International Journal of Future Generation Communication and Networking
Vol. 13, No. 1, (2020), pp. 869 - 876

OCR, Translation and Text to Speech to Aid Illiterate Drivers

K.N.V. Satyaranayana[1], Dr. P V Rama Raju[2], M.V. Santosh Naidu[3], L. Pradeep[3], K. Abhilash[3], K. Chaitanya[3]

[1]Asst.Professor, [2]Professor, [3]B.Tech students
[1,2,3]Department of ECE, SRKR Engineering College (A), Bhimavaram, India.

Abstract

People travel to different places every day. Most of them travel to new places as a part of profession. Drivers move from one state to another and they need to understand languages of other states. A person can't understand all the languages and a lot of them can't read their own language. This article addresses these issues by taking the picture of unknown language script, translate it into the required language and give speech output of the translated text. The captured image is pre-processed using OpenCV, the text is extracted using Google Vision, the extracted text is translated using Google Translator and the speech output is obtained through Google text to speech.

Keywords: *Optical character recognition, translation, text to speech, OpenCV.*

1. Introduction:

Literacy rate in India is just 73 percent according to 2011 census. It implies almost 300 million people are illiterate. Even urban literacy rate is just at 84 percent. India is a diverse country with 22 official languages. Hence, when travelling across India, language might be a problem. For drivers, travelling is a profession and they encounter this problem frequently. All the sign boards, city names and banners are also in state's native language. A few people can't even read their own language script. The proposed model aims at solving this problem by designing an Android application which takes the image of the unknown script using the mobile camera, translating the unknown script into their native language and to aid drivers who cannot read the script, a voice reads the translated text.

This paper is organized into five sections which also includes this section; Section 2 illustrate related work; Section 3 explains the proposed model; Section 4 presents the experimental results; Section 5 summarizes the findings.

2. Related work:

Venkateswarlu, S. & Duvvuri, Duvvuri B K Kamesh & Jammalamadaka, Sastry & Rani, R. (2016)[1] developed an innovative, efficient and real-time cost beneficial technique that enables user to hear the contents of text images instead of reading through them. It combines the concept of Optical Character Recognition (OCR) and Text to Speech Synthesizer (TTS) in Raspberry pi. K. H. Aparna, V. Subramanian, M. Kasirajan, G. V. Prakash, V. S. Chakravarthy and S. Madhvanath [2] developed a system for online recognition of handwritten Tamil characters using by comparing the unknown stroke of a handwritten character and comparing it with the available database of strokes. Sasikumar, Aravind. (2015) [3] designed a cost effective text to speech system to help visually impaired people using Matlab which reads the text and read the text aloud. Haque, S. M [4] worked on automatic detection of Bengali text on roads for visually impaired and speech is synthesized for the translated text. M. Nagamani, S. Manoj Kumar, S. Uday Bhaskar [5] designed a system to recognize digits, convert them into telugu language script and synthesize speech output for the telugu script. S. C. Madre and S. B. Gundre [6] proposed a method to convert text using segmentation and extraction and convert the text character into audio signal. T. Yamaguchi, Y. Nakano, M. Maruyama, H. Miyao and T. Hananoi [7] presented a digital recognition system which recognizes telephone numbers written on sign boards. Badhe, Darshan and Pravin M Ghate [8]

ISSN: 2233-7857 IJFGCN

designed a system to recognize Marathi text and synthesize voice output for the recognized text. K. Nirmala kumari, Meghana Reddy J [9] designed a device which performs optical character recognition and text to speech system using raspberry pie. Rithika.H, B.Nithya Santhoshi, [10] designed a model which performs optical character recognition, translates the recognized text and synthesizes the text in desired language.

3. Proposed Model:

3.1.1 Preprocessing:

Preprocessing is binarization of image for character recognition. Image Binarization is converting a pixel image into a black and white image, reducing the information contained in the pixel image to black and white, a binary image. This is also known as image thresholding. But to perform this we need to convert the color image into grayscale image. For this we used Luminosity method as this is the best method when compared to average method as average method is simply taking the average of R, G, B values of the image. The value of the grayscale image is calculated using the formula

New grayscale image = ((0.3 * R) + (0.59 * G) + (0.11 * B))

This grayscale image is then binarized. Binarization works by finding a threshold value in the histogram. This value is estimated in such a way that the histogram is divided into two parts, each part representing the character and the background. Thresholding can be either simple thresholding (global thresholding) or adaptive thresholding. In simple thresholding, same threshold is applied for every pixel. For a pixel value smaller than the threshold, it is set to zero and for a pixel value greater than threshold it is set to one. But if image has different lighting conditions, information might get lost in thresholding. Hence adaptive thresholding is used. In Adaptive thresholding, the code automatically determines the threshold for a pixel based on a small region around it. So, we obtain different thresholds for different sections of the image. The threshold is either determined by calculating the mean of the neighborhood minus the constant C or calculating the guassian-weighted sum of the neighborhood values minus the constant C. Simple thresholding and adaptive thresholding are compared in the figure 1.

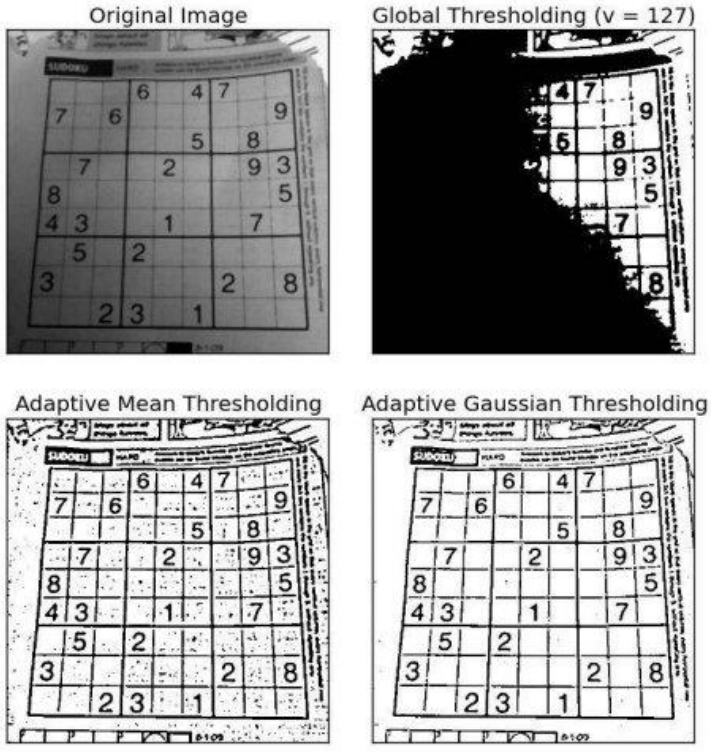


Figure 1. Comparing different thresholds

ISSN: 2233-7857 IJFGCN

As seen in the results, adaptive guassian thresholding gives better image. Hence, it is used for binarization. All this thresholding is achieved using OpenCV. With this, preprocessing of the image for better results is done. This preprocessed image is used to recognize characters in it.

3.1.2 Optical character recognition:

Optical character recognition (OCR) is a technology used to convert images, in the form of PDF files or images, to searchable, editable data. Paper documents—such as brochures, invoices, contracts, etc.—are sent via email. This process usually involves a scanner that converts the document to lots of different colors, known as a raster image. In order to extract the data and repurpose the content of the document, an OCR engine is necessary. The OCR engine detects the characters present in the image, puts those characters into words, and then into sentences, enabling you to search and edit the content of the document. Optical character recognition process includes image analysis.

Image analysis is the extraction of meaningful information from images mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face.

Google Vision is an optical character recognition engine, one of the most accurate OCR engines currently available. It is licensed under Apache v2.0 and has been developed by Google. Google vision can process right to left text such as Arabic or Hebrew, many Indic scripts as well. Google vision is suitable for use as a backend and can be used for more complicated OCR tasks including layout analysis.
This project supports Indian languages like Hindi, Bengali, Telugu, Tamil, Gujarati, Kannada, Malayalam, Marathi, and Nepali. This recognized text is now translated using Google translator.

3.1.3 Text translation using Google Translator:

Google translator is a free multilingual statistical and neural machine translation service developed by Google, to translate text and websites from one language into another. Google Translator can dynamically translate text between thousands of language pairs. Translation lets websites and programs programmatically integrate with the translation service. Google translation uses a translating method to a system called neural machine translation. It uses deep learning techniques to translate whole sentences at a time.

Neural machine translation:
Google Neural Machine Translation (GNMT) is a neural machine translation (NMT) system developed by Google and introduced in November 2016, that uses an artificial neural network to increase fluency and accuracy in Google. Google Translate's neural machine translation system uses a large end-to-end artificial neural network that attempts to perform deep learning, in particular, long short-term memory networks.
GNMT improves the quality of translation over SMT in some instances because it uses an example-based machine translation (EBMT) method in which the system “learns from millions of examples”. It translates whole sentences at a time, rather than just piece by piece. It uses this broader context to help it figure out the most relevant translation, which it then rearranges and adjusts to be more like a human speaking with proper grammar. The GNMT network can undertake Interlingua machine translation by encoding the semantics of the sentence, rather than by memorizing phrase-to-phrase translations.
By using millions of examples, GNMT improves the quality of translation, using broader context to deduce the most relevant translation. The result is then rearranged and adapted to approach grammatically based human language.
When Google Translate generates a translation proposal, it looks for patterns in hundreds of millions of documents to help decide on the best translation. By detecting patterns in documents that have already been

ISSN: 2233-7857 IJFGCN

translated by human translators, Google Translate makes informed guesses (AI) as to what an appropriate translation should be.
Now, speech is synthesized for the translated text using Google Text-to-speech.

3.1.4 Text to speech using Text-to-Speech Speech Synthesis:

Google Text-to-Speech converts text into human-like speech in more than 180 voices across 30+ languages and variants. It applies ground-breaking research in speech synthesis (WaveNet) and Google's powerful neural networks to deliver high-fidelity audio. It generates speech that mimics human voices and sounds more natural, reducing the gap with human performance by 70%.
Cloud Text-to-Speech is powered by WaveNet, software created by Google's UK-based AI subsidiary DeepMind. WaveNet uses machine learning to generate speech. It then waveforms from a database of human speech and re-creates them at a rate of 24,000 samples per second.

Deep learning-based Synthesis:
It is a synthesis process used by WaveNet for speech synthesis.
Given an input text or some sequence of linguistic unit Y-Target speech X can be derived by

$$X=\arg\max P(X|Y,\theta)$$ where θ is the model parameter.

The input text will first be passed to an acoustic feature generator, then the acoustic features are passed to the neural vocoder. For the acoustic feature generator, the Loss function is typically L1 or L2 loss. These loss functions put a constraint that the output acoustic feature distributions must be Gaussian or Laplacian. This is shown in figure 1. Here Loss (human) is the loss from the human voice band and is a scalar typically around 0.5.

$$loss = \alpha loss_{human} + (1-\alpha) loss_{other}$$

Loss function

Speech synthesis:
Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.
The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer.
A text-to-speech system is composed of two parts:
Front-end:
The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion.
Back-end:
Converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

ISSN: 2233-7857 IJFGCN

Typical TTS system:

A typical TTS system is shown in figure 2.

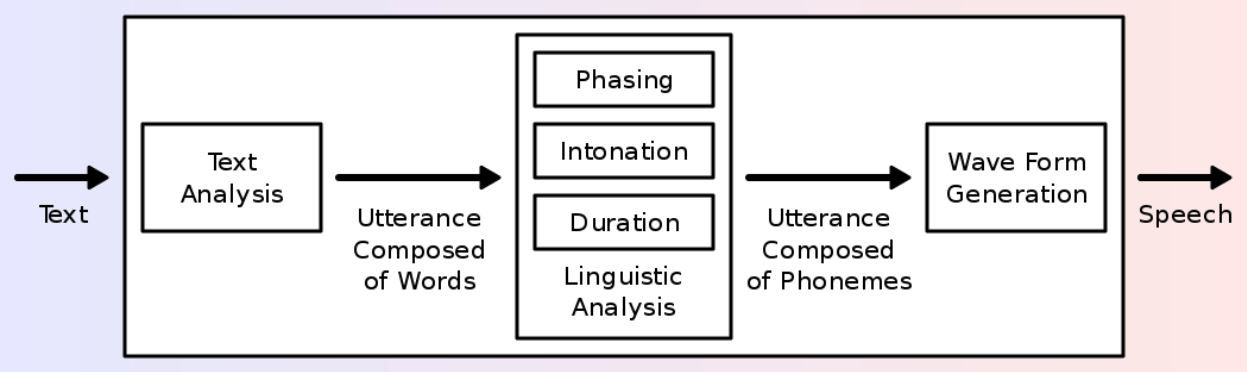


Figure 2. Typical TTS system

The synthesized text is our required output. The whole process flow is shown in the figure 3.

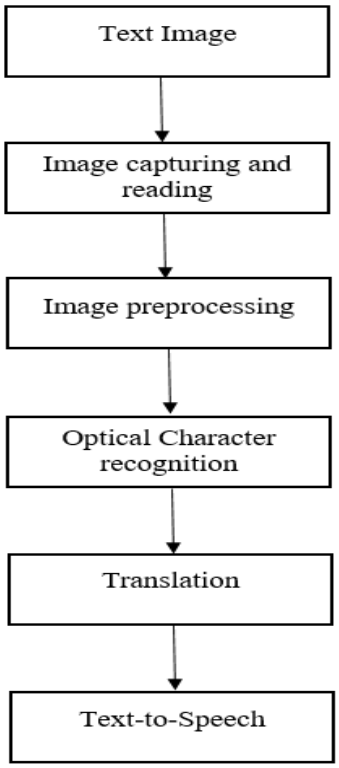


Figure 3. Process flow

4. Experimental Results:

The entire procedure consists of three stages. First we capture the image of the required text which is then preprocessed to get better results. The input image is shown in the figure 4.

ISSN: 2233-7857 IJFGCN

Figure 4. Input Image

First the image is converted into grayscale and the grayscale image is binarized using guassian adaptive thresholding. This image is binarized using OpenCV which inverts the color of the image. This is done to isolate the text from the image. The binarized image is shown in the figure 5.

Figure 5. Binarized image

The text from this image which is required to be translated is recognized using Google Vision and this result is shown in the figure 6.

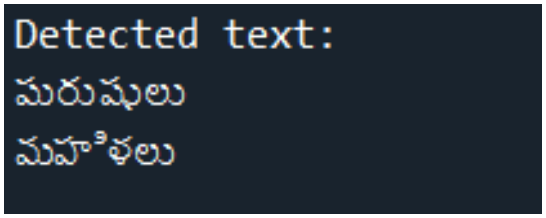


Figure 6. Recognized Text

Now this detected text is translated into user required language using Google Translate. Here it is translated into Hindi language. The result is shown in figure 7.

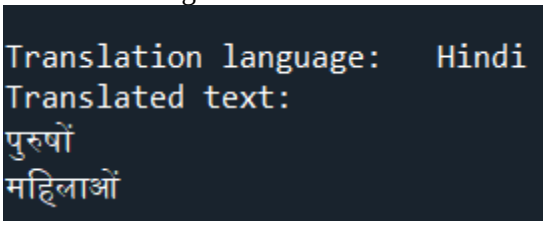


Figure 7. Translated Text

Using Text to speech the translated text speech is synthesized simultaneously and voice is heard through speaker.

ISSN: 2233-7857 IJFGCN

5. Conclusion:

The proposed application captures an image using smartphone camera, extracts the text from it using Google Vision, translates it into desired language using Google translate and synthesizes a voice output for the translated text using native android text to speech. It aids illiterate drivers in understanding the unknown language script with only an affordable smartphone. The preprocessing helps in getting steady output even if camera is not properly aligned to capture image.

REFERENCES:

[1] Venkateswarlu, S. & Duvvuri, Duvvuri B K Kamesh & Jammalamadaka, Sastry & Rani, R. (2016). Text to speech conversion. 9. 10.17485/ijst/2016/v9i38/102967.
[2] K. H. Aparna, V. Subramanian, M. Kasirajan, G. V. Prakash, V. S. Chakravarthy and S. Madhvanath, "Online handwriting recognition for Tamil," Ninth International Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan, 2004, pp. 438-443.
[3] Sasikumar, Aravind. (2015). Text to Speech Conversion System using OCR. International Journal of Emerging Technology and Advanced Engineering (IJETAE),. 9001.
[4] Haque, S. M., et al. "Automatic detection and translation of bengali text on road sign for visually impaired." (2007)
[5] M. Nagamani, S. Manoj Kumar, S. Uday Bhaskar "Image to Speech conversion for Telugu language" International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 6, November (2013)
[6] S. C. Madre and S. B. Gundre, "OCR Based Image Text to Speech Conversion Using MATLAB," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 858-861.
[7] T. Yamaguchi, Y. Nakano, M. Maruyama, H. Miyao and T. Hananoi, "Digit classification on signboards for telephone number recognition," Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., Edinburgh, UK, 2003, pp. 359-363 vol.1.
[8] Badhe, Darshan and Pravin M Ghate "Marathi Text to speech synthesis using Matlab" (2015)
[9] K. Nirmala kumari, Meghana Reddy J, "Image text to speech conversion using OCR Technique in raspberry pi", IJAREEIE, Vol.05, 2016.
[10] Rithika.H, B.Nithya Santhoshi, "Image Text to Speech Conversion in the Desired Language by Translation with Raspberry Pi" IEEE 2016.

ABOUT AUTHORS

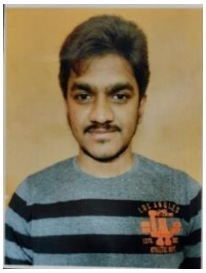
K. N. V. SATYANARAYANA

Presently working as assistant professor in Department of Electronics and Communication Engineering, S.R.K.R. Engineering College, Bhimavaram, A.P, India. He is currently pursuing PhD from Annamalai University. His current research interests include Image processing, Signal processing and Internet of things.

Dr. P.V. RAMA RAJU

Presently working as a Professor of Department of Electronics and Communication Engineering, S.R.K.R. Engineering College, A.P, India. His research interests include Biomedical Signal Processing, Signal Processing, Image Processing, VLSI Design, Antennas and Microwave Anechoic Chambers Design. He is author of several research studies published in national and international journals and conference proceedings.

ISSN: 2233-7857 IJFGCN



M. V. SANTOSH NAIDU

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R Engineering College, Bhimavaram, AP, India.

L. PRADEEP

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R Engineering College, Bhimavaram, AP, India.

K. ABHILASH

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R Engineering College, Bhimavaram, AP, India.

K. CHAITANYA

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R Engineering College, Bhimavaram, AP, India.

ISSN: 2233-7857 IJFGCN