Universal Interpreter

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

10/03/2020

1 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

2 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 Acknowledgement

2 Abstract

3 Introduction **1**
3.1 Problem Definition . . . 1
3.2 Objectives . . . 1
3.3 Standardised Forms of Communication . . . 1
3.3.1 Morse Codes . . . 1
3.3.2 American Sign Language . . . 1
3.4 Tools . . . 1
3.4.1 Frontend Tools . . . 1
3.4.2 Backend Tools . . . 1

4 System Requirements **2**
4.1 Software Requirements . . . 2
4.2 Hardware Requirements . . . 2

5 System Design **3**
5.1 Android Application . . . 3
5.1.1 Application Structure . . . 3
5.1.2 Frontend Features . . . 3
5.1.3 Backend Features . . . 3
5.1.4 Integration of Frontend and Backend . . . 3
5.2 Inception v3 Neural Network . . . 3
5.3 Application Programming Interfaces . . . 3
5.3.1 TensorFlow Lite API . . . 3
5.3.2 Firebase API . . . 3
5.3.3 Google Text-to-Speech API . . . 3
5.3.4 Google Speech-to-Text API . . . 3

6 Project Implementation **4**
6.1 Project Flow . . . 4
6.1.1 Phase I . . . 4
6.1.2 Phase II . . . 4
6.1.3 Phase III . . . 4
6.2 Android Application Implementation . . . 4
6.2.1 Design Ideologies . . . 4
6.2.2 Application Phases . . . 4
6.2.3 Database Structure . . . 4
6.3 Inception v3 Neural Network . . . 4
6.3.1 Data Collection . . . 4
6.3.2 Model Training . . . 4
6.3.3 Model Predictions . . . 4

7 Project Snapshots **5**
7.1 Landing Page 5
7.2 Settings/Preferences Page 5
7.3 Input Methods 5

8 Conclusion **6**

9 References **7**

List of Figures

List of Tables

3 Introduction

3.1 Problem Definition

3.2 Objectives

3.3 Standardised Forms of Communication

3.3.1 Morse Codes

3.3.2 American Sign Language

3.4 Tools

3.4.1 Frontend Tools

3.4.2 Backend Tools

4 System Requirements

4.1 Software Requirements

4.2 Hardware Requirements

5 System Design

5.1 Android Application

5.1.1 Application Structure

5.1.2 Frontend Features

5.1.3 Backend Features

5.1.4 Integration of Frontend and Backend

5.2 Inception v3 Neural Network

5.3 Application Programming Interfaces

5.3.1 TensorFlow Lite API

5.3.2 Firebase API

5.3.3 Google Text-to-Speech API

5.3.4 Google Speech-to-Text API

6 Project Implementation

6.1 Project Flow

6.1.1 Phase I

6.1.2 Phase II

6.1.3 Phase III

6.2 Android Application Implementation

6.2.1 Design Ideologies

6.2.2 Application Phases

6.2.3 Database Structure

6.3 Inception v3 Neural Network

6.3.1 Data Collection

6.3.2 Model Training

6.3.3 Model Predictions

7 Project Snapshots

7.1 Landing Page

7.2 Settings/Preferences Page

7.3 Input Methods

8 Conclusion

9 References