Grenze International Journal of Engineering and Technology, Jan Issue

Automated Text Extraction from Images using Optical Character Recognition

Arya Karambelkar[1], Param Mamania[2], Niti Shah[3] and Dr. Nandana Prabhu[4]
[1-4]K. J. Somaiya College of Engineering/Information Technology, Mumbai, India
Email: arya.karambelkar@somaiya.edu, param.m@somaiya.edu, niti.ms@somaiya.edu, nandanaprabhu@somaiya.edu

***Abstract*—In the modern day, practically everything is mechanized, and data is saved and shared digitally. In other circumstances, though, the data won't be digitized, and it could be necessary to separate the text from it to keep it digitally. The technique of extracting text using optical character recognition has been entirely transformed by the newest technologies, such as text recognition software. So, the goal of the software is to automatically retrieve text from images. This requires an optical character recognition system that incorporates multiple algorithms. Tesseract is currently the most accurate optical character recognition engine developed by HP Labs and currently owned by Google. This paper talks about extraction of text from images using text localization, segmentation, and binarization techniques.**

***Index Terms*— Optical Character Recognition, TEXTractor.**

I. Introduction

Text in photographs is automatically recognized by optical character recognition technology. With the use of various Optical Character Recognition (OCR) technologies, this text may be easily extracted from scanned photos. It works with images that contain text. The tool's output is based on the input image type. Although perfect precision is not feasible, it is preferable to have something than nothing. Text extraction may be particularly challenging at times because of the various font sizes, styles, icons, and dark backgrounds. OCR programs will produce better results when working with high resolution documents. Only a few of the many OCR tools now on the market are open source and free.

Various organizations can use these tools to extract and store useful information from the image. Users can use it to save time and effort while typing. It will reduce the burden of document analysis and understanding. In addition to this, it will also reduce human labour and revolutionize the system by automating the OCR method for text extraction. Many other domain-specific OCR applications, such as receipt, invoice, check, and legal billing document OCR, have been created using OCR engines.

Automatic number plate recognition, passport recognition and information extraction, data entry for documents, automatic key information extraction from insurance documents, promptly creating textual versions of printed documents, extracting contact information from business cards into a contact list, and making e-images of printed documents searchable are all possible uses for them.

II. Literary Review

Text detection and recognition is a well-known issue that has been studied and continually improved in response

Grenze ID: 01.GIJET.9.1.522_2

to the evolving difficulties in the photos and videos on the internet. In recent years, several techniques for text extraction from pictures and videos have been proposed.

Neru Saxena et al. carried out a detailed and comparative analytical review of existing techniques on Text Detection and Recognition from Printed Images (TDE-PI) for English text [1]. They depicted how these techniques recognize the textual entities in terms of character, word and line levels. Researchers have laid a planned process of extracting and recognizing the data under processing.

The Computer Science Department from Calicut University published a paper in which they worked on the classification and Recognition Techniques that are used for handwritten document images [2]. The accurate recognition is directly dependent on the nature of the material to be read and by its quality. They have used the word recognition distances for improving the word matching accuracy. They discovered that edge-based and morphological approaches may efficiently locate and extract text from photos. When compared to morphological and edge-based techniques, the remaining methods, region and texture-based techniques, perform poorly.

Marcin Namysl et al. presented a segmentation-free OCR system that combines deep learning methods, synthetic training data generation, and data augmentation techniques [3]. Using massive text corpora and more than 2000 typefaces, they created synthetic training data. The architecture of their system is universal, and can be used to recognize printed, handwritten or scene text. Alpha compositing with background textures which is a unique data augmentation method is presented and assessed in terms of how it affects the overall robustness of recognition.

X. Vu et al. resolved a task that was organized as a multi-tasking model on a dataset containing 2,436 Vietnamese receipts [4]. They were given the task of creating a model that can identify the textual information of the four needed pieces of information on receipts and forecast the quality of the receipt based on readable information. They predicted this by calculating the root-mean-square-error (RMSE) and Character Error Rate (CER).

Francisca Nwokoma et al. explored problems related to camera-based OCR and other factors that resulted in its computational complexity and suggested some methods for extraction of texts from images [5]. Every camera-based OCR system has significant difficulties in the initial phase of picture capture. A document will be skew, slanted, and out of alignment when the document picture is inadequately captured. Text detection is difficult in case of documents with more than one-part, complicated backgrounds, badly captured pictures, multiple fonts, or other irregularities compared to regular, standard font, plain printed text documents, with which the traditional desk OCR is familiar.

J. Memon et al. systematically extracted and analyzed research publications on six widely spoken languages [6]. They found that certain approaches work better with one script than another; for instance, multilayer perceptron classifiers performed better with Bangla and Devanagari numerals but only averagely with others. The discrepancy may have resulted from the way a particular approach represents various character types and the dataset's quality. Additionally, it was noted that Convolutional Neural Networks (CNN) are being used more often by academics to recognize handwritten and mechanically printed letters. This is because architectures based on CNN are highly suited for image-based recognition applications.

III. METHODOLOGY

A. Obtaining the dataset

The type of data used is qualitative, i.e., descriptive, and conceptual findings collected by observations and interviews. A variety of images have been gathered covering all exceptions to test the accuracy of text extraction. There are primarily three types of image extensions present in the data - jpeg, jpg, and png. This was done to find out whether the text extraction works accurately on all image types. All these images were combined to form a final dataset that was used to develop test cases. Some examples of images that were used are black text on white background and white text on black background. This will test whether the application can extract text when the background is not white. Images having multiple font styles in the same paragraph are also used. Moreover, images comprised of handwritten text are also present in the dataset. In this way, the dataset accurately captures all plausible exceptions in the dataset of 20 images. In addition to this, there are images having different sizes and dimensions. This is essential to ensure versatility of the text extractor in the future. The aim of the exercise is to check the accuracy of the OCR application.

B. Building the application

NumPy is a prominent Python module for scientific computing [7] [8]. In addition to array objects (used to represent vectors, matrices, images, and much more), NumPy also has linear algebra functions. For activities like aligning, warping, modelling variations, classifying, grouping, and other actions with pictures, the array object

makes it possible to perform essential operations like matrix multiplication, transposition, equation system solution, vector multiplication, and normalization.
To develop this application, Tesseract is essential and thus, Tesseract was installed [9]. The images need to be converted to arrays. Vectors, matrices, and pictures may all be represented as multi-dimensional arrays in NumPy. Like a list (or list of lists), an array can only include entries of the same type.
Then, the image is converted to grayscale. This is because converting the image to grayscale makes computations simpler and eliminates redundant steps. A collection of Python bindings called OpenCV-Python was created to address issues with computer vision [10]. A function is used to change the color space of an image. OpenCV offers more than 150 different color-space conversion techniques.
To represent a PIL image, the Image module is utilized next. This module just supplies a class with the same name [11]. The module also offers a variety of factory operations, including tools for generating new pictures and loading images from files. An image memory is created from an object exporting the array interface.
Now, the image object is given as an input to the Pytesseract module to read text from an image [12]. The picture text is changed into a Python string which can then be used as desired. Then, the text is displayed on the screen.

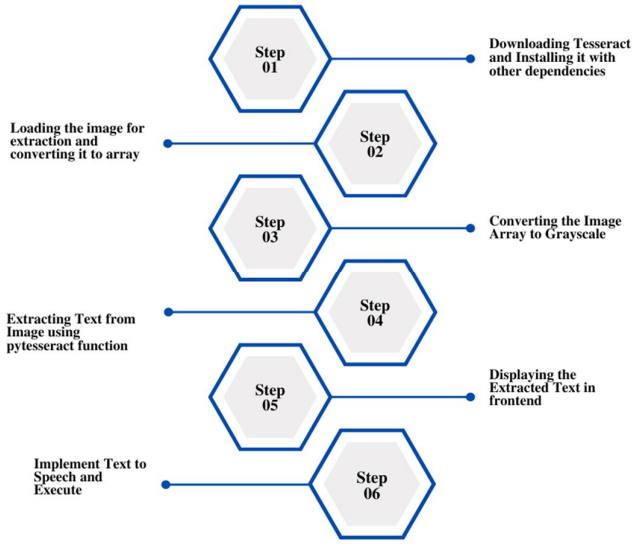

Figure 1. Flowchart of the application

In addition to this, text to speech functionality has also been implemented in the application which will enable people to listen to the text as well. This is done by using the pyttsx3, which is a text-to-speech conversion library in Python [13]. The speech feature is especially useful when someone wants to multitask. For instance, the speech feature can be started, and the person can listen to the text while performing other activities. Moreover, this is also beneficial for the visually impaired people.

C. Working

The application - TEXTractor - has a captivating front-end with information about the benefits of the Text Extraction application as well as a brief history of OCR. In addition to this, there is also a compare and contrast between other OCR tools and our application. The entire website is responsive, multi-page and carefully developed aiming to optimize the user experience.
The text extraction section of the application has a browse button from where the user can select the image. The image can be uploaded in any format or any size. Once the image is selected and extracted, the image will be displayed on the right as a reference and the extracted text will be visible on the left side. There is also a speech button that will read out the extracted text for the ease of the user. All the features are intuitive and easy to use.

IV. Results and Inference

To analyze the results and accuracy of the application, the outputs of each test case are noted down in table format. Two tables are used, one for character accuracy and other for word accuracy.

A. Character Accuracy

Character accuracy is calculated by comparing the text from the image with the extracted text.
In Table 1, the total number of characters in the image is counted and then compared with the correctly identified characters in the extracted text. Some of the images show 100% accuracy whereas, some of the images undergo some extraction errors as seen in the table.

TABLE I. CHARACTER ACCURACY

Image	Total Number of Characters	Correctly Identified Characters	Accuracy
Image1.jpeg	61	61	100%
Image2.jpg	386	386	100%
Image3.png	174	174	100%
Image4.jpeg	79	76	96.20%
Image5.jpeg	15	15	100%
Image6.jpeg	14	14	100%
Image7.jpeg	51	50	98.10%
Image8.png	55	53	96.30%
Image9.jpeg	957	957	100%
Image10.jpeg	491	491	100%
Image11.png	111	111	100%
Image12.jpg	48	19	39.58%
Image13.jpg	248	248	100%
Image14.jpg	94	93	98.90%
Image15.jpg	73	41	56.16%
Image16.jpeg	12	3	25%
Image17.jpg	282	282	100%
Image18.jpg	196	196	100%
Image19.jpg	575	575	100%
Image20.jpg	253	253	100%
Average Character Accuracy			90.51%

B. Word Accuracy

Word accuracy is like character accuracy, the only difference being that each word is compared instead of characters.
In the same way, the total number of words in the image is compared with the correctly identified words after extraction and is formulated in Table II.

V. CONCLUSIONS

As we can see from the results that Average Character Accuracy is 90.51% and the Average Word Accuracy is 87.70%. Hence, it can be concluded that TEXTractor is a reliable application. Also, it is visible from the test cases that the application could accurately convert different colored text as well (for instance, Image 5, Rainbow colored text). It also worked perfectly for black background with white text on it and white background with black text on it. It gave almost accurate results for Handwritten text as well. Hence, from our testing we conclude that the application - TEXTractor - accurately converts any text image into copyable and editable text with minimal error. However, a few drawbacks of TEXTractor are that it cannot properly identify characters in presence of a destructive background image, or if the font is curved. A possible solution to this is to use some AI algorithms which find the next closest possible word from the dictionary with the help of NLP. Some things that can be done as the future scope of TEXTractor are, a mobile based camera application like Google Lens, text extraction from videos, automatic number plate recognition, converting handwriting in real-time to control a computer (Pen Computing) [16].

TABLE II. WORD ACCURACY

Image	Total Number of Words	Correctly Identified Words	Accuracy
Image1.jpeg	10	10	100%
Image2.jpg	63	63	100%
Image3.png	27	27	100%
Image4.jpeg	9	6	66.67%
Image5.jpeg	3	3	100%
Image6.jpeg	3	3	100%
Image7.jpeg	10	10	98.10%
Image8.png	11	10	90.90%
Image9.jpeg	152	152	100%
Image10.jpeg	86	86	100%
Image11.png	24	24	100%
Image12.jpg	9	4	44.40%
Image13.jpg	29	29	100%
Image14.jpg	16	15	93.75%
Image15.jpg	12	7	58.33%
Image16.jpeg	2	0	0.00%
Image17.jpg	44	44	100%
Image18.jpg	33	33	100%
Image19.jpg	93	93	100%
Image20.jpg	45	45	100%
Average Word Accuracy			87.70%

REFERENCES

[1] Saxena, Neeru and Parveen, Humera, Text Extraction Systems for Printed Images: A Review (February 10, 2019). International Journal of Advanced Studies of Scientific Research, Vol. 4, No. 2, 2019.

[2] Asst. Prof. Dept. Computer Science and Engineering and Dept. Computer Science and Engineering, , Calicut University, Jyothi, TEXT EXTRACTION FROM IMAGES: A SURVEY, Volume 3, No.2, February 2014.

[3] Namysl, Marcin & Konya, Iuliu. (2019). Efficient, Lexicon-Free OCR using Deep Learning. 295-301. 10.1109/ICDAR.2019.00055.

[4] X. Vu, Q. Bui, N. Nguyen, T. Hai Nguyen and T. Vu, "MC-OCR Challenge: Mobile-Captured Image Document Recognition for Vietnamese Receipts," 2021 RIVF International Conference on Computing and Communication Technologies (RIVF), 2021, pp. 1-6, doi: 10.1109/RIVF51545.2021.9642077.

[5] Nwokoma, Francisca & Odii, Juliet & Ayogu, Ikechukwu & Ogbonna, James. (2021). Camera-based OCR scene text detection issues: A review. World Journal of Advanced Research and Reviews. 12. 484-489. 10.30574/wjarr.2021.12.3.0705.

[6] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," in IEEE Access, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[7] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

[8] http://www.python.org

[9] Patel, Chirag & Patel, Atul & Patel, Dharmendra. (2012). Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International Journal of Computer Applications. 55. 50-56. 10.5120/8794-2784.

[10] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc."

[11] https://pillow.readthedocs.io/en/stable/

[12] Saoji, Saurabh & Singh, Rajat & Eqbal, Ashiq & Vidyapeeth, Bharati. (2021). TEXT RECOGNITION AND DETECTION FROM IMAGES USING PYTESSERACT. Journal of Interdisciplinary Cycle Research. XIII. 1674-1679.

[13] https://pyttsx3.readthedocs.io/en/latest/
[14] Garg, Shreshtha & Kumar, Kapil & Prabhakar, Nikhil & Ratan, Amulya & Trivedi, Aayush. (2018). Optical Character Recognition using Artificial Intelligence. International Journal of Computer Applications. 179. 14-20. 10.5120/ijca2018916390.
[15] Chen, Jinying & Cao, Huaigu & Natarajan, Premkumar. (2015). Integrating natural language processing with image document analysis: what we learned from two real-world applications. International Journal on Document Analysis and Recognition (IJDAR). 18. 10.1007/s10032-015-0247-x.
[16] https://lens.google
[17] C. Patel, A. Patel and D. Patel, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study", International Journal of Computer Applications, vol. 55, no. 10, pp. 0975-8887, October 2012
[18] V. Prasad and Y. Singh, "A study on structural method of feature extraction for Handwritten Character Recognition", Indian Journal of Science and Technology, March 2013
[19]] Karthick, K.B. Ravindrakumar, R. Francis and S. Ilankannan, "Steps Involved in Text Recognition and Recent Research in OCR; A Study", International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 1, May 2019, ISSN 2277-3878
[20] M. N, S. R, Y. G and V. J, "Recognition of Character from Handwritten", 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1417-1419, 2020
[21] S. Muthuselvi and P. Prabhu, "Digital Image Processing Techniques-A Survey", International Journal of Open Information Technologies, vol. 5, no. 11, May 2016
[22] N. Islam, Z. Islam and N. Noor, "A Survey on Optical Character Recognition System", Journal of Information & Communication Technology-JICT, vol. 10, no. 2, December 2016
[23] Satish Kumar, Sunil Kumar and Dr. S. Gopinath, "Text Extraction from Images", International Journal of Advanced Research in Computer Engineering & Technology-2012
[24] Ray Smith, "An Overview of the Tesseract OCR Engine", Proc. of ICDAR 2007, vol. 2, pp. 629-633, 2007