American Journal of Electronics & Communication
Sayan Kumar Garai et. al., American Journal of Electronics & Communication, Vol. III (2), 8-11

A Novel Method for Image to Text Extraction Using Tesseract-OCR

Sayan Kumar Garai
skgarai789@gmail.com
Sayan Ghoshal
sayanghoshal2000@gmail.com
Ojaswita Paul
ojaswitapaul29@gmail.com
Neepa Biswas
biswas.neepa@gmail.com
Upayan Dey
upayan10@gmail.com
Dr. Sandip Mondal
sandip.mandal@uem.edu.in
University of Engineering & Management (UEM),
Department of CSE, Kolkata, India

***Abstract*— Text extraction process can play a vital role for detecting valuable information from a selected image. This text extraction process involves text detection, localization, marking, tracking, extraction, enhancement and finally recognition task. It is a difficult task to detect these text characters, because of their variation of size, style, font, orientation, alignment, contrast, color and textured background. There is a growing demand of information detection, indexing and retrieval from various multimedia documents nowadays. Several methods have been developed for extraction of text from an image. This article proposes a novel method for image to text extraction. In this paper, we are presenting a multiresolution morphology based text segmentation process suitable for various types of non-text elements like drawing, pictures, halftones or etc. For image processing, python library OpenCV is used and for text extraction Tessaract is used. Python Imaging Library (PIL) is capable to handle the opening and manipulation of images in many formats in Python. Also we are in testing of such an application that can give output in every language correctly.**

Keywords*— *Image Processing, Application, Text Segmentation, Python, Artificial Intelligence, Computer Vision

I. INTRODUCTION

Nowadays, many documents like newspapers, faxes, printed information, written notes are scanned and kept for backup. The scanned materials are stored as image formats which are not editable. Modify and note down all the important texts of that picture is very hard and time-consuming. So therefore there is a requirement of such application to extract the word from an image or scanned documents and prepare it in such a way that it is editable and saved easily. For this purpose, we are using OCR systems. OCR stands for Optical Character Recognition [1] which can read from images or scanned documents and convert it into an editable format. For over two decades, OCR systems are in wide use to provide automated text entry into computerized systems. OCR is a branch of Artificial Intelligence (AI), where AI [2] is a wide-ranging branch of computer science concerned with building smart machines which are capable of performing tasks that requires human like intelligence [3]. This transforms a two-dimensional image of text documents from its image representation into machine-readable text. We know that images take more storage space compared to word or text files. So it is essential to store the information in such a way as users can easily search and edit those data as well as save the space. As time is passing by the demand for applications that can recognize texts and characters from scanned images is increasing.

OCR has several sub-processes to perform so that it can provide accurate results. The sub-processes are:

- Preprocessing
- Text Localization
- Character Segmentation
- Character Recognition
- Post Processing

Tesseract OCR generally uses Long Short Term Memory (LSTM) which uses Artificial Recurrent Neural Networks [4]. This approach is generally used in natural language processing. LSTM uses feedback connection in neural network which helps to proves the entire sequence of data. It significantly reduces errors that are found during the process of character recognition. Tesseract pre-assumes that the input image is in a form of binary image and processing takes place. Tesseract OCR is an open source system which performs well in handwritten text also.

Tesseract engine stores the input images into binary format. At the first step of processing, it recognizes connected components and the outlines are embedded into blob which are organized into different text lines. Identified text lines are further separated according to its pitching. If a fixed pitch is detected between the characters then text recognition is done.

II. RELATED WORK

Text extraction using OCR is not a new method, many research developments are there on this selected domain. Connected components based method, Sliding window based method, Hybrid method, Edge based method, Color based method, Texture based method, Corner based method, Stroke based method etc. are various approaches used in OCR system.

One significant work has been done for Braille Translation has been done using this OCR tool [6]. Here the scanned images of old books are converted from grayscale to binary image using Adaptive Threshold. After that character recognition is done using Tesseract API followed by post processing spell checker API JOrtho.

Another research work for the Braille Code recogniser OCR is done [7]. For this development, skew angle detection and adaptive thresholding approaches are taken which can eventually help a lot of visually impaired people.

A new proposal was developed for detecting text from both computer generated image and natural image in article [12]. For detecting and recognize text they have utilized Maximally Stable Extremal Regions properties. Geometric and stroke width properties are used to reduce possible

regions. At last stage Tesseract OCR is used for separating text and non-text group.

There are many research papers on text extraction using Line Segmentation Method. In this method [8], each and every line is being detected first, and then they are converted into grey-scale image and then the text extracted. An effective method is developed for segmenting handwritten text lines originated from historical document images. Proposed method performs well with difficult hand writing also.

III. PROBLEM STATEMENT

The modelling and processing of the artificial intelligence is changing time to time. Whilst researching we came across lots of methods that are already used by our peers to achieve the same. We faced hurdles like recognition rather analysing the image, how can we separate each lines from one another and thereafter read words one by one. Problems like in case of an image with curved texts or to detect the entire page where we need to read the borders of the document. These are one of the many hurdles we faced while doing the project whose solution is discussed briefly. Flow chart of the proposed is represented in Figure 1.

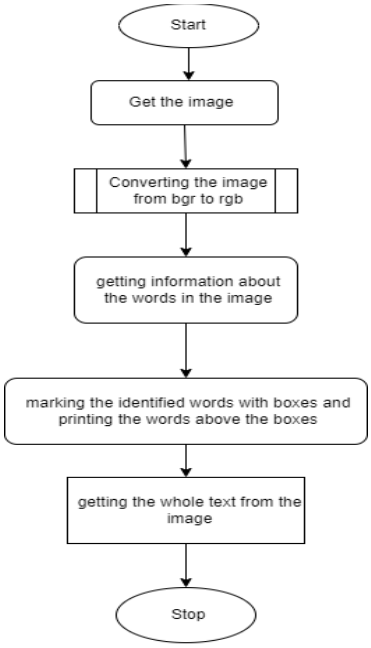

Figure 1: Flow chart of the proposed work

Preprocessing

We proposed the approach of Text segmentation using Tesseract OCR. Initially after the image is scanned we converted it into a bgr format from rgb image, for that helps us getting the near accurate recognition. Then we collect each and every information about each word in the image, and then we mark the words using a box around that, and we print the recognized text above the box. This helps in differentiating the text element and no-text element in the image so that it can be handled differently.

IV. PROPOSED SOLUTION

Our project is mainly an .exe file, and to install the executable file, we need to simply double click on it. To use our project, users don't need to install any other thing, or any other application.

Even, internet is also not required to run this executable file, and this is a small application so big amount space is also not needed to install this .exe file.

There are basically three types of solution related to these projects. The solutions are as follows:

Line Segmentation

The first step for this kind of problem statement is line segmentation [5]. It is a step for recognizing a document. In this process the handwritten lines or the printed lines of the documents are separated and are further processed for word segmentation. Further the document is sent for word recognition and other steps which are related to recognition of the documents. The line segmentation process has faced a lot of difficulties with noises present in the documents.

Tesseract OCR

Tesseract is an open source Optical Character Recognition (OCR) tool [9] for python which recognizes and reads the text which are present in an image. It is useful for all kind of images like .jpg, .jpeg, .png, etc. This tesseract OCR can be additionally used as a script which will print the recognized texts in place of writing it in a file.

Out of the above two process we have used the Tesseract OCR method as a solution to our project. Tesseract is used to extract texts. There are further two more methods of extracting the document under Tesseract OCR.

The two ways of extraction of document are:

i. Text Line Detection Method

Text line detection [10] based is a ridge based text line extraction method for the captured documents. The method consists of two algorithms: a) Gaussian filter bank smoothing and b) Ridge Detection

In this ridge based detection method we detect the completeness of the page of the document. The parameter of the ridge detection method is measured first and then the filters are generated. After that the set of filters are applied to each pixel of the document and the best output response is selected for the smooth images.

ii. Page Frame Detection

In this method the left and right borders are calculated initially using a straight-line approximation algorithm. The estimation of upper and lower border of the selected document is done by taking the top most and bottom most ridges within the borders [11]. Initially the page frame detects the non-text element. After dragging and extending the upper and lower borders are selected and final page frame is identified.

A. *Experimental Setup*

This whole project has been done using python language, as python is one of the most commonly used for AI-ML. our project is partially based on ML. Our project is basically an .exe file.

Our project is so simple that everyone can use it. For setting up our project in a device, user doesn't need to install any other software or they don't need any other hardware

equipment. Our project can work in offline mode, so internet is also not required to install or work on our project.

B. Preprocessing

- After importing an image, we check the position of the words (i.e. page no., block no., line no., word no., height, width, etc. as shown in Figure 2)

level	page_num	block_num	par_num	line_num	word_num	left	top	width	height	conf	text
1	1	0	0	0	0	0	1340	462	-1		
2	1	1	0	0	0	19	11	1296	434	-1	
3	1	1	1	0	0	19	11	1296	434	-1	
4	1	1	1	1	0	19	11	1296	51	-1	
5	1	1	1	1	1	19	11	160	30	96.344353	Mahatma
5	1	1	1	1	2	191	14	117	29	96.239525	Gandhi
5	1	1	1	1	3	319	24	63	20	96.574783	was
5	1	1	1	1	4	392	25	69	27	96.574783	very
5	1	1	1	1	5	473	17	108	29	96.901390	honest
5	1	1	1	1	6	591	19	75	28	96.936356	from
5	1	1	1	1	7	679	20	41	28	95.823509	his
5	1	1	1	1	8	731	21	167	29	95.823509	childhood.
5	1	1	1	1	9	912	24	40	27	95.743416	He
5	1	1	1	1	10	964	31	91	21	95.743416	never
5	1	1	1	1	11	1066	25	137	28	96.763054	resorted
5	1	1	1	1	12	1214	29	33	25	96.763054	to
5	1	1	1	1	13	1257	34	58	28	96.880539	any
4	1	1	1	2	0	22	60	1289	47	-1	
5	1	1	1	2	1	22	60	95	30	96.180084	unfair
5	1	1	1	2	2	127	64	109	27	96.662064	action.
5	1	1	1	2	3	248	65	65	27	96.973038	One
5	1	1	1	2	4	324	64	61	36	96.997414	day,
5	1	1	1	2	5	395	66	87	28	96.938080	while
5	1	1	1	2	6	499	67	112	28	93.666214	Gandhi
5	1	1	1	2	7	622	76	59	20	93.666214	was
5	1	1	1	2	8	692	70	24	26	94.132088	in
5	1	1	1	2	9	727	69	109	34	94.132088	school,
5	1	1	1	2	10	847	79	36	19	96.872765	an
5	1	1	1	2	11	896	73	150	34	96.092880	Inspector
5	1	1	1	2	12	1055	73	33	28	96.092880	of
5	1	1	1	2	13	1095	73	120	30	95.820251	schools
5	1	1	1	2	14	1225	83	86	21	95.820251	came
4	1	1	1	3	0	22	112	1286	47	-1	
5	1	1	1	3	1	22	118	40	21	94.565918	on
5	1	1	1	3	2	68	108	13	47	94.565918	a
5	1	1	1	3	3	105	112	74	28	96.798729	visit.

Figure 2: Text positioning method

V. Result Analysis

After installing the executable file, a folder will be installed. Inside that folder, there will be an executable file. After running the application, user interface will be opened. Where user can extract text from an image. In the UI, users can upload the by clicking on **"Browse the image".** [We can import, any type of images, like- .png, .jpg, .jpeg, etc.]

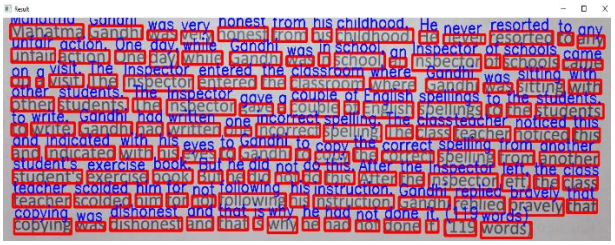

Figure 3: Text identification process

And, after successful importation of an image, a message-box will popup which will show **"picture imported successfully"**.
After that, to extract the text, simple we'll have to click on **"Convert to Text"**. And, after extracting the text, again a message-box will pop-up showing that the **"text extracted successfully"**. The process is shown in Figure 4.

- After that, the identified words are marked with boxes (colour red), and the identified text is shown (colour blue) over the boxes in Figure 3.

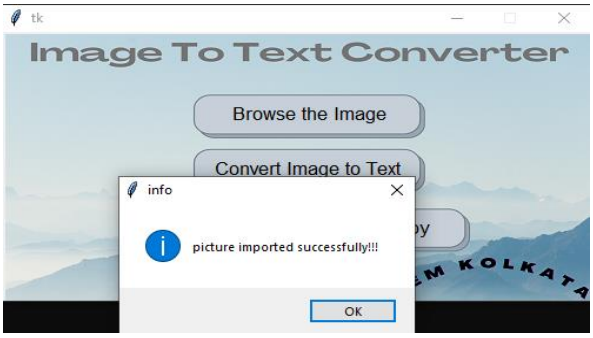

Figure 4: Image to Text conversion

After that, to extract the text, simple we'll have to click on **"Convert to Text"**. And, after extracting the text, again a message-box will pop-up showing that the **"text extracted successfully"**. The process is shown in Figure 5.

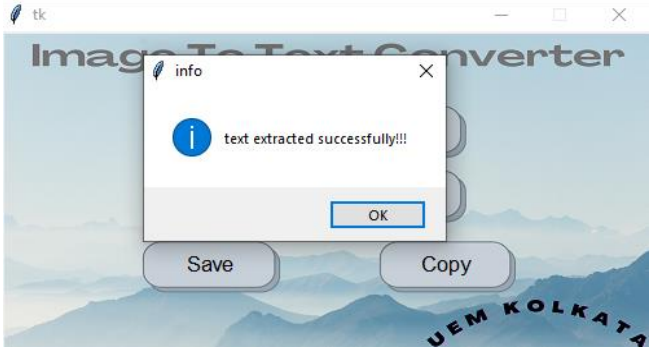

Figure 5: Text extraction process

After clicking on ok, the output text will be shown on the command prompt. The process is shown in Figure 6.

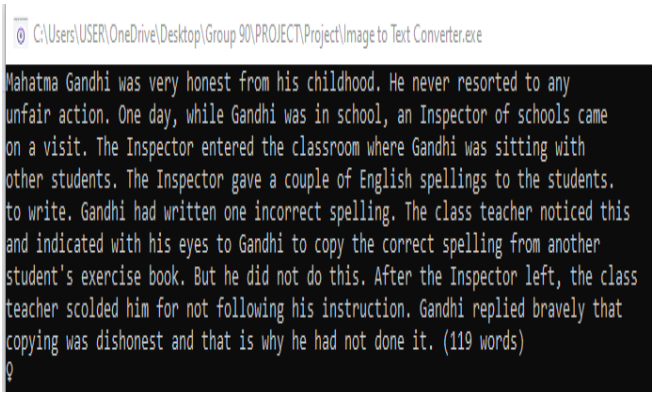

Figure 6: Output at command prompt

There are two extra options named **"Save"** (to save the text as **Document.txt** at the **"**Download**"** section) and, **"Copy"** (to copy the whole text).

VI. Conclusion

The purpose of the project is to get our hands deep into the AI field and understanding various capabilities of the same. The project helps to read the textual data present in a page or pic of a page which we may need to extract to present the data for some special purpose required by the user.

This project was chosen as it intrigued us as a learner and we have been using this technique in another popular app know as Google Lens thus we tried to make a lens of our own. This application can also be made multilingual so that it can give output in any language, i.e. if the input is in English and the user need the output in another language, then it will be easy for them. In future this work can be extend for multilingual approach. The challenge here is to work with Tesseract with Multiple Languages.

References

[1] Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, *2*(1), 72-75.

[2] Santosh, K. C., Antani, S., Guru, D. S., & Dey, N. (Eds.). (2019). Medical Imaging: Artificial Intelligence, Image Recognition, and Machine Learning Techniques. CRC Press.

[3] Zhang, X., & Dahu, W. (2019). Application of artificial intelligence algorithms in image processing. *Journal of Visual Communication and Image Representation*, *61*, 42-49.

[4] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

[5] Nicolaou, A., & Gatos, B. (2009, July). Handwritten text line segmentation by shredding text into its lines. In 2009 10th international conference on document analysis and recognition (pp. 626-630). IEEE

[6] Chakraborty, P., & Mallik, A. (2013). An open source tesseract based tool for extracting text from images with application in braille translation for the visually impaired. *International Journal of Computer Applications*, *68*(16).

[7] Hermida, X. F., Rodriguez, A. C., & Rodriguez, F. M. (1996). A Braille OCR for Blind People. *Proceedings of ICSPAT-96. Boston (USA)*.

[8] Sanchez, A., Suarez, P. D., Mello, C. A. B., Oliveira, A. L. I., & Alves, V. M. O. (2008, November). Text line segmentation in images of handwritten historical documents. In *2008 First Workshops on Image Processing Theory, Tools and Applications* (pp. 1-6). IEEE.

[9] Smith, R. (2007, September). An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 629-633). IEEE.

[10] Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2008). Text line detection in handwritten documents. *Pattern recognition*, *41*(12), 3758-3772.

[11] Shafait, F., Beusekom, J. V., Keysers, D., & Breuel, T. M. (2007, June). Page frame detection for marginal noise removal from scanned documents. In Scandinavian Conference on Image Analysis (pp. 651-660). Springer, Berlin, Heidelberg.

[12] Özgen, A. C., Fasounaki, M., & Ekenel, H. K. (2018, May). Text detection in natural and computer-generated images. In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.