ISSN: 2455-2631 © February 2019 IJSDR | Volume 4, Issue 2

Augmented Reality Text Translation for Indian Languages

[1]Apoorva Teli, [2]Jinal Udvadia, [3]Anushree Baliga, [4]Suvarna Bhat

[1,2,3]Student, [4]Assistant Professor
Department of Computer Engineering
Vidyalankar Institute of Technology, Mumbai, India

***Abstract*: Often the information displayed in local places (e.g. street nameplates, shop names, banners etc.) is in their regional language which is very difficult for visitors/tourists to understand. For e.g. when one visits South India, all the nameplates are in their local language. So, a tourist has to hire a translator or take help from the local public who know English. We would be performing real-time text translation where a user points his camera towards the foreign text. Our app will quickly scan and identify foreign text, and then translate the text to English, and overlay the translated text on the screen. We will be using a well-suited method to correctly find the text located in a frame of the camera stream retaining its position, and it will be put together with OCR and a text translation service.**

***Index Terms*: Augmented Reality, Text Translation, Cloud Translation, OCR, Tesseract, Text Detection, Android.**

I. Introduction

One of the most common communication techniques is written text. When an individual encounters text in a completely unfamiliar language (See Fig. 1), manually typing it for translation becomes inconvenient. We devise a system that automatically translates the visual text in order to settle this difficulty.

Figure 1 Example of a poster in Gujarati and shop banner in Kannada

We present an android augmented reality (AR) translation/transliteration system, using the camera which lets the user translate the language on the fly only by pointing it toward the foreign text to view its translation and transliteration in English. Our application supports the translation of the following languages: Hindi, Kannada, Gujarati, Tamil, Telugu.

Our android application supports minimum sdk version 5.0 for more robust and seamless integration of OCR library. For OCR, we are using the Tesseract engine with trained model for Indian languages. For augmented reality overlay, we are using the Unity framework which is written in C#.

II. Overview

Augmented reality is technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view [1]. It is the combination of digital data along with the user’s surroundings in real time. Dissimilar to virtual reality, which generates a completely artificial domain, augmented reality uses existing surrounding and overlays new data over it. It is essential to get an overview of the advancements in augmented reality as it is a very promising technology. Following is a brief history of augmented reality:

AR in the 1960s. In 1968 Ivan Sutherland and Bob Sproull created a first head-mounted display, they called it The Sword of Damocles. Obviously, it was a rough device that displayed primitive computer graphics.

AR in the 1970s. In 1975 Myron Krueger created Videoplace – an artificial reality laboratory. The scientist envisioned the interaction with digital stuff by human movements. This concept later was used for certain projectors, video cameras, and onscreen silhouettes.

AR in the 1980s. In 1980 Steve Mann developed a first portable computer called EyeTap, designed to be worn in front of the eye. It recorded the scene to superimposed effects on it later and show it all to a user who could also play with it via head movements. In 1987 Douglas George and Robert Morris developed the prototype of a heads-up display (HUD). It displayed astronomical data over the real sky.

AR in the 1990s. The year 1990 marked the birth of the "augmented reality" term. It first appeared in the work of Thomas Caudell and David Mizell – Boeing company researchers. In 1992 Louis Rosenberg of the US Air Force created the AR system called "Virtual Fixtures". In 1999, a group of scientists led by Frank Delgado and Mike Abernathy tested new navigation software, which generated runways and streets data from a helicopter video.

AR in the 2000s. In 2000 a Japanese scientist Hirokazu Kato developed and published ARToolKit – an open-source SDK. Later it was adjusted to work with Adobe. In 2004 Trimble Navigation presented an outdoor helmet-mounted AR system. In 2008 Wikitude made the AR Travel Guide for Android mobile devices.

AR today. In 2013 Google beta tested the Google Glass – with internet connection via Bluetooth. In 2015 Microsoft presented two brand new technologies: Windows Holographic and HoloLens (an AR goggles with lots of sensors to display HD holograms). In 2016 Niantic launched Pokemon Go game for mobile devices. The app blew the gaming industry up and earned $2 million in a just first week [2].

We studied some examples of AR applications to know about the impact of its use in text translation:

1. *TranslatAR: A Mobile Augmented Reality Translator* - Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, Matthew Turk: In this paper, a prototype for a real-time mobile, visual translation system is presented that replaces the original text in the live camera stream, matching background and foreground colours estimated from the source images [3].
2. *Automatic text detection for mobile augmented reality translation* - Petter, Fragoso, Turk, Baur: This system improves upon the TranslatAR app as it automatically finds the text for translation. It scans the input image until it finds an area of interest that comprises text, enlarges the area of interest and produces the final bounding box [4].
3. *Word Recognition Incorporating Augmented Reality For Linguistic E-Conversion* - Faustina Jeya Rose. R, Bhuvaneswari. G: This system uses smartphone's camera for detecting English text and overlays its Tamil translation along with the meaning of the captured word [5].
4. *Mobile Camera Based Text Detection and Translation* - Derek Ma, Qiuhau Lin, Tong Zhang: An Android based application for real-time text extraction, recognition and translation from English to Chinese and is suitable for large text such as road signs etc [6].

III. Tools and Technologies

A. Augmented Reality

Augmented reality apps are written in special 3D programs that allow the developer to tie animation or contextual digital information in the computer program to an augmented reality "marker" in the real world. When a computing device's AR app or browser plug-in receives digital information from a known marker, it begins to execute the marker's code and layer the correct image or images.

AR applications for smartphones typically include global positioning system (GPS) to pinpoint the user's location and its compass to detect device orientation. Sophisticated AR programs used by the military for training may include machine vision, object recognition and gesture recognition technologies [7]. With the increasing device processing power, enhanced audio-video capabilities of a smartphone, in-built camera available on a phone can be optimally utilized for realizing a mobile augmented reality application [5].

B. Tesseract

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0, and development has been sponsored by Google since 2006 [8]. The language packages are called 'tesseract-ocr-langcode' and 'tesseract-ocr-script-scriptcode', where langcode is three letter language code and scriptcode is four letter script code. Tesseract needs traineddata files which support the legacy engine, for example those from the tessdata repository [9]. Tesseract is a very popular OCR because it is free and supports many languages.

C. Cloud Translation API (formerly Google Translate API)

The Cloud Translation API provides a simple programmatic interface for translating an arbitrary string into any supported language using state-of-the-art Neural Machine Translation. It is highly responsive, so websites and applications can integrate with Translation API for fast, dynamic translation of source text from the source language to a target language (such as French to English). Language detection is also available in cases where the source language is unknown. The underlying technology is updated constantly to include improvements from Google research teams, which results in better translations and new languages and language pairs [10].

D. Android Platform (minimum SDK API 21: Android 5.0)

Android is the most successful and popular mobile operating system used widely by almost 88% [11] of original equipment manufacturers (OEM) across the globe. Mobile applications for android platform can be developed using Android SDK provided by Google. Android SDK includes tools required for building, packaging and running android applications on a real device. It also includes APIs, emulator and debugging tools necessary for any android application developer. Considering the popularity and availability of low-cost Android-based smartphones in India, proposed mobile augmented reality application for text translation is to be developed for Android platform [5].

E. Unity 3D (version 2018.3)

Unity 3D is a gaming and movie software (Like Maya). Unity is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites. Supported platforms include BlackBerry 10, Windows Phone 8, Windows, OS X, Android, iOS, Unity Web Player (including Facebook). Using Unity we can also export the .APK file in mobile memory. The language that's used in Unity is called C# (pronounced C-sharp). All the languages that Unity operates with are object-oriented scripting languages [12].

F. Vuforia Engine (Version 7.5)

Vuforia Engine is a software platform for creating Augmented Reality apps. Developers can easily add advanced computer vision functionality to any application, allowing it to recognize images and objects, and interact with spaces in the real world. The Vuforia text recognition engine relies on the UTF-8 character encoding standard and can recognize any character listed in characters supported by text recognition [13].

IV. IMPLEMENTATION DETAILS

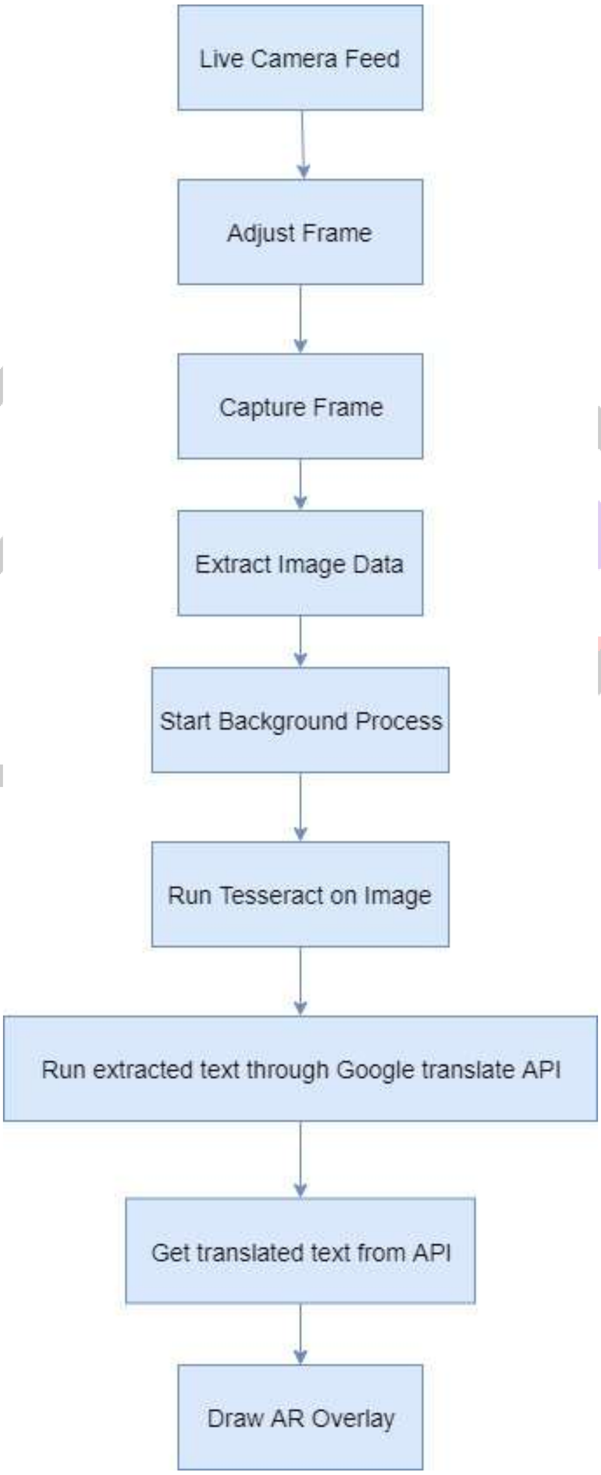

Figure 2 System Flow

A. Optical Character Recognition (OCR)

The Tesseract OCR engine is originally developed as proprietary software at Hewlett-Packard between 1985 and 1995 and has been sponsored by Google since 2006 [9]. tess-two is a fork of Tesseract Tools for Android that adds some additional functions. In the background grey-scaling, edge detection, line segmentation, word segmentation, character segmentation and detection part are done by Tesseract OCR engine.

An overview of Tesseract OCR engine:

The first step is a connected component analysis in which outlines of the components are stored. This is a computationally expensive design decision at the time but has a significant advantage: by inspection of the nesting of outlines, and the number of child and grandchild outlines, it is simple to detect inverse text and recognize it as easily as black-on-white text. Tesseract is probably the first OCR engine able to handle white-on-black text so trivially. At this stage, outlines are gathered together, purely by nesting, into Blobs. Blobs are organized into text lines, and the lines and regions are analyzed for fixed pitch or proportional text.

Text lines are broken into words differently according to the kind of character spacing. Fixed pitch text is chopped immediately by character cells. The proportional text is broken into words using definite spaces and fuzzy spaces. Recognition then proceeds as a two-pass process.

In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page. Since the adaptive classifier may have learned something useful too late to contribute near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces and checks alternative hypotheses for the x-height to locate small-cap text [14].

B. Text translation

For the translation part, we need a translation API so that it can take the Indian language text as an input and return the corresponding English translation of that input text. We have chosen the Google translation API to translate the text. To enable this feature into your application you need to be online.

The Google Translate API takes the source language code, target language code and the source text as its parameter. We set the source language code to the language selected by the user through a drop-down menu, and target language code to eng for English. After that, it translates the Indian language text into English. If internet connectivity is lost, then the application shows “Translation Unavailable”.

C. Translation Overlay

Unity is a cross-platform game engine developed by Unity Technologies. To overlay the translated text on the app screen, we use the Unity engine. During the live stream, the foreign text acts as an image target and is overlaid by a 2D plane which contains the processed translated text.

V. EVALUATION

The performance of our system will be evaluated by the rate of successful recognition. We decide to use recognition rate rather than successful translation rate as the criterion of performance, because the recognition rate more directly measures the successfulness of OCR engine, whereas the measure of translation rate can be influenced by the Google translation engine, over which we have no control. The recognition rate is defined as the ratio between the number of successfully recognized letters and the total number of letters in a test image.

VI. CONCLUSION AND FUTURE SCOPE

We have proposed an augmented reality-based android application, which can detect Indian language text from the live camera stream and the English translation is returned live. We will develop this application using open source tools such as tess-two (a fork of Tesseract tools for Android), Tesseract OCR engine and Cloud Translation API.

Our application will function relatively better for text of greater size than that of smaller size. This can be made more efficient by supplying training images with smaller DPI. If we could train the model with a smaller DPI, it may give us desirable efficiency for small text also. The precision can also be lost due to inadequate lighting and focusing issues and it may also display improper text. We will be trying to identify just a narrow range of font styles for this version of our application. In the revised version, we will try to operate on a wider range of font styles.

Also, a dictionary like Wikipedia, Oxford and so on, can be embedded in the application so that after the application has finished translating, the dictionary will generate the meaning of that word.

REFERENCES

[1] “What is Augmented Reality?”, WNO, 2018. [Online]. Available: https://wno.org.uk/news/what-is-augmented-reality.

[2] “What is Augmented Reality (AR) and how does it work”, ThinkMobiles. [Online]. Available: https://thinkmobiles.com/blog/what-is-augmented-reality/.

[3] Fragoso, V. Gauglitz, S. Zamora, S. Kleban, J. and Turk, M. “TranslatAR: A mobile augmented reality translator”, IEEE Workshop on Applications of Computer Vision (WACV 2011), 2011.

[4] Petter, M. Fragoso, V. Turk, M. Baur, C. "Automatic text detection for mobile augmented reality translation", 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011.
[5] Faustina Jeya Rose.R, Bhuvaneswari.G. "Word recognition incorporating augmented reality for linguistic e-conversion", 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.
[6] Ma, D. Lin, Q. Zang, T. "Mobile camera based text detection and translation", 2011.
[7] Margaret Rouse, "augmented reality (AR)", WhatIs, 2016. [Online]. Available: https://whatis.techtarget.com/definition/augmented-reality-AR.
[8] "Tesseract (software)", Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Tesseract_(software).
[9] "Tesseract open source OCR engine (main repository)", GitHub. https://github.com/tesseract-ocr/tesseract.
[10] "Cloud Translation", Google. [Online]. Available: https://cloud.google.com/translate/.
[11] "Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018", Statista. [Online]. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/.
[12] "Coding in C# in Unity for beginners", Unity. [Online]. Available: https://unity3d.com/learning-c-sharp-in-unity-for-beginners.
[13] "Getting started with Vuforia engine in Unity", Vuforia. [Online]. Available: https://library.vuforia.com/articles/Training.
[14] Ray Smith "An overview of the tesseract OCR engine", Google. [Online]. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/33418.pdf