Document Segmentation and Language Translation Using Tesseract-OCR

Sahil Thakare
Electronics and Telecommunication Department
SGGSIE&T
Nanded, India
2015bec053@sggs.ac.in

Ajay Kamble
Computer Science Department
SGGSIE&T
Nanded, India
2015bec071@sggs.ac.in

Vishal Thengne
Electronics and Telecommunication Department
SGGSIE&T
Nanded, India
2015bec604@sggs.ac.in

Mrs U.R.Kamble
Electronics and Telecommunication Department
SGGSIE&T
Nanded, India
urkamble@sggs.ac.in

***Abstract*—Document segmentation and Translation are one of the key areas in pattern recognition and natural language processing. This paper presents details about translation in terms of a web application that accepts image document as an input, where input document is a user define image file containing text in any language available in the Python-tesseract library and does its exact translation in any supported languages using Google Translator (i.e Googletrans). Python script and various libraries are used to approach various challenges in segmentation and translation of a document.**

Keywords—Tesseract, optical character recognition (OCR), Segmentation, Googletrans

I. INTRODUCTION

human-computer interaction and Machine learning are the most challenging research fields since the change for the better, over time of digital computers. In Optical Character Recognition (OCR), the text lines, words, and symbols in a document must be broken into parts properly before recognition. In English, a huge volume of textual data is available on the internet and the same is for Hindi and various other languages used worldwide. But, most of the available tools and techniques are English oriented. This shows that there is a lack of a tool that does efficient text mining for languages other than English. To overcome this constraint, in this paper we have proposed a software interface i.e web application, that does an automated translation of image document into any language of Googletrans library. Python-tesseract is an optical character recognition (OCR) tool for python software. That is, it will identify and read the text deeply set within part of images. whereas Google trans is a free and unlimited python library that implements Google Translate API. The web-application that can be built on the explain concepts will take document image file as an input and output the translation.

II. RELATED WORK

➢ *Pre-Processing of Document during Segmentation*

In the pre-processing phase, there is a series of operations performed on the scanned input image. It improves the image creating and displaying it good for division of something into smaller parts and after noise reduction the image is segmented [4].

An image is processed through a series of algorithms which are the steps to pre-processing

- De Skewing: Process by which the bounding box of scanned text is detected, image is then rotated so that document should follow a typical upright position.
- De-Noising: Process by which the noise coming from the image taking device is reduced. Noise can be present in different forms which have to be removed.
- Character Enhancing: De-noising can lead characters to reduce their clarity and loses their edges. So, characters edges must be enhanced by applying a sharpening image.
- Histogram Equalization: Depending on scanning device some part of the image may be more exposed than others. Which explains the reason why some part of the document is brighter than the others. Histogram Equalization process aims to balance the brightest and darkest regions of the image.
- Line Word Character Segmentation: The best work of OCR can be seen when individual characters are identified [5].

➢ *Character Recognition*

Once the position of the text is determined in the image, pattern recognition techniques are applied to correctly identify that particular character. There are two different types of pattern recognition techniques which is feature-based and feature-less techniques. Feature-based depend on explicit characteristics of character such as horizontal and vertical lines and line intersection. The feature of that particular character are compared with that of known character to identify the most closely related character.

The feature-based technique is also used for recognition of handwritten character with the use of trained models on SVM, k-nearest neighbours and Neural Networks [7]. The featureless techniques are used with documents having a consistent set of character fonts, or with the document having no inconsistencies in the image. This will identify the characters solely on the basis of the grey-scale value of character block[8].

Part No: CFP1858A-ART; ISBN No:978-1-5386-8492-4/18/$31.00 ©2018 IEEE

➢ *Segmentation*

The most important process here is the segmentation phase. This is done by separating individual characters from the image. Sometimes while dealing with handwritten characters the segmentation process gets difficult because of variability in paragraphs, words of line and characters of a word, size and skew. Different cases like when the words are attached or overlapped, this will increase the difficulty in segmentation task[4].

III. Review and Word Recognition of Tesseract

Tesseract is designed to be language-independent [10]. At the beginning aim of Tesseract was to recognize white on black. Which led to the design in way of connected parts analysis and operating on outlines of parts. Fig. 1 shows the basic components block diagram for Tesseract [10].

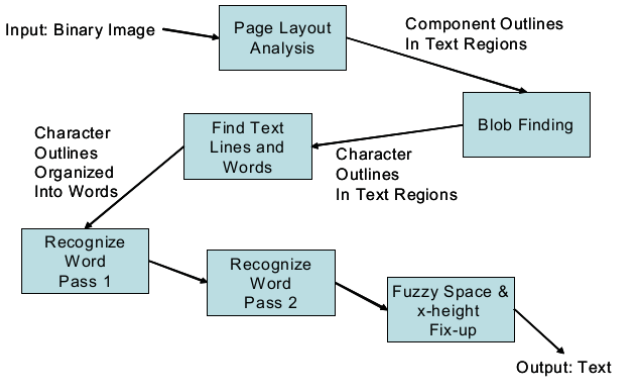

Fig. 1. Block diagram of basic components of Tesseract [10].

Once we finish CC analysis, we have to find blobs in the text region, which are the putative classifiable unit, and which can be overlapping Connected parts, and inner loop outlines or holes. After Detecting which outlines make blobs, the line finder detects lines by virtue of the vertical overlapping adjacent character on a line [11]. After text lines, a fixed-pitch detector checks for fixed pitch character layout and runs the algorithms of word segmentation according to the fixed persuade decision.

Fig.2 shows the block diagram for Tesseract word identifier [10]. Word recognizer treat each blob as different, and presents the results to a dictionary search to find a word in the combinations of classifier choices for each blob in the word. If the result is bad, further it chops the poorly recognized characters, where it improves classifier confidence. After all cutting possibilities are over, a better search of resulting segmentation graph puts back together chopped character fragments, or parts that were distinguished into multiple CCs. Each step, new blob combinations are classified, and the classifier results are given to the dictionary again. Output for each word is a character string that had the best overall distance based rating.

The words are processed twice. First time successful words are sent to an adaptive classifier for training. As soon as it has sufficient samples, it can provide classification output result, even on the first given pass. further , text which are not good which are to be given to pass 1 goes again for the one time, if the adaptive identifier has obtained more data after first pass over the word.

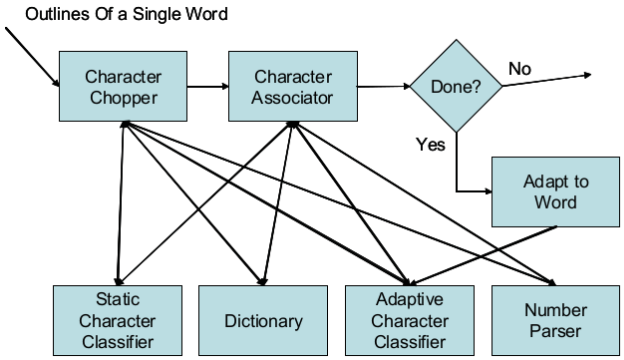

Fig. 2. Block diagram for Tesseract word identifier [10].

IV. Stages of Segmentation and Translation

➢ *Detection of Language on the given image*

The image is given to the web application where the library detects the language on the image and carry out the processing to extract the text from the image. Step by step procedure helps to extract text from the document. Character detection is the first step towards extracting text from the document. Starting from Line Finding algorithm to recognized skew page. Baseline fitting for handling pages with curved baselines. Fixed pitch text is chopped into characters using pitch. And many other algorithms are used by tesseract-OCR to detect character and read text from the document [9].

➢ *Extracting text from the image*

Tesseract-OCR is used for an image for an text to be extracted. Tesseract in its own is a very highly optimized group of algorithms. 'Python-tesseract' module in python implement tesseract-OCR to convert the image into text. Its implementation is simple to carry out, as we need to include the module and call the defined method i.e 'image_to_string(image,lang='lang_code')' which converts the given image to string. We have to explicitly provide the language of text available on the document [1]. Tesseract converts and returns the text available in an image [2].

V. Language Support

➢ *Googletrans Language Support*

A free polyglot machine translation service which is developed by google, to translate the text. It allows a user to use a interface for translation, Cell phone application for iOS and android, and an API for developers to build browser extensions and software applications.

Every language is determined by its language code called ISO-639-1 Code like Arabic - ar , Irish - ga , Marathi - mr , Hindi - hi. Google Translator is available for more than 100 languages and helps over 500 million people daily [6].

Part No: CFP1858A-ART; ISBN No:978-1-5386-8492-4/18/$31.00 ©2018 IEEE

➢ *Tesseract Type and Language Support*

Tesseract-OCR is highly efficient and easy to handle library with efficient implementation. The library function that we are using image_to_string(image_file,language) which supports '.jpg', '.png', '.gif', '.bmp', '.tiff', '.bmp'. The code for various languages in tesseract library are noted at https://cloud.google.com/translate/docs/languages[2][3].

VI. Result

With the help of python modules i.e. Python-tesseract, PIL and Googletrans a python script is created by which one image is taken as input and its characters are extracted and then this text is fetched to Googletrans to translate this text into another language. Fig.3 shows the raw python script for the same.

```
from PIL import Image
import pytesseract
import cv2
from googletrans import Translator

translator = Translator()
# Asking user to input image name
img_name = input("Input the name of the Image : \n")
#Asking user to input the language code for the text in the image(This
is can done dinamically in web apllication)
lan = input("Enter the language code for selected image : \n")
img = cv2.imread(img_name)
# Extracting text from the document
text = pytesseract.image_to_string(Image.open(img_name),lang=lan)
print("The text in the image file is as follows: \n")
print(text)
# converting that text into the required format
translated = translator.translate(text,dest='en')
print("\n\nThe translated Text will be like this = \n")
print(translated.text)
```

Fig. 3. Raw Python script for conversion of an image containing any language into English (we can decide the language of choice just by changing the value in the attribute of 'dest' in the translator. Translate method.

This is further implemented as a web application in which we can browse any image and ask the application to convert the image text into the required language text. Fig.4 is the input to the web-application. The link for the web application is http://mystikolabs.xyz:8080/.

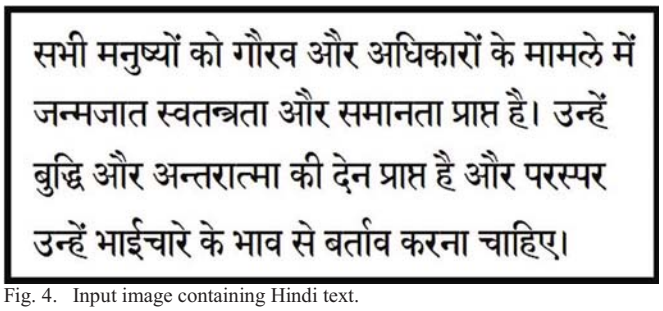

Fig. 4. Input image containing Hindi text.

One of the limitations of the proposed method is that the user has to provide the input language manually. To overcome this limitation, we are currently working on determining the input language automatically.

VII. Conclusion

Here we have presented a method to use segmentation and translation together in order to separate a document in such a way that it will reduce the complexity to understand a document and make that document easily available in the most understandable form anyone could need. The technology used for Optical Character Recognition will help to get that document readily converted into the characters which can then be translated to any language known to Google Translate API. We have shown that any document whose image is available with us can be read and translated by means of some python scripting and which will ultimately help anyone to understand it in his/her known language. Fig.5 is the output from the working web-application.

We can extend this application in such a way that we will have real time application in which we can directly convert any document into our known native language. Any unknown and non-understandable document can be translated within a fraction of seconds [6].

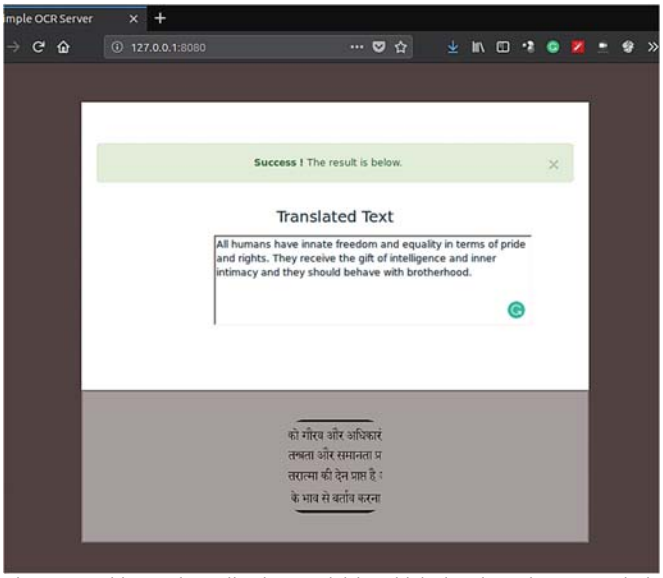

Fig. 5. Working web-application model in which the above input sample is given as input and the extracted text is then translated from Hindi ->English

Moreover, the language recognition has no bindings, not only Hindi as well as any native languages like Bengali, Marathi, Tamil, Gujarati, Meetei Meyak, Oriya, Santali etc.

REFERENCES

[1] R. smith, “An Overview of the Tesseract OCR Engine”, Proc.International Conference on Document Analysis and recognition,2007

[2] https://pypi.org/project/Python-tesseract/

[3] https://github.com/madmaze/Python-tesseract

[4] K. Elissa, Hiral Modi,M.C.parikh, “A Review On Optical Character Recognition Techniques”, International Journal of Computer Application,2017

[5] Charles Florin, "OCR(Optical Character Recognition - Digitizing Records)" ,White Papers, 2016

[6] https://pypi.org/project/Googletrans/

[7] T. de Campos, B. Babu, and M. Varma. Character recognition in natural images. In VISAPP, Feb. 2009. 1, 4, 5

[8] Wan, L., Zeiler, M., Zhang, S., LeCun, Y., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In *Proceedings of the international conference on machine learning (ICML'13)*.

[9] Y. M. Y. Hasan and L. J. Karam, "Morphological text extraction from images," in IEEE Transactions on Image Processing, vol. 9, no. 11, pp. 1978-1983, Nov. 2000.doi: 10.1109/83.877220

[10] Smith, R “Hybrid Page Layout Analysis via Tab-Stop Detection, Document Analysis and Recognition” Proc. 10 th Int. Conf. on Document Analysis and Recognition, 2009.

[11] Smith, R., “A simple and efficient skew detection algorithm via text row accumulation” Proc. 3 rd Int. Conf. on Document Analysis and Recognition, 1995, pp1145-1148.