../style/theme.dart' as Theme;
import 'dart:io';
import 'package:path/path.dart';
import 'package:dio/dio.dart';

class Translator extends StatefulWidget {
```

```
 var filePath;
 String text, urlLink;
 final String uid;
 Translator({this.text, this.uid, this.filePath, this.urlLink});
 @override
 _TranslatorState createState() => _TranslatorState(
    text: text, uid: uid, filePath: filePath, urlLink: urlLink);
}

class _TranslatorState extends State<Translator> {
 var filePath;
 final String uid, urlLink;
 String outputLanguage = "English";
 String languageout = "English";
 var output;
 String text;
 final List<String> lang = ["Nepali", "Hindi", "Punjabi"];
 _TranslatorState({this.text, this.uid, this.filePath, this.urlLink});

 language() async {
  if (languageout == "Nepali") {
   setState(() {
    outputLanguage = "nep";
   });
  }
  if (languageout == "Hindi") {
   setState(() {
   outputLanguage = "hin";
   });
  }
  if (languageout == "Punjabi") {
   setState(() {
    outputLanguage = "pan";
   });
  }
  try {
   FormData formData =
       FormData.fromMap({"image": await MultipartFile.fromFile(filePath)});

   Response response = await Dio()
       .post(urlLink + "upload/detect/" + outputLanguage, data: formData);
   print("File upload response: $response");
```

Language Translation

```
import './home.dart';
import './ocr.dart';
import 'package:flutter/material.dart';
import 'package:flutter_speed_dial/flutter_speed_dial.dart';
import 'package:translator/translator.dart';
import '../style/theme.dart' as Theme;
import 'camera_or_gallery.dart';

class TranslatorOnly extends StatefulWidget {
 final String uid;
 TranslatorOnly({this.uid});
 @override
 _TranslatorOnlyState createState() => _TranslatorOnlyState(uid: uid);
}

class _TranslatorOnlyState extends State<TranslatorOnly> {
 final String uid;
 _TranslatorOnlyState({this.uid});

 String outputLanguage = "English";
 String inputLanguage = "English";
 String languagein = "English";
 String languageout = "English";
 String text = "Type in your language";
 Translation output;

 final List<String> lang = ["English", "Nepali", "Hindi", "Punjabi"];

 language() async {
   if (languagein == "English") {
     if (languageout == "Hindi") {
       setState(() {
         inputLanguage = "en";
         outputLanguage = "hi";
       });
     }
     if (languageout == "Nepali") {
       setState(() {
         inputLanguage = "en";
         outputLanguage = "ne";
       });
     }
     if (languageout == "Punjabi") {
```

```
    setState(() {
      inputLanguage = "en";
      outputLanguage = "pa";
    });
  }
}
//////////////////////////////////////////////////////////////////
if (languagein == "Hindi") {
  if (languageout == "Nepali") {
    setState(() {
      inputLanguage = "hi";
      outputLanguage = "ne";
    });
  }
  if (languageout == "Punjabi") {
    setState(() {
      inputLanguage = "hi";
      outputLanguage = "pa";
    });
  }
  if (languageout == "English") {
    setState(() {
      inputLanguage = "hi";
      outputLanguage = "en";
    });
  }
}

if (languagein == "Nepali") {
  if (languageout == "Hindi") {
  setState(() {
      inputLanguage = "ne";
      outputLanguage = "hi";
    });
  }
  if (languageout == "Punjabi") {
    setState(() {
      inputLanguage = "ne";
      outputLanguage = "pa";
    });
  }
  if (languageout == "English") {
    setState(() {
      inputLanguage = "ne";
```

```
        outputLanguage = "en";
      });
    }
  }

  if (languagein == "Punjabi") {
    if (languageout == "Hindi") {
    setState(() {
        inputLanguage = "pa";
        outputLanguage = "hi";
      });
    }
    if (languageout == "Nepali") {
      setState(() {
        inputLanguage = "pa";
        outputLanguage = "ne";
      });
    }
    if (languageout == "English") {
      setState(() {
        inputLanguage = "pa";
        outputLanguage = "en";
      });
    }
  }

  final translator = GoogleTranslator();
  var translation = await translator.translate(text,
      from: inputLanguage, to: outputLanguage);
  setState(() {
    output = translation;
  });
}
```

Chapter 7

Testing

7.1 Test Cases

SR No	Test objective	Condition for test-success	Expected result	Test result
1.	To upload image to OCR using gallery	Access to the gallery	Successful access to the system	As expected
2.	To upload image to OCR by clicking a live picture	Access to the camera	Successful access to the camera	As expected
3.	To test OCR	Good image quality	Scan complete image without distortion	As expected
4.	To upload image for object detection through gallery	Access to the gallery	Successful access to the system	As expected
5.	To upload image for object detection by taking a live picture	Access to the camera	Successful access to the camera	As expected
6.	To test Object detection	Good image quality	Single bounding box	As expected

Table 7.1 Test cases

7.2 Types of Testing used

Unit Testing:

Unit testing is the process of checking small pieces of code to ensure that the individual parts of a program work properly on their own, speeding up testing strategies and reducing wasted tests.

Performance Testing:

Performance testing examines the speed, stability, reliability, scalability, and resource usage of a software application under a specified workload. The focus of Performance Testing is checking a software program's

- Speed – Determines whether the application responds quickly For speed, it checks how much time is required to upload an image or to detect an object from an image?
- Scalability – Determines maximum user load the software application can handle. Here, it determines whether the software product is stable in case of varying workloads.

Load Testing:

This type of non-functional software testing process determines how the software application behaves while being accessed by multiple users simultaneously. For example, it examines, is it possible for multiple users to access the same application at the same time or not?

User Interface Testing:

Type of testing which is performed to check how user-friendly the application is. It is performed by testing teams. Basically, it checks how easily one can explore various features offered by our application? (Such as text recognition, language translation and object detection features offered by our application)

Functional Testing:

Functional testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements. Basically, it checks whether the application is satisfying all the business cases or not? In our application, it will verify working of all the features. One of them is object detection where the image upload option from Camera or Gallery is tested.

Chapter 8

Results Analysis and Discussions

Following are the screenshots of the interface and output of the proposed system.

Figure 8.1 Result of OCR and Language translation

The operation of Optical Character Recognition is depicted in Fig 8.1. The user must first choose the input and output languages, after which the picture for character recognition must be added. Once the picture has been added, it will recognize the text using OCR and translate it into the output language using the "googletrans" library, before displaying the recognized text.

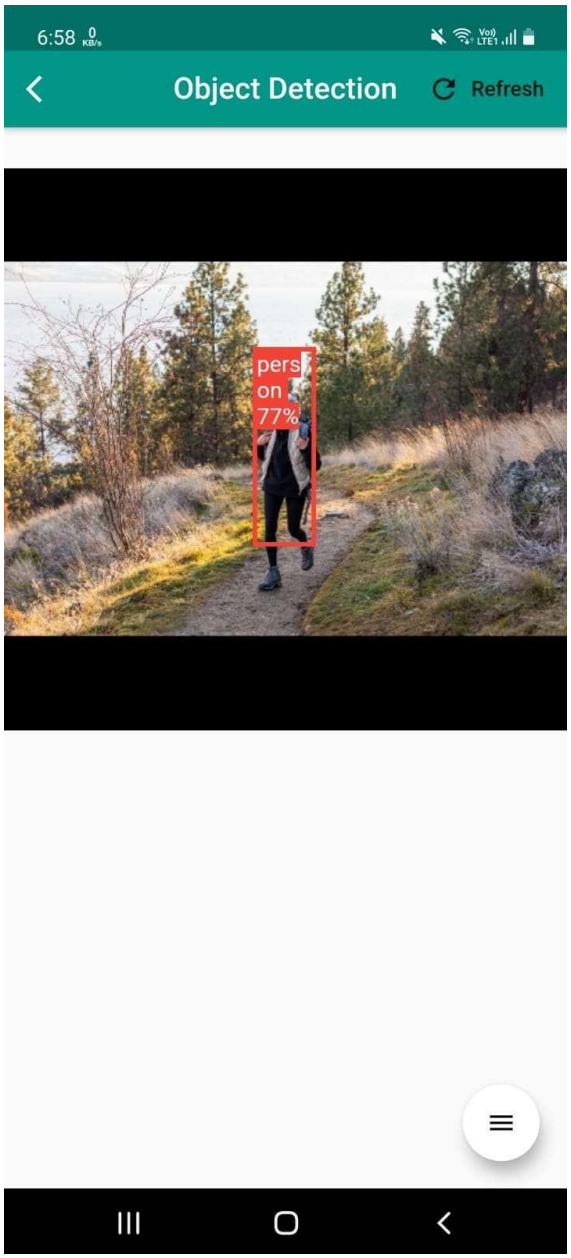

Figure 8.2 Result of object detection

Object detection is depicted in the above figure. We utilized the COCO dataset, which comprises roughly 328k photos, for object detection. YOLOv4 has been utilized. When compared to YOLOv3, YOLOv4 has a higher mAP (mean average precision). The major motivation for employing this approach was a 10% increase in mean average accuracy (mAP). YOLOv4 has an average precision of 38 to 44, while YOLOv3 has an average precision of 31 to 33.

Figure 8.3 Result of object detection after language translation

The translated text of object detection is shown in the above figure. The user is given the option of translating the text that was found throughout the detection process. The user must first pick a language and then click the Translate button. The translation activity is started in the backend and utilizes the “googletrans” library to translate the content.

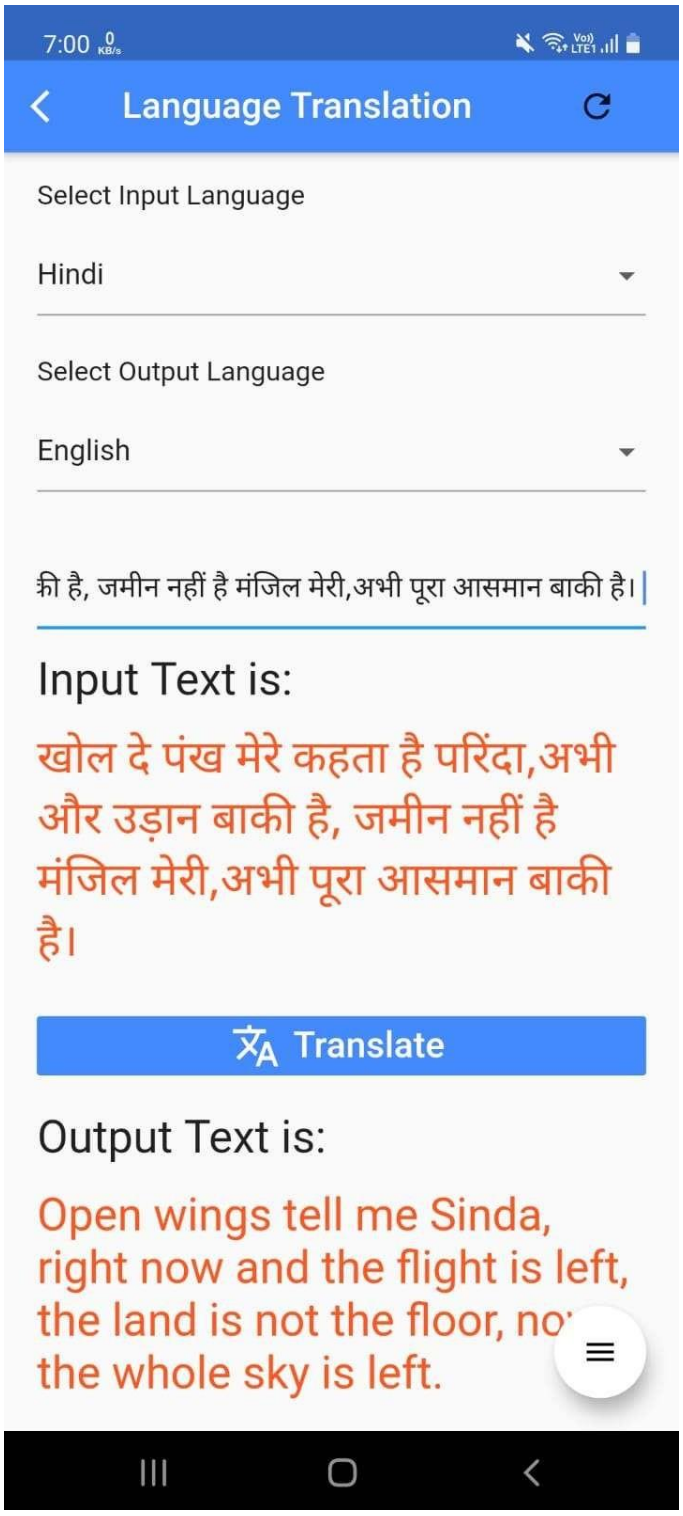

Figure 8.4 Result of language translation

Figure 8.4 depicts how Language Translation works. The user must first choose the input and output languages before adding the text to be translated. When text is entered, the "googletrans" library is used to translate it. Finally, the translated text is shown.

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

For both characteristics, the created program can perform text recognition, object identification, and language translation into a chosen language with high accuracy. This application may be improved to handle the issue of translating pdfs and other documents from one language to another.

9.2 Future Scope

1. This project may be improved by converting detected text to editable text, allowing the user to amend the text that was identified from an image before translating it.
2. The identified text may be turned into voice in a variety of languages.
3. The project may be improved to deal with realtime data instead of labels from a prepared dataset for object detection.

References

[1]Thakare, Sahil, et al. "Document Segmentation and Language Translation Using Tesseract-OCR." 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2018.

[2]Li, Gaohe, Xinhao Li, and Bo Xu. "Numerical Simulation Technology Study on Automatic Translation of Foreign Language Images Based on Tesseract-ORC." 2019 International Conference on Robots & Intelligent System (ICRIS). IEEE, 2019.

[3]Liu, Chengji, et al. "Object detection based on YOLO network." 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018.

[4]K. Elissa, Hiral Modi,M.C.parikh, “A Review On Optical Character Recognition Techniques”, International Journal of Computer Application,2017.

[5]Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[6]Memon, Jamshed, et al. "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)." IEEE Access 8 (2020): 142642-142668.

[7]Du, Juan. "Understanding of object detection based on CNN family and YOLO." Journal of Physics: Conference Series. Vol. 1004. No. 1. IOP Publishing, 2018.

[8]Tao, Jing, et al. "An object detection system based on YOLO in traffic scene." 2017 6th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2017.

[9]Fang, Wei, Lin Wang, and Peiming Ren. "Tinier-YOLO: A real-time object detection method for constrained environments." IEEE Access 8 (2019): 1935-1944.

Publication

International Research Journal of Engineering and Technology-

- Chaitra Naik, Amruta Khot, Arti Jha and Sejal D'mello (2022, March) "Text Recognition, Object Detection and Language Translation App", Atharva College of Engineering, Mumbai, India.

Acknowledgements

We feel great pleasure that we present the report on our project work at the end of eighth semester. We take this opportunity to share a few words of gratitude to all those who have supported us in making it possible. We extend our heartfelt gratitude to our project guide Prof. Sejal D'mello for her continuous guidance and approachability. We would like to express our gratitude towards her constant encouragement, support and guidance throughout the development of the project. Our regular meetings proved to be a boon in the timely completion of this stage of the project.