Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction of project

The project SmartEye: Real time text recognition and translation application for Indic Languages is an android application specially developed for the tourists. The main intention behind developing this application is to overcome the common problem of language barrier faced by the tourists while they travel other countries. The tourists are not able to read the signboards written in Marathi, Kannada or any other languages. The motivation behind implementing this real time text translation mobile application is to help tourists navigate in a foreign language environment (English) and overcome language barriers. This application translates the text written in Devanagari script i.e.; from Marathi script to English transliteration. According to a recent survey, almost 48, 15,421 foreign tourists visited Maharashtra (i.e. 24.7% of total foreign tourists in India) in the year 2012-13[1].Our application helps the tourists to read and understand the Marathi signboards in English which is a global language. Signboard plays an important role in while travelling as the tourists are new to the place. The foreign visitors do not understand the language in which the signboard is written, with the consequent loss of all the important information tourists may get lost on their way, for an example, a tourist might not be able to understand a sign in a foreign country which may depict the warnings or hazards in the specified region.

The Android platform has gained popularity in recent years in terms of share market and number of available applications. Android makes it easier for consumers to get and use new content and applications on their Smart phones. These advancements together with the growing prevalence of smart phones, have made it feasible and in some cases preferable to run OCR software on mobile platforms[2]. The motivation of a real time text translation mobile application is to help tourists navigate in a foreign language environment[3]. Our aim is to develop an Android-platform based text translation application. That is able to recognize the text present on signboard. Images taken with

a mobile phone camera, translates the text from Marathi to English, and display the translation result back on the screen of the mobile phone. This android application translates the text from Marathi to English within a few seconds. When the user starts this application, the application starts capturing an image containing text of interest using the Mobile camera. The specified area of the image is processed on the device in order to optimize it for transfer and input to the OCR database. The processed data is sent to a database where recognition takes place. The results are delivered back to the device and a translated string is obtained. The application once developed will include automatic text detection, an OCR (optical character recognition), text correction, and text translation. Although some of the current version of application is limited to translation from other languages, it can be easily extended into a much wider range of language sets[4]. The existing applications are developed which only work for some specific languages such as Chinese, Japanese, French, Arabic and English, although any application isnt yet developed for translating a Devanagari script we are going to work on this project where Devanagari script too can be recognized and translated[5].

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

In earlier days, food ordering was a completely manual process where a waiter used to note down orders from the customers using pen and paper, take the orders to the kitchen, bring the food and make the bill. Although this system is simple it requires extensive investment in purchase and storage of paper, large manpower and also is prone to human errors and greater time consumption.

In order to overcome these limitations in manual system, some systems were developed later like PDA based systems and multi-touchable restaurant management systems to automate food ordering process.

2.1.1 Need of Indic language translation

According to a recent survey, almost 48, 15,421 foreign tourists visited Maharashtra (i.e. 24.7% of total foreign tourists in India) in the year 2012-13. Marathi is whats known as an Indo-Aryan language. It is spoken in the state of Maharashtra in India and one of the 23 official languages of India. As of 2001 there were around 73 million speakers. Marathi has the 4th largest number of speakers in India[1]. Marathi is spoken mostly in the areas of Maharashtra and parts of neighboring states of, Madhya, Chhattisgarh, Gujarat, Pradesh, Goa, Karnataka, and Andhra Pradesh, union-territories such as Daman and Diu and Dadra and Nagar Haveli. The cities of Baroda, Surat, and Ahmedabad (Gujrat), Belgaum in the Karnataka region, Indore, Hyderabad of the Andhra Pradesh region, Gwalior of the Madhya Pradesh region, and Tanjore which is in Tamil Nadu, each have considerable Marathi-speaking communities. Marathi is also spoken by emigrants that have communities in places as far apart as The USA, Israel, Mauritius, and Canada[1].

2.1.2 The Existing translators

- **Google translate:** This translation application works online and uses OCR concept for translation of text.
- **Word Lens translator:** This application is available for Chinese, Japanese, French, Portuguese and Italian languages. It is a real time application which also works on the principle of OCR.
- **Easy Language translator:** This application requires typed text as an input which is to be translated. After giving the input the application will translate that given input text to the required language.

2.1.3 Translation Process

In this paper author has proposed an Android application that detects the text information within an image taken with a mobile phone camera, extracts it, recognizes it and translates it from Marathi to English. For this translation the concept OCR is used. In this the Devanagari script is divided into 3 zones namely: Middle zone, Upper zone and Lower zone[2]. Upper zone and middle zone are separated by a head line, and the detected characters are replaced with the Devanagari characters resulting to the translation of Marathi Script to English.

2.1.4 Working of OCR

An Android application that detects the text information within an image taken with a mobile phone camera, extracts it, recognizes it and translates it from Marathi to English. For this translation the concept OCR is used and OCR process is divided into 4 phases those are Preprocessing, Segmentation, Feature Extraction and Classification [4]. In Preprocessing thinning and filling techniques are used to obtain characters from the image, Segmentation is used to fragment each character from a word present in the image. Feature extraction is used to classify the characters according their type, and translator translates Marathi script to English transliteration [4].

Chapter 3

OBJECTIVE AND SCOPE

3.1 Objective and Scope

3.1.1 Objectives

The main objective of this project is to overcome the problem of language barrier. The application helps tourists to read and understand the Marathi signboards. This application gives the instant results to the user. There is no need of the internet to use this application because this is an offline application. Once the application is installed then user can use this application long time. The application can be used even in the remote areas where there will be no network. This application is very easy to use and there are no difficulties to use this application and get the desired result. The tourists who are unable to recognize the text, there is no need to type the word by them to get the result. The user has to only hold mobile in front of the Marathi signboard. The application recognizes the text from signboard and translates it to the English and the result gets display at the place of Marathi signboard image.

3.1.2 Scope of project

- **Real time application:** This aims to develop an Android platform based text recognition and translation application that is able to recognize the text present on signboard images taken with mobile phone camera, translate the text from Marathi to English and display translation result by back on the screen of mobile phone. This is a real time application, where the person gets the desired result within a second.
- **User friendly:** A person just need to hold the camera in front of the signboard whose translation is to be done and then the translation will be on the screen. There is no need to learn extra information regarding how to use this application.

Any person is aware of using smart phone and applications on the smart phones can easily use this application too.

- **Offline:** This application works offline which once installed will not require the internet. This application can be used even in the remote areas where there will be no network.

3.1.3 Out of Scope

- **Does not recognize handwritten text:** This application doesnt recognize hand written text. While considering hand written text, we have to consider the fonts of different styles as well as with distinct sizes. This is too difficult process to recognize the handwritten text because the handwriting varies from person to person.
- **Does not give meaning:** This application does not provide the meaning of the word because some words are the names of the cities and villages, as it is specially developed for the tourists to guide them through the route. Also the words are pronounced differently which will be very difficult to recognize the correct one.

Chapter 4

REQUIREMENT ANALYSIS

4.1 Software And Hardware Reqireents:

In requirements there are two types developer side and end user side requirements

4.1.1 Developer side requirements:

4.1.1.1 Software Requirements:

- Operating System- Windows XP and above versions
- Eclipse Indigo
- JDK

Windows Operating System:

Windows XP (codenamed Whistler) is a personal computer operating system produced by Microsoft as part of the Windows NT family of operating systems. The operating system was released to manufacturing on August 24, 2001, and generally released for retail sale on October 25, 2001. Development of XP began in the late 1990s as "Neptune", an operating system built on the Windows NT kernel which was intended specifically for mainstream consumer usean updated version of Windows 2000 was also originally planned for the business market. However, in January 2000, both projects were shelved in favor of a single OS codenamed "Whistler", which would serve as a single OS platform for both consumer and business markets. Windows XP was a major advance from the MS-DOS based versions of Windows in security, stability and efficiency due to its use of Windows NT underpinnings. It introduced a significantly redesigned graphical user interface and was the first version of Windows to use product activation in an effort to reduce software piracy.

Upon its release Windows XP received generally positive reviews, with critics noting increased performance (especially in comparison to Windows ME), a more intuitive user interface, improved hardware support, and its expanded multimedia capabilities Despite some initial concerns over the new licensing model and product activation system, Windows XP eventually proved to be popular and widely used. It is estimated that at least 400 million copies of Windows XP were sold globally within its first five years of availability, and at least one billion copies were sold by April 2014. Windows XP remained popular even after the release of newer versions, particularly due to the poorly received release of its successor Windows Vista. Vista's 2009 successor, Windows 7, only overtook XP in total market share at the end of 2011.

Eclipse:

In computer programming, Eclipse is an integrated development environment (IDE). It is developed by Eclipse Foundation. The stable release is Stable release 4.4.1 (Luna) / 26 September 2014. It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-INS, eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, FORTRAN, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

JDK:

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK) [citation needed]. On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun

contributed the source code to the Open JDK. The stable release of the JDK is Java Standard Edition 8 Update 20 (1.8.0.20) / August 19, 2014. The developers of JDK are Oracle Corporation.

4.1.2 End user side:

Operating System- Android 4.0 or above

Android Operating System:

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google OHA. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, and other electronics [6].

Android is the most popular mobile OS. As of 2013, Android devices sell more than Windows, iOS, and Mac OS devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013 the Google Play store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in AprilMay 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

The Androids Initial release is in September 23, 2008 and the latest release is in the 4.4.4 KitKat / June 19, 2014. The first release of android is Alpha (1.0). We are using the Android 4.0 "Ice Cream Sandwich" in our project. The brief description of android 4.0 is here.

Android 4.0 "Ice Cream Sandwich": Android 4.0 "Ice Cream Sandwich" is a version of the Android mobile operating system developed by Google. Unveiled on October 19, 2011, Android 4.0 builds upon the significant changes made by the tablet-only release

Android 3.0 "Honeycomb", in an effort to create a unified platform for both smart phones and tablets, whilst simplifying and modernizing the overall Android experience around a new set of human interface guidelines. As part of these efforts, Android 4.0 introduced a new visual appearance codenamed "Holo", which is built around a cleaner, minimalistic design, and a new default typeface named Robot[7].

4.1.2.1 Minimum Hardware requirement:

- Smart Mobile Phone
- RAM 256 MB
- Camera 2 mega pixel

Chapter 5

SYSTEM DESIGN

5.1 An Overview of The UML

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

THE UML LANGUAGE

A language provides a vocabulary and the rules for combing words in that vocabulary for the purpose of the communication. A modeling language is a language whose vocabulary and rules focus on conceptual and physical representation of a system. A modeling language such as the UML is thus a standard language for software blueprints. In this context, specifying means building models that are precise, unambiguous, and complete. In particular, the UML addresses the specification of all the important analysis, design and implementation decision that must be made in developing and deploying a software intensive system. The UML is not a visual programming language, but its model can be directly connected to a verity of programming languages. This means that its possible to map from a model in the UML to a programming language such as java, cpp, or visual basic or even to tables in a relational database. Things that are best expressed graphically are done so graphically in the UML, whereas things that best expressed textually are done so in the programming language. A healthy software organization produces all sorts of artifacts in addition to raw executable code. These artifacts include requirements, architecture, design, source code, project plans, tests, prototypes,

releases. The UML addresses the documentation of a systems architectures and all of its details. The UML also provides for expressing requirements and for tests. Finally, The UML provides a language for modeling the activities of project planning and release management.

5.2 Goals of UML

The primary goals in the design of the UML were:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.Provide a formal basis for understanding the modeling language
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

5.3 A Conceptual Model of The UML

To understand the UML, you need to form a conceptual model of the language, and this requires learning three major elements: the UMLs basic building blocks, the rules that dictate how those building blocks may be put together, and some mechanisms that apply throughout the UML. Once you have grasped these ideas, you will be able to read UML models and create some basic ones. As you gain more experience in applying the UML, you can build on this conceptual model, using more advanced features of the language.

5.3.1 Building Blocks of The UML

The vocabulary of the UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

These are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams groups interesting collections of things.

5.4 Diagrams in The UML

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). You draw diagrams to visualizing a system from different perspectives, so a diagram is a projection into a system. For all but the most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams. In theory, a diagram may contain any combination of things and relationships. The views that comprise the architecture of software intensive system. For this reason, the UML includes following diagrams:

- Use case diagram
- Class diagram
- Sequence diagram
- Deployment diagram

5.4.1 Use Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is a just special kind of diagram and shares the same common properties as do all other diagram-a name and graphical contents.

5.4.1.1 Contents

Use case diagrams commonly contain

- **Use Case**

  Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is rendered as an ellipse with solid lines usually including its name.

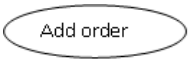

FIGURE 5.1: Use Cases

- **Actors**

  An actor represents a role that an outsider takes on when interacting with the business system. For instance, an actor can be a customer, a business partner, a supplier, or another business system and every actor has a name.

- **Dependency, generalization, and association relationships.**

  A dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing.

FIGURE 5.2: dependencies

A *generalization* is a relationship in which objects of specialized elements (the child) are substitutable for objects of the generalized element.

An *association* is a structural relationship that describes a set of links, a link being connection among objects

Like all other diagrams, use case diagram may contain notes and constraints.

5.4.1.2 Common uses

Use case diagram typically contain in one of two ways.

- To model the context of the system

  Here system involves drawing line around the whole system and actors outside of the system and interact with it.

- To model the requirement of a system

  Here specifies what the system should do, independent of how that system should do.

5.4.1.3 Use-Case Scenarios

Use Case	Description
Translate	1. User holds the phone in-front of the text. 2. SmartEye captures the text/image. 3. SmartEye invokes OCR. 4. OCR detects the text. 5. OCR returns the detected text to SmartEye. 6. SmartEye will invoke translator. 7. SmartEye will pass detected text to the translator. 8. Translator will map the detected text against the database. 9. Translator will return the translated text to SmartEye. 10. SmartEye invokes Displayer. 11. SmartEye passes translated text to Displayer. 12. Displayer changes the font size, color of translated text, as per detected text. 13. Displayer displays the text at detected text location.
	Alternate Flow: 1. User holds the phone in-front of signboard/text. 2. SmartEye captures the text/image. 3. SmartEye invokes OCR. 4. OCR detects the text. 5. If OCR doesnt detect the text, SmartEye again capturers the text.

TABLE 5.1: Use Case Scenario Table

5.4.1.4 Use-Case Diagram

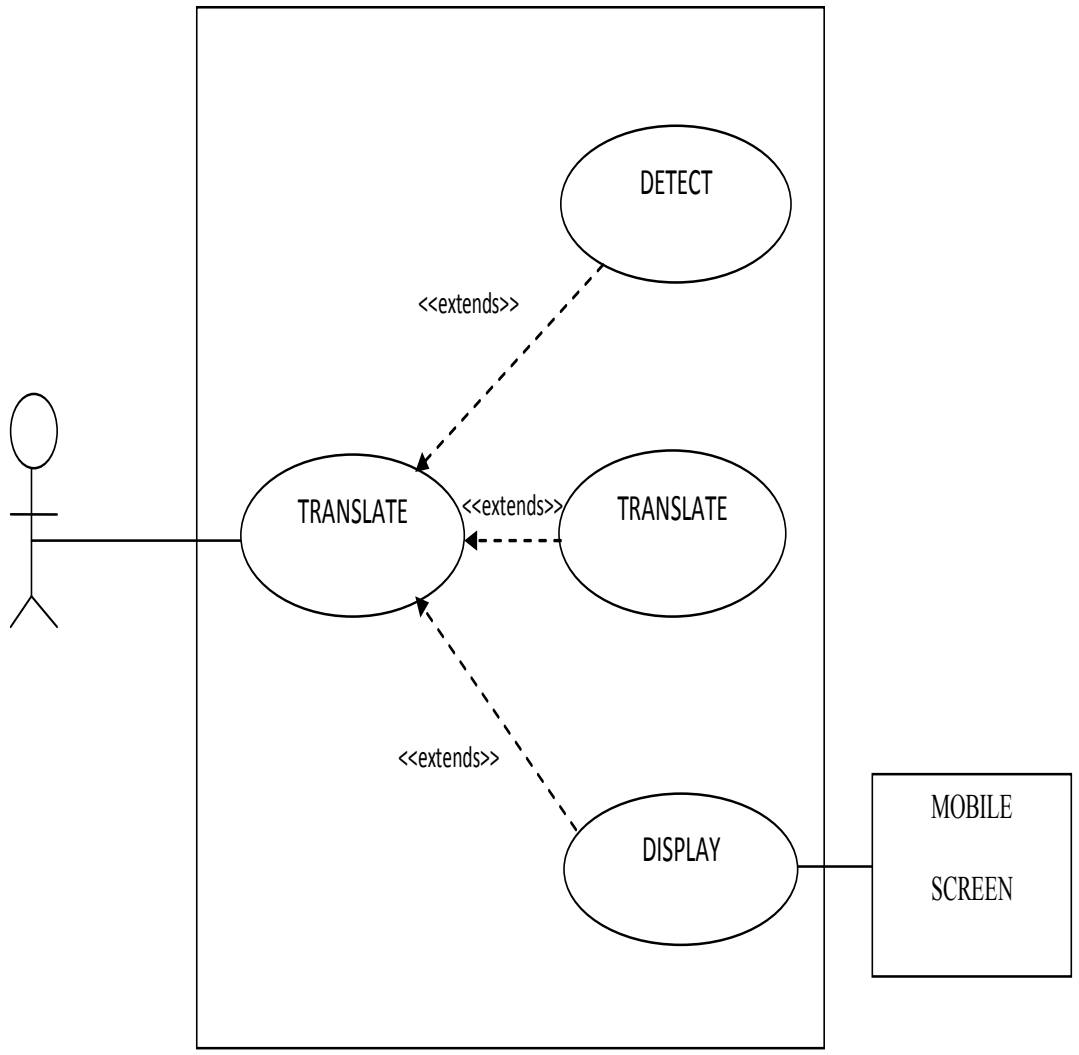

FIGURE 5.3: Use-Case Diagram For SmartEye

5.4.2 Sequence Diagram

5.4.2.1 Contents

Sequence diagram commonly contains

- Objects
- Links
- Messages

5.4.2.2 Definition And Overview

A *sequence* diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. You use sequence diagrams to illustrate the dynamic view of a system. An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases.

Sequence diagram have two features that distinguish them from collaboration diagrams.

- First, there is the object lifeline. An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. So these objects are at the top of the diagram. With their lifelines drawn from the top of the diagram to the bottom
- Second, there is the focus of control. The focus of control is a tall, thin rectangle that shows the period of time during which an object is performing an action, either directly or through a subordinating procedure. The top of the rectangle is aligned with the start of the action; the bottom is aligned with its completion and also it can be marked by replay message.

5.4.2.3 Sequence Diagrams

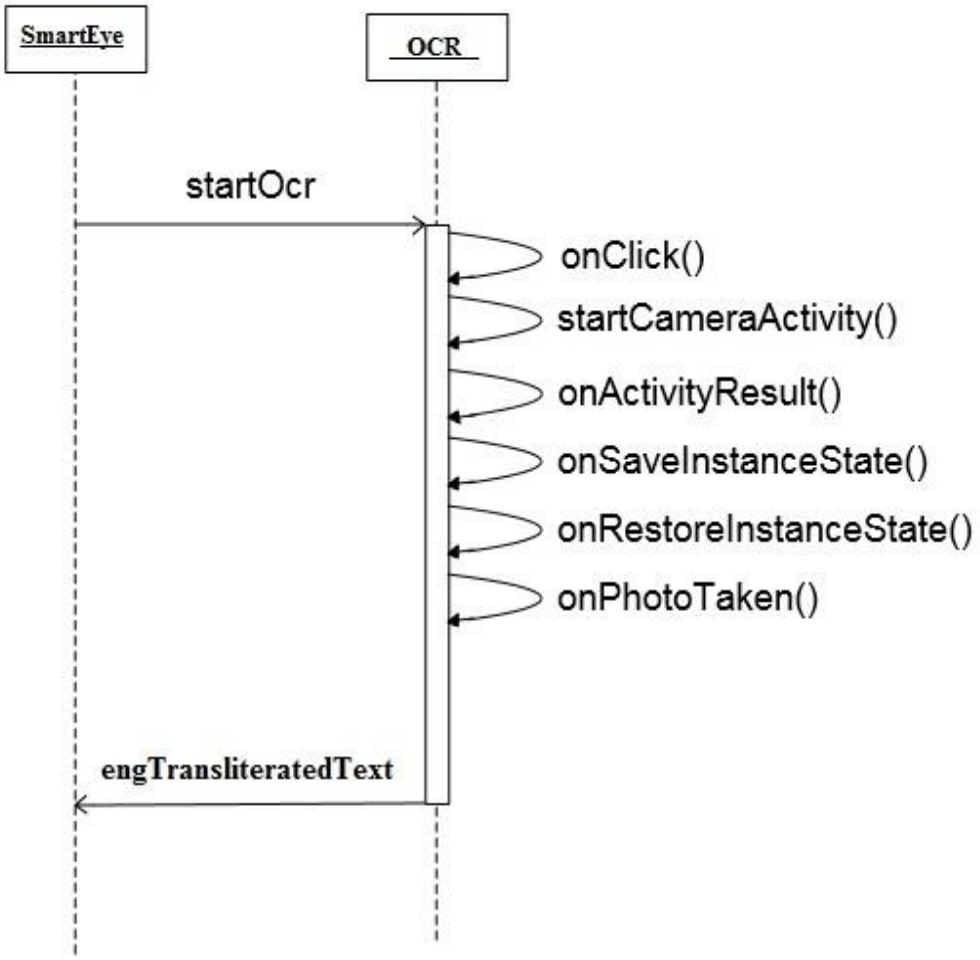

FIGURE 5.4: Sequence Diagram For SmartEye

5.4.3 Class Diagram

5.4.3.1 Contents

Class diagram commonly contain the following things:

- Classes
- Interfaces

- Collaborations
- Dependency, generalization, and association relationships.

5.4.3.2 Definition and Common Uses

A class diagram is a diagram that shows a set of classes, interfaces and their relationships. Graphically, a class diagram is a collection of vertices and arcs. A class diagram will shares the same common properties as do all other diagrams. A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved. In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

- Class: A definition of objects that share given structural or behavioral characteristics.
- Attribute: A typed value attached to each instance of a classifier.
- Operation: A method or function that can be performed by instances of a classifier

5.4.3.3 Class Diagram:

5.4.4 Deployment Diagram

subsectionDefinition A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagram address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components .

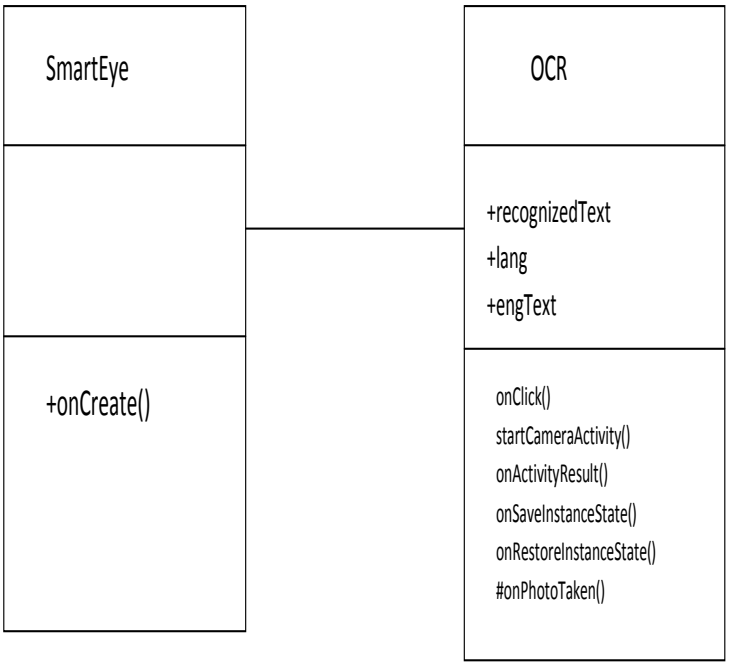

FIGURE 5.5: Class Diagram For SmartEye

5.4.4.1 Nodes and Components

The UML provides a graphical representation of node. This canonical notation permits you to visualize a node apart from any specific hardware. Using stereotype this notation to represents specific kinds of processors and devices.

A *node* is a physical element that exists at run time and represents a computational resource, generally having at least some memory, and often processing capability. Graphically, a node is rendered as a cube. Every node must have a name that distinguishes it from other nodes. A name is a textual string. Components are things that participate in the execution of a system; nodes are things that execute components. Components represent the physical packaging of otherwise logical elements; nodes represent the physical development of components and components that things are executed by nodes. The UML can often use stereotypes to specify new kinds of nodes that you can use to represent specific kinds of processors and devices. A *Processor* is a node that has processing capability, meaning that it can be executed by component. A*device* is a node that has no processing capability and general, represents something that interfaces to real world.

5.4.4.2 Deployment Diagram

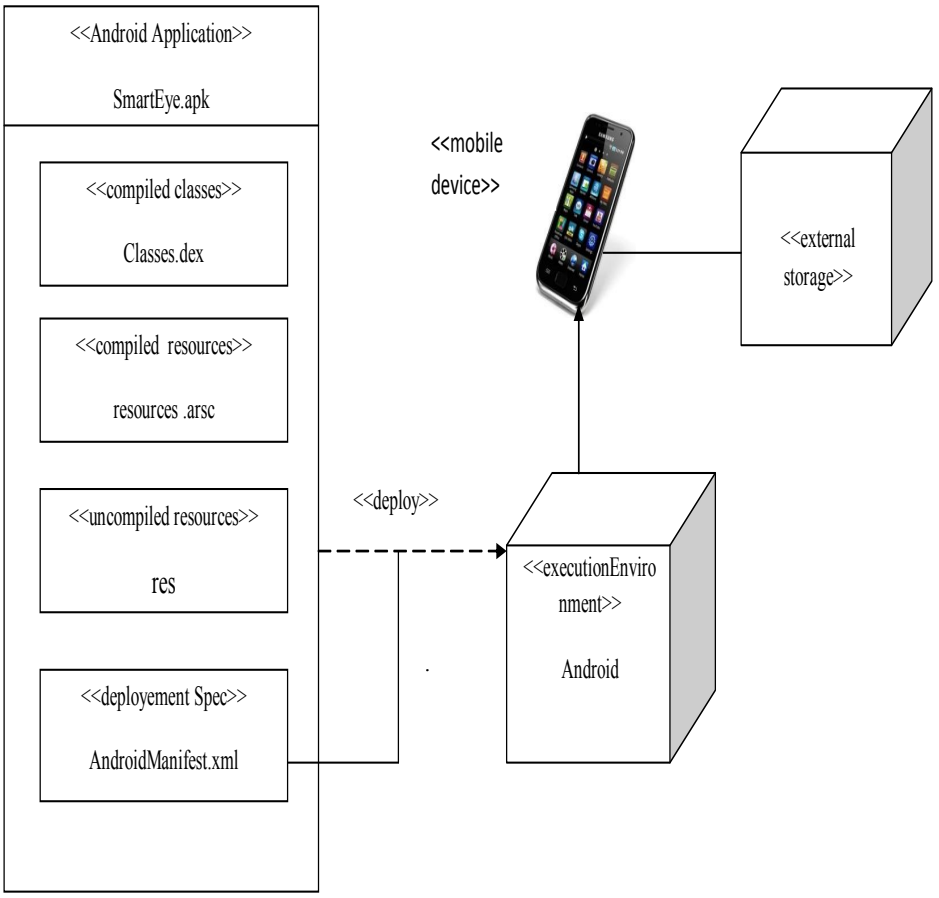

FIGURE 5.6: Deployment Diagram For SmartEye

Chapter 6

CODING

6.1 Introduction of tools and Installation

6.1.1 Android

There's no other software quite like Android. Google engineered Android, and Google own apps run better on it. And with millions of apps, games, songs, and videos on Google Play, Android is great for fun, and for getting things done.

Android devices come in all kinds of sizes, with all sorts of features, and in all sorts of prices. Each version of Android is named after a dessert, and the most recent version of Android is lolipop. With Android, youre in control of your mobile experience.

The world is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera , Music player, Tablet PC, T.V. , Web browser etc. . And with the new technologies, new software and operating systems are required.

- What is android

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

- ADT Bundle

The Android SDK is a software development kit which provides API libraries and necessary developer tools necessary for building Android applications. Android SDK is officially provided by android developers.

steps for the installation and set-up of Android development environment:

1. Download Eclipse
2. Download JDK and install it, set the environment path.
3. Download ADT plugin inside Eclipse.
4. Set the Preference with Android-SDK path.
5. Download the latest platform-tools and everything.

The ADT Bundle includes everything you need to begin developing apps:

1. Eclipse + ADT plugin
2. Android SDK Tools
3. Android Platform-tools
4. The latest Android platform
5. The latest Android system image for the emulator

Yes there are also possible ways if you want to use existing version of Eclipse or any other IDE.

- **Setting Up the ADT Bundle:**

As you have downloaded ADT bundle, follow below steps to setup it:

1. Unpack the ZIP file named "adt bundle osplatform.zip " and save it to an appropriate location such as a "Development" directory in your home directory.
2. Open the adt bundle osplatform goto eclipse and next directory and launch eclipse.

6.2 Coding

```
                /* Text Reorganization */
        protected void onPhotoTaken() {
                _taken = true;

BitmapFactory.Options options = new BitmapFactory.Options();
                options.inSampleSize = 4;

Bitmap bitmap = BitmapFactory.decodeFile(_path, options);

try {
    ExifInterface exif = new ExifInterface(_path);
    int exifOrientation = exif.getAttributeInt(
   ExifInterface.TAG_ORIENTATION,ExifInterface.ORIENTATION_NORMAL);

     Log.v(TAG, "Orient: " + exifOrientation);

     int rotate = 0;

switch (exifOrientation) {
    case ExifInterface.ORIENTATION_ROTATE_90:
        rotate = 90;
        break;
   case ExifInterface.ORIENTATION_ROTATE_180:
        rotate = 180;
        break;
  case ExifInterface.ORIENTATION_ROTATE_270:
        rotate = 270;
        break;
        }
s Log.v(TAG, "Rotation: " + rotate);

        if (rotate != 0) {

        // Getting width & height of the given image.
        int w = bitmap.getWidth();
        int h = bitmap.getHeight();

        // Setting pre rotate
        Matrix mtx = new Matrix();
```

```
        mtx.preRotate(rotate);
        // Rotating Bitmap
        bitmap = Bitmap.createBitmap(bitmap, 0, 0, w, h,
mtx, false);
        }

        // Convert to ARGB_8888, required by tess
        bitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);

        } catch (IOException e) {
        Log.e(TAG, "Couldn't correct orientation: " +
e.toString());
        }

        // _image.setImageBitmap( bitmap );

        Log.v(TAG, "Before baseApi");
        TessBaseAPI baseApi = new TessBaseAPI();
        baseApi.setDebug(true);
        baseApi.init(DATA_PATH, lang);
        baseApi.setImage(bitmap);
        String recognizedText = baseApi.getUTF8Text();
          baseApi.end();

              /*Mapping of letters*/
                recognizedText = recognizedText.trim();
         //byte[] mar=
recognizedText.getBytes(Charset.forName("UTF-8"));
                //char m = '\u092E';
                //char d = '\u0926';
                //char n = '\u0928';
        /*
        String myText=new String("\u092E" + "\u0926" +
"\u0928");

                if(recognizedText.equals(myText))
                {
                        recognizedText="mdn";
                }
                else
                {
```

```
                recognizedText="no";

        }
        */
        String engText=new String();

        for(int i=0;i<recognizedText.length();i++)
        {
                switch(recognizedText.charAt(i))
                {
                case '\u0902':
                        engText=engText+"n";
                        break;
                case '\u0904':
                        engText=engText+"a";
                        break;
                case '\u0905':
                        engText=engText+"a";
                        break;
                case '\u0906':
                        engText=engText+"aa";
                        break;
                case '\u0907':
                        engText=engText+"i";
                        break;
                case '\u0908':
                        engText=engText+"ii";
                        break;
                case '\u0909':
                        engText=engText+"u";
                        break;
```

6.3 Issues

1) When we cloned tesser-act ocr engine from "https://code.google.com/p/tesseract-ocr/downloads/list". Tesseract Tools for Android is a set of Android APIs and build files for the Tesseract OCR and Leptonica image processing libraries. The ocr engine ran sccessfully with the english trained data, but when we tried it for marathi trained data the message "Unfortunately, application stopped " popped up. For resloving the issue, we checked out with the Japanese, Kannada trained data file. We also tried different

versions of Hindi trained data files. And finally this issue was resolved by adding Konkani trained data file.

2) Faced problem with SDK manager as path variable was not set properly.We fixed it by changing "PATH" in enviroment variables.

3) Faced problem while mapping English letters with Marathi, So we used Unicodes of Devanagari script.

6.4 Unicode

Unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems. Developed in conjunction with the Universal Character Set standard and published as The Unicode Standard, the latest version of Unicode contains a repertoire of more than 110,000 characters covering 100 scripts and multiple symbol sets. The standard consists of a set of code charts for visual reference, an encoding method and set of standard character encodings, a set of reference Data computer files, and a number of related items, such as character properties, rules for normalization, decomposition, collation, rendering, and bidirectional display order (for the correct display of text containing both right-to-left scripts, such as Arabic and Hebrew, and left-to-right scripts). As of June 2014, the most recent version is Unicode 7.0. The standard is maintained by the Unicode Consortium.

Unicode's success at unifying character sets has led to its widespread and predominant use in the internationalization and localization of computer software. The standard has been implemented in many recent technologies, including modern operating systems, XML, the Java programming language, and the Microsoft .NET Framework.

Unicode can be implemented by different character encodings. The most commonly used encodings are UTF-8, UTF-16 and the now-obsolete UCS-2. UTF-8 uses one byte for any ASCII character, all of which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters. UCS-2 uses a 16-bit code unit (two 8-bit bytes) for each character but cannot encode every character in the current Unicode standard. UTF-16 extends UCS-2, using one 16-bit unit for the characters that were representable in UCS-2 and two 16-bit units (4 8 bit) to handle each of the additional characters.

FIGURE 6.1: character encodings in UTF-8

Chapter 7

TESTING

7.1 What is Software Testing

Software testing is the process of analyzing or operating software for the purpose of finding bugs. Testing can be described as a process used for reveling defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attribute.The fundamental objective of testing is to find defects ,as early as possible and get them fixed.

Software Testing Process

- Test Planning high level plans which list test objectives, test approach, measurement criteria along with test schedule and resources.
- Test Design create test cases, identify test cases for automation(if applicable),prioritize test cases and finalize test iterations.
- Test Implementation Create test scripts using automated testing tools.
- Test Execution Execute the test cases on the test environment and test reports.
- Test analysis Use test and project metrics to calculate key indicators. The data usually will be obtained from your defect tracking system.
- Postmortem reviews Discuss lessons learnt and identify strategies which will prevent such problems in future.

7.2 Test methods

7.2.1 Black box testing

It is also called as functional testing, it is the process of giving the input to the system and checking the output of the system.Without bothering about the system that how the system generates the output. It is also called as Behavior testing.

- Approach to testing where the program is considered as a Black Box.
- Testing based solely on analysis of requirements user specification, user documentation etc.
- The test cases are based on the specifications.
- Black box testing techniques apply to all levels of testing.
- Test planning and design can begin early in the software process.
- Tests are done from a users point of view.

7.2.2 White Box Testing

White box testing or structural testing considers facets like programming style, control method, source language, database design. A test for remote monitoring routine can be an example of structural test. This type of testing helps to uncover defects at structural level. The tests go below the top or functional layer to uncover the defects.

- Testing that takes into account internal structure and flow of a system or component.
- The testing is based on code structure or the algorithm.
- White box testing assumes that the procedural design and code is known to the tester.
- Obliviously test design can be done only after coding is complete.
- White box tests are inherently finite.

7.3 Test cases and test data

- Test data are inputs that have been devised to test the system.

- Test cases are inputs and output specification plus a statement of the function under test.
- Test data can be generated automatically or real.

TABLE 7.1: Test Cases For SmartEye

Sr.No.	TC_ID	Objectives	Prerequisites	Steps to be followed	Expected Result	Remark
1	Start the application.	Application should be installed in the phone.	1.Go to menus. 2.Click on the application icon.	The Application should start.	Application gets started successfully.	Pass
2	Click on the OCR button.	Application should be started.	Click on the OCR button.	Camera should open.	Camera gets started.	Pass
3	Capture the image.	Camera should be started.	Take a high resolution image.	High resolution image should be captured.	High resolution image will be captured.	Pass
4	Save the image.	Image should be captured.	Click on the save button.	Transliterated text should be displayed.	Transliterated text will be displayed.	Pass
5	Recapture image if image is not clear.	Camera should be started.	Take a clear image.	High resolution image should be captured.	High resolution image will be captured.	Pass
6	Checking with Shivaji01 font.	Image should be captured.	Click on the save button.	Transliterated text should be displayed.	Transliterated text will be displayed.	Pass

Continued on next page

Table 7.1 — continued from previous page

Sr.No.	TC_ID	Objectives	Prerequisites	Steps to be followed	Expected Result	Remark
7	Checking with Shivaji02 font.	Image should be captured.	Click on the save button.	Transliterated text should be displayed.	Transliterated text will be displayed.	Fail
8	Checking with Shivaji03 font.	Image should be captured.	Click on the save button.	Transliterated text should be displayed.	Transliterated text will be displayed.	Fail
9	Checking with Handwritten text.	Image should be captured.	Click on the save button.	Transliterated text should be displayed.	Transliterated text will be displayed.	Fail

Chapter 8

DEPLOYMENT

8.1 Snapshot

8.1.1 Start Application:

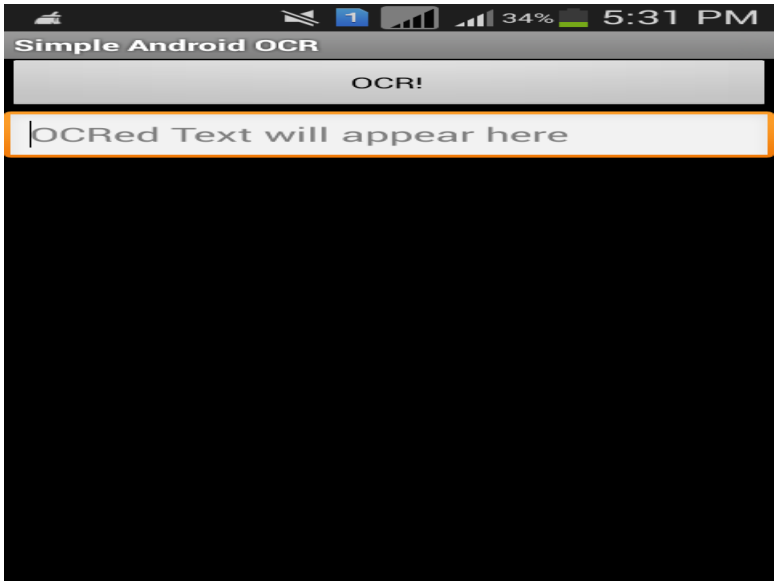

FIGURE 8.1: Start window

When we click on the SmartEye icon, application gets started and we get the above display.

8.1.2 Image Capture:

On clicking the OCR button, Camera gets started. Capture the image of high resolution for better OCR recognition.

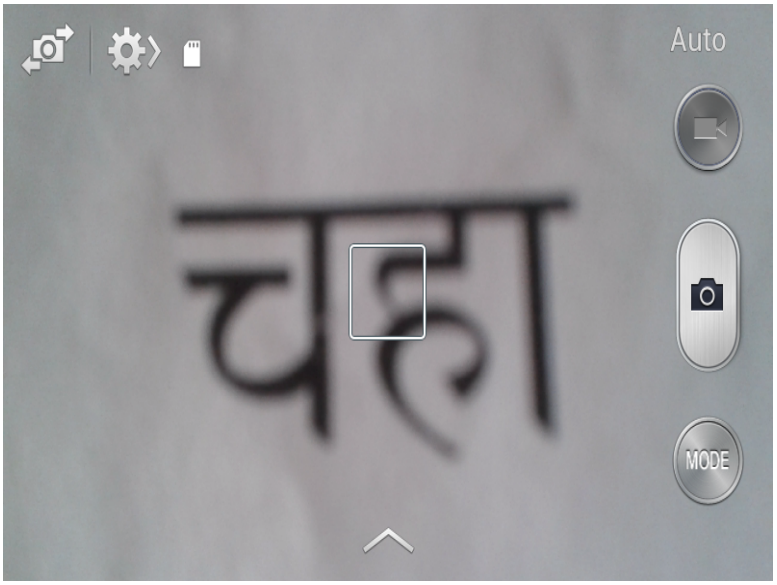

FIGURE 8.2: capturing image

it

8.1.3 Save Image:

On capturing the image, click on the save button so as to get the desired output.

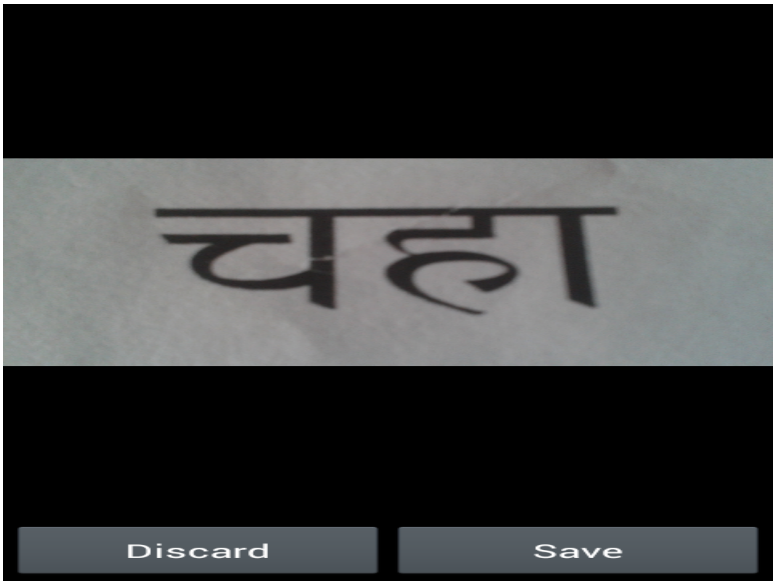

FIGURE 8.3: save image

8.1.4 Result:

On clicking the save button, we get the required output.

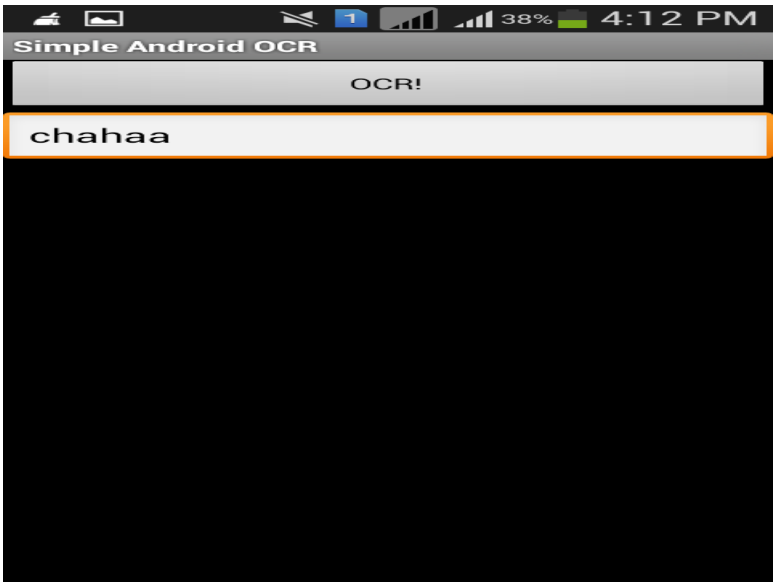

FIGURE 8.4: Output Window.

.

Chapter 9

CONCLUSION

9.1 Conclusion

This project named SmartEye: Real time text recognition and translation application for Indic Languages is a mobile camera based android application. This application recognizes the text from Marathi signboard and translates it to the particular output language and display the result instantly. The main feature of this application is it works offline. This application saves the time required to typing and recognizing the signboards.

Now a days use of mobile is broadly increased and almost each and every person uses mobile phone, in which lots of applications run. Our project is also compatible with mobile phones which make it portable thats a desired property of any software. Hence we can conclude this project is real time application and very useful for tourist navigation and language understanding. We proposed a system to translate signboard images taken with a mobile phone camera from Marathi to English. Since the computational resources of these devices are limited, we had to use fast, simple and accurate possible algorithms to work in the most common situations. Our system shows some characteristics that make it interesting and deserve further research.

Bibliography

[1] Google. Maharashtra tourism, the official website of maharashtra tourism development corporation, govt. of india., 2014. URL http://www.maharastratourism.gov.in/.

[2] Gunvantsinh Gohil, Rekha Teraiya, and Mahesh Goyani. Chain code and holistic features based ocr system for printed devanagari script using ann and svm. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 3(1), 2012.

[3] Google., 2014. URL www.cdac.in/.

[4] AA Tayade, RV Mante, and PN Chatur. Text recognition and translation application for smartphone.

[5] Google. Uml standard diagrams, learning uml in easy steps, 2014. URL http://www.tutorialspoint.com/uml/uml_diagrams.html./.

[6] Jonathan Simon. *Head First Android Development*. OReilly Media, 2012.

[7] Masumi Nakamura Zigurd Mednieks. *Programming Android*. OReilly Media, 2012.