Computer Vision based Text to Language Translation

Devanshu Roy[1], Dhanyata S. Nayak[2], Bhuvaneshwari G.[3], Chandana B. N.[4], Dr. Shantala Devi Patil[5]

[1,2,3,4]*School of Computer Science Department,* [5]*Associate Proffesor, Reva University, Bangalore, India*

Abstract— Blind people face immense difficulty understanding and comprehending text due to inability to see said text, and need external assistance to guide them with the same. Also, tourists face the obvious issue of language gaps and comprehension difficulties of the local language. The goal here is to give users a way to understand any text with their own comfortable language. This paper proposes an application that uses a fast and accurate text to language translation system and uses a text-to-speech library to speak the translated text out to the user. In this work, we propose Computer Vision to capture and detect text from cameras and Tesseractv5, an OCR based on a Long Short-Term Memory (LSTM) to classify said text into alphabets. Then we use translation on said text, to the desired language and use Text-to-Speech libraries to output the same in computer generated speech. Furthermore, also a Python module has been developed that can be used to convert English text to computerized Braille script.

Keywords—Language translation, OCR, Text-to-Speech, Computer Vision, OpenCV, computerized Braille.

1. INTRODUCTION

Billions of people in the world are visually impaired. It is a huge struggle for them to understand texts written on a piece of paper, and thus forcing them to use systems like Braille text, to feel and understand text. Although there are many electronic equipment that can assist in reading, there are few to actually translate the same for them. Also, life would be more convenient if said blind people could use a computer program to read out that same text to them, thus making them independent in the process.

Also, in today's world, where travelling is so common between various nations, language gaps are a challenge to a satisfied tourism experience. Also, the biggest issue tourists might face when they visit a certain place, is to understand the texts in the national/local language of their own place. This paper proposes a Computer Vision solution to capture and detect text from cameras and Tesseractv5, an OCR based on a Long Short-Term Memory (LSTM) which is a type of Recurrent Neural Network (RNN) to classify said text into alphabets. Then translation is used on said text using Python module 'translate', to the desired language and use Text-to-Speech library, called gTTS in Python, to output the same in computer generated speech.

Furthermore, for people who are both blind and deaf, a Python module has been developed that can convert English text to Braille. This means the translation from source language can be done to English, then converted to Braille text, which can have further applications for comprehension of the text to the blind and deaf user.

2. LITERATURE SURVEY

Handwriting-feature-based algorithms have been designed to approach the handwriting problem [1] as if it were a voice recognition problem, that is, by treating the input as a signal.

User input in handwriting recognition has a defined beginning and end, similar to how data points are segmented [2].

Optical character recognition (OCR) is a technique for converting scanned or printed text images[3], as well as handwritten text, into editable text that may be further processed.

The output of OCR is the text, which is saved in a file when the text is converted to speech (speech. txt). To transform the text to speech, Festival software[4] is utilised.

[5] illustrates a practical use of these technologies, in which a TTS approach is leveraged to provide voice-assisted text reading, followed by OCR to detect it.

Edge AI accelerators like the Neural Compute Stick-2, OpenVINO, and TensorflowLite are featured in [6,] as well as smart depth sensors such as OpenCV AI Kit-Depth.

It was suggested that a person with a visual impairment could benefit from a low-cost mobile application[7]. Optical character recognition (OCR) and text-to-speech (TTS) technology can be used in the smartphone app. The users will hear an alarm sound to let them know what occurred in the mobile application.

3. TESSERACT OCR

Tesseract [8] (latest versions 5.x.x) is an Open-Source Optical Character Recognition (OCR) Engine that is licensed under Apache 2.0 license. Starting from version 4.x.x, it has implemented an LSTM Deep Learning model, that is based on a Recurrent

Neural Network (RNN), giving fast and accurate results. The inclusion of the RNN LSTM has also given the users with the facility to train their own data sets and models, to be used with Tesseract OCR Engine.

4. METHODOLOGY

Firstly, camera input is taken, image is extracted and preprocessed. Alternatively, an image stored in local storage can also be used. Then, a competent Optical Character Recognition algorithm called Tesseractv5 is implemented to extract text from the image. Tesseractv5 is an Open-Source OCR algorithm.

The text is then processed, and then translated using translation APIs, one of the best being Python "translate" API.

Translated text then is fed to a Text-to-Speech API, to speak out the translated text to the user. gTTS Python module, based on Google Text-to-Speech is a competent API for the same. Refer the below figure 1.1 for the flowchart of the workflow of the translation module.

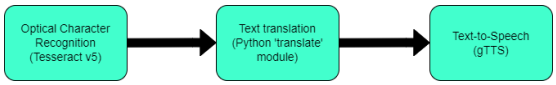


Fig 1.1 – Flowchart of Workflow

The Python library "pytesseract" is used to make usage of Tesseract in Python. pytesseract (Python Tesseract) is a wrapper for the Open-source OCR engine called Tesseract.

OpenCV is also used, which is an Open-source Computer Vision library. It is used to read the image file locally, or via camera input, and then resize the image to a smaller resolution than the system resolution. OpenCV also has a functionality to eliminate noise in captured images by blurring the image using Gaussian blur.

```
void cv::GaussianBlur ( InputArray   src,
                        OutputArray  dst,
                        Size         ksize,
                        double       sigmaX,
                        double       sigmaY = 0 ,
                        int          borderType = BORDER_DEFAULT
                      )
```

Fig 1.2 – cv2.GuassianBlur() official Syntax

This helps eliminate noise in images by smoothening the edges in the images.

Hence resizing and gaussian blurring is the necessary pre-processing applied on the images.

After detecting the text from image, this text is then stored in a variable, and pass this to a translate module. Python 'translate' module is the module of choice. Then the Translator class from the module is imported. Then an object is created with this class, and it has a method 'obj.translate(to_lang="", from_lang="")' that takes in two arguments, to_lang to specify the destination language and from_lang to specify the source language. The package 'langdetect' is used, which is a Python library to auto detect the language from the OCR text, and then specify the user's selected translation language, then pass them in the function.

The output from this function is a text object, containing the translated text. Now, a Text-to-Speech library (gTTS) in Python is used to convert this text to computer speech. gTTS generates .mp3 files with the computer speech, and these files are to be deleted after code execution. Hence os.remove in Python is used, to remove the generated audio files (.mp3 files). This requires importing of the 'os' Python library.

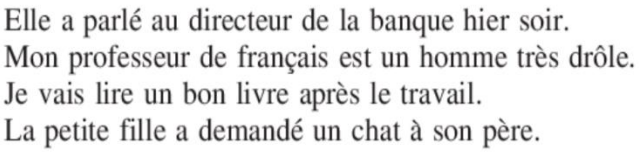
A Python module 'braille.py' is also developed which contains a method 'toBraille(text)'. This takes in text input in English, and return the corresponding Braille output using the ASCII characters equivalent of the Unicode for the Braille characters. The Braille ASCII characters were taken from Calculla's website [9].

5. EXPERIMENTAL RESULTS

The resultant program was tested with multiple test input images:

French to English:

Input image:



Output from OCR:

```
In [1]: runfile('C:/dROY_stuff/Education/MajorProject/C
Education/MajorProject/Code/MajorProj')
Detected language = fr
Elle a parlé au directeur de la banque hier soir.

Mon professeur de francais est un homme trés dréle.
Je vais lire un bon livre aprés le travail.

La petite fille a demandé un chat a son pére.
```

Translated output:

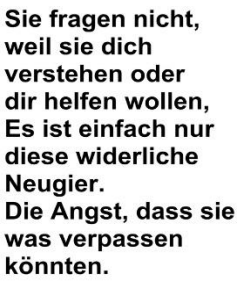
```
She spoke to the bank manager last night.

My French teacher is a very good man.
I'm gonna read a good book after work.

The little girl asked her father for a cat.
```

German to English:

Input image:



Output from OCR:

```
Education/MajorProject/C
Detected language = de
Sie fragen nicht,
weil sie dich
verstehen oder

dir helfen wollen,
Es ist einfach nur
diese widerliche
Neugier.

Die Angst, dass sie
was verpassen
konnten.
```

Translated output:

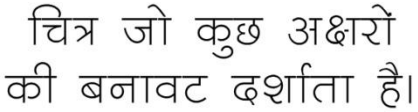
```
They do not ask,
because they ask you
or

want to help you,
It's just
this disgusting
curiosity.


what could be missed.
```

Hindi to English:

Input image:



Output from OCR:

```
In [1]: runfile('C:/dROY_s
Education/MajorProject/Cod
Detected language = hi
चित्र जो कुछ अक्षरों
की बनावट दर्शाता है।
```

Translated output:

```
Pictures that contain a few letters The texture of the carpet is imposing.
```

(Note: Indian language translation isn't fully accurate, there is a scope for improvement)

English to Braille:

```
PS C:\dROY_stuff\Education\MajorProject\Code\MajorProj>  & 'C:\Users\
\lib\python\debugpy\launcher' '63384' '--' 'c:\dROY_stuff\Education\Ma
Please enter some English text:This is a test for English to Braille.
Entered text:  This is a test for English to Braille.
Braille text:  ⠠⠞⠓⠊⠎ ⠊⠎ ⠁ ⠞⠑⠎⠞ ⠋⠕⠗ ⠑⠝⠛⠇⠊⠎⠓ ⠞⠕ ⠃⠗⠁⠊⠇⠇⠑⠲
```

This Braille text was verified online to translate back to the input text.

Since this approach uses Tesseractv5, which is an Open-source OCR software based on an RNN architecture, it leverages a Recurrent Neural Network for the OCR detection, hence giving us more accurate results than competing free solutions for OCR. Furthermore, with the introduction of v5.x.x, the operations were reduced from double operations to float operations, which reduced the resource overhead, hence it gave a better and faster performance compared to v4.x.x of Tesseract.

6. Conclusion and Future Enhancement

So, this program was able to detect, translate and output text in an image file, captured either from a camera input, or stored locally. The braille module was also successfully able to convert English text to corresponding Braille output. The Braille output is limited to Grade 2 Braille only. Furthermore, there are some translation inconsistencies with Indian languages, which have future scope of improvement.

With sufficient image data sets, hand writing detection can also be implemented by training the data sets using the LSTM model built into Tesseract OCR.

7. Acknowledgement

It is our privilege to express our sincere gratitude to our major project guide, Dr. Shantala Devi Patil, for sharing her valuable inputs, guidance, cooperation and constant encouragement throughout the duration of our project.

We express our sincere gratitude to our Head of Department Prof. Sunil Kumar S. Manvi for allowing us to present the project on "**Computer Vision Based Text to Language Translation**".

We also take this opportunity to thank all our lecturers who have helped us in this project.

We thank and pay respect to our parents and friends for their support and encouragement throughout.

Last but not the least we would like to thank all those who helped us directly or indirectly throughout the project.

8. References

[1] Starner, T., Makhoul, J., Schwartz, R., and Chou, G. "On-line cursive handwriting recognition using speech recognition methods" in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994. ICASSP-94., 1994, vol. v (apr 1994), V/125 –V/128 vol.5.

[2] Plamondon, R., and Srihari, S. "Online and off-line handwriting recognition: a comprehensive survey" in *Pattern Analysis Machine Intelligence, IEEE Transactions* on 22, 1 (jan 2000), 63–84.

[3] Archana A, Shinde D. "Text pre-processing and text segmentation for OCR" in *Int. Journal Computer Science Engineering Technology*. 2012:810–12.

[4] Kamesh DBK, Nazma SK, Sastry JKR, Venkateswarlu S. “Camera based text to speech conversion, obstacle and currency detection for blind persons” in *Indian Journal Science Technology*. 2016 Aug; 9(30)

[5] R. Ani, E. Maria, J.J. Joyce, V. Sakkaravarthy and M.A. Raja, "Smart Specs: Voice Assisted Text Reading system for Visually Impaired Persons Using TTS Method" in *IEEE Int. Conf. Innovations Green Energy Healthcare Technologies (IGEHT)*, Mar. 2017.

[6] Jagadish K. Mahendran, Daniel T. Barry, Anita K. Nivedha, Suchendra M. Bhandarkar; “Computer Vision-Based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recognition (CVPR) Workshops*, 2021, pp. 2418-2427.

[7] Mohd Nadhir Ab Wahab, “Text Reader for Visually Impaired Person” in *5th Int. Conf. Electronic Design (ICED) 2020*, Perlis, Malaysia. Ser. 1755 012055

[8] Singh, K. D., & Ahmed, S. T. (2020, July). Systematic Linear Word String Recognition and Evaluation Technique. In *2020 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0545-0548). IEEE.

[9] Accessed: Apr. 5, 2022. [Online]. Available: https://calculla.com/ascii_braille