

Universal Interpreter

Jerry Allan Akshay, Shiva Pundir, Rahul R and Sai Kumar

10/03/2020

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We express our sincere gratitude to the University Visvesvaraya College of Engineering, Bangalore, for having given us the opportunity to carry out this project. We would also like to thank our Principal, Dr. H. N. Ramesh, for providing us all the facilities to work on this project.

We wish to place on record our grateful thanks to Dr. Dilip Kumar, Head of the Department, Department of Computer Science and Engineering, for providing us a wonderful opportunity to learn and the needful resources for this project.

We consider it a privilege and an honour to express our sincere gratitude to our guide, Mrs. P. Deepa Shenoy, for her valuable guidance throughout the tenure of this project work, and whose support and encouragement made this work possible.

We also thank our parents and our friends for their help, encouragement and support. Last but not the least; we thank God Almighty, without whose blessings this wouldn't have been possible.

Thanking You

Jerry Allan Akshay

Shiva Pundir

Rahul R.

Sai Kumar

Abstract

The project 'Universal Interpreter' was built as an application to help the differently-abled people, i.e. deaf, dumb, blind or combination of any, by using Image Recognition and AI/ML.

The present world we live in, a world dominated by visual and audio peripherals, can prove to be a tough place to live in for differently abled people. Hence, this project was inspired by the need to use the very same dominating technologies to help the differently abled overcome their challenges.

With Universal Interpreter, we make use of Morse Codes and the American Sign Language - two of the standardised forms of communication all over the world - to ensure that any differently-abled person can use the application for its intended purposes.

Table of Contents

1	Introduction	1
1.1	Problem Definition	1
1.2	Objectives	1
1.3	Standardised Forms of Communication	2
1.3.1	Morse Codes	2
1.3.2	American Sign Language	2
1.4	Tools	2
1.4.1	Front-end Tools	3
1.4.2	Back-end Tools	5
2	System Requirements	9
2.1	Software Requirements	9
2.2	Hardware Requirements	9
3	System Design	10
3.1	Android Application	10
3.1.1	Application Structure	10
3.1.2	Front-end Features	10
3.1.3	Back-end Features	10
3.1.4	Integration of Front-end and Back-end	10
3.2	Inception v3 Neural Network	10
3.3	Application Programming Interfaces	10
3.3.1	TensorFlow Lite API	10
3.3.2	Firebase API	10
3.3.3	Google Text-to-Speech API	10
3.3.4	Google Speech-to-Text API	10
4	Project Implementation	11
4.1	Project Flow	11
4.1.1	Phase I	11
4.1.2	Phase II	11
4.1.3	Phase III	11

4.2	Android Application Implementation	11
4.2.1	Design Ideologies	11
4.2.2	Application Phases	11
4.2.3	Database Structure	11
4.3	Inception v3 Neural Network	11
4.3.1	Data Collection	11
4.3.2	Model Training	11
4.3.3	Model Predictions	11
5	Project Snapshots	12
5.1	Landing Page	12
5.2	Settings/Preferences Page	12
5.3	Input Methods	12
6	Conclusion	13

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Definition

The world we live in is a world dominated by technology made for people with no impairments. Smartphones and Computers form a major portion of this technology. In the current age and time, most of the communication between people takes place through the use of smartphones and computers via SMS(s), Emails WhatsApp messages and so forth.

As such, physically impaired people don't get to reap the benefits of smartphones, computers and other technologies catered to the non-impaired as there are no appropriate interfaces designed for use by them on these devices.

1.2 Objectives

The main objective of this project is to set up interfacing mechanisms for the physically-impaired - specifically the deaf, dumb and blind - and design an application for the purpose of communication between the users of the aforementioned application. The objectives can be postulated as follows:

- To set up interfacing mechanisms using Morse Codes, Text, Speech and the American Sign Language.
- To design an Android Application for the purposes of communication using the aforementioned interfaces.
- To structure and use a no-SQL database in the appropriate format for the purpose of use of the Android Application.

- To train and use a Neural Network Model for the purposes of recognizing the Hand Gestures in the form of the American Sign Language through the use of a Camera.

1.3 Standardised Forms of Communication

In general, communication is simply the act of transferring information from one place, person or group to another. Every communication involves (at least) one sender, a message and a recipient. This may sound simple, but communication is actually a very complex subject. The transmission of the message from sender to recipient can be affected by a huge range of things. These include our emotions, the cultural situation, the medium used to communicate, and even our location. The complexity is why good communication skills are considered so desirable by employers around the world: accurate, effective and unambiguous communication is actually extremely hard.

With regards to communication, there are various internationally accepted and standardised forms of communication. These standardised forms imply the universal understanding of the information being conveyed with the help of the standardised medium.

Since our project is aimed to help the physically impaired such as those who are deaf, dumb, blind, or a combination of these, we make use of two particular standardised forms of communication that are proven to be used effectively for communication by the physically impaired, namely Morse Codes and the American Sign Language (ASL). These two forms of communication are discussed in detail below.

1.3.1 Morse Codes

1.3.2 American Sign Language

1.4 Tools

There are various tools that have been used to realize the project Universal Interpreter. The combinatorial usage of these tools has helped to fabricate this project. As such, there are two categories of tools - The Front-end Tools and the Back-end Tools.

Front-end and Back-end are terms used by programmers and computer professionals to describe the layers that make up hardware, a computer program or a website which are delineated based on how accessible they are to a user. These two categories of tools are further discussed below.

1.4.1 Front-end Tools

The layer above the back end is the front end and it includes all software or hardware that is part of a user interface. Human or digital users interact directly with various aspects of the front end of a program, including user-entered data, buttons, programs, websites and other features. Most of these features are designed by user experience (UX) professionals to be accessible, pleasant and easy to use. The Front-end tools used for this project are mentioned below.

Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

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

- Apply Changes to push code and resource changes to your running app without restarting your app
- 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

Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

The use of Python can span through countless yarns of thread. Few of the many capabilities of Python are (but are not limited to):

- It can be used on a server to create web applications.
- It can be used alongside software to create workflows.
- It can connect to database systems. It can also read and modify files.
- It can be used to handle big data and perform complex mathematics.
- It can be used for rapid prototyping, or for production-ready software development.

With the use of certain frameworks and libraries, Python can very well be used to develop certain front-end interfaces.

1.4.2 Back-end Tools

The back end refers to parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user. Most data and operating syntax are stored and accessed in the back end of a computer system. Typically the code is comprised of one or more programming languages. The back end is also called the data access layer of software or hardware and includes any functionality that needs to be accessed and navigated to by digital means.

A back-end application or program supports front-end user services, and interfaces with any required resources. The back-end application may interact directly with the front end or it may be called from an intermediate program that mediates front-end and back-end activities. Few of the back-end tools used in this project are discussed below.

Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of

October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

Firebase evolved from Envolv, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolv provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolv to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

In May 2012, a month after the beta launch, Firebase raised \$1.1 million in seed funding from venture capitalists Flybridge Capital Partners, Greylock Partners, Founder Collective, and New Enterprise Associates. In June 2013, the company further raised \$5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners.

In 2014, Firebase launched two products. Firebase Hosting and Firebase Authentication. This positioned the company as a mobile backend as a service.

In October 2014, Firebase was acquired by Google. A year later, in October 2015, Google acquired Divshot, an HTML5 web-hosting platform, to merge it with the Firebase team.

In May 2016, at Google I/O, the company's annual developer conference, Firebase introduced Firebase Analytics and announced that it was expanding its services to become a unified backend-as-a-service (BaaS) platform for mobile developers. Firebase now integrates with various other Google services, including Google Cloud Platform, AdMob, and Google Ads to offer broader products and scale for developers. Google Cloud Messaging, the Google service to send push notifications to Android devices, was superseded by a Firebase product, Firebase Cloud Messaging, which

added the functionality to deliver push notifications to both iOS and web devices. In January 2017, Google acquired Fabric and Crashlytics from Twitter to add those services to Firebase.

In October 2017, Firebase has launched Cloud Firestore, a real-time document database as the successor product to the original Firebase Realtime Database.

Neural Network Model

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections.

All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

For this particular project, we are using the Inception v3 Neural Network Model. Inception v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for GoogLeNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. Just as ImageNet can be thought of as a database of classified visual objects, Inception helps classification of objects in the world of computer vision. One such use is in life sciences, where it aids in the research of Leukemia. It was "codenamed 'Inception' after the film of the same name".

In our particular use-case scenario, we use to Inception v3 Model to recognize the Hand Symbols for the different alphabets present in the Amer-

ican Sign Language (ASL).

Python

As discussed in the previous subsection on front-end tools, Python can very well be used for backend purposes too. With the use of the right APIs (Application Programming Interfaces), Python can be used for a wide variety of tasks such as server side programming, machine learning predictions, and many more.

Java

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets)

In our particular project, we use Java for the design of the android application as it is one of the accepted languages to code in Android Studio.

Chapter 2

System Requirements

2.1 Software Requirements

2.2 Hardware Requirements

Chapter 3

System Design

3.1 Android Application

3.1.1 Application Structure

3.1.2 Front-end Features

3.1.3 Back-end Features

3.1.4 Integration of Front-end and Back-end

3.2 Inception v3 Neural Network

3.3 Application Programming Interfaces

3.3.1 TensorFlow Lite API

3.3.2 Firebase API

3.3.3 Google Text-to-Speech API

3.3.4 Google Speech-to-Text API

Chapter 4

Project Implementation

4.1 Project Flow

4.1.1 Phase I

4.1.2 Phase II

4.1.3 Phase III

4.2 Android Application Implementation

4.2.1 Design Ideologies

4.2.2 Application Phases

4.2.3 Database Structure

4.3 Inception v3 Neural Network

4.3.1 Data Collection

4.3.2 Model Training

4.3.3 Model Predictions

Chapter 5

Project Snapshots

5.1 Landing Page

5.2 Settings/Preferences Page

5.3 Input Methods

Chapter 6

Conclusion

References