Optical Character Recognition

Orville "El" Anderson
and10393@umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris Minnesota USA

Abstract

This paper looks at Optical Character Recognition(OCR) as a means to improve usability and accessibility of Scanned Documents. The focus is on how to compare OCR implementations, the weaknesses of OCR, and how to can create tests to specifically target those weaknesses.

Keywords: optical character recognition, scanned documents, visual noise, datasets

1 Introduction

Scanned documents are images of physical documents typically made with a scanner or camera. These documents are widely used because they are easy to create and share. Scanned documents have some hidden disadvantages compared to digital documents. Digitized documents have the advantage of being search-able and editable in a way that scanned documents are not.[1]

Digitizing these physical documents is generally time consuming. This is why we have Optical Character Recognition(OCR), a technology specifically made to extract text from images.

2 Background

Pages to Reference[1, 4, 5]

OCR begins with acquiring an image. It will depend on the model, but most OCR will take files in .pdf, .jpg, and .png formats. These files will then go through the three stages of OCR, then will be output in the format of choice. This again will vary depending on the model, but all OCR will return some variation of the text extracted from the page and a separate description of where each of the words were found on the page.

One big factor in the accuracy of OCR is the quality of the initial image. Marks on the physical document, book spines, and low image resolution all add additional complexity to the process.

2.1 Document Layout Analysis

Document layout analysis, DLA, is a general pre-processing step. The purpose of this step is to identify what part of the image is text and what is not. This step addresses some of the complexity introduced when the image was made. This step can crop any dark and can rotate the image as a whole. It exists to remove any visual noise generated when scanning(or handling) the physical document and is sometimes used to remove tables and images from the page. This step frequently includes converting the input image to a binary image, where each pixel is marked as a text or non-text pixel, to reduce computation costs.

[1]Digital documents also have the advantage of increased accessibility and usability for blind and low-vision users.

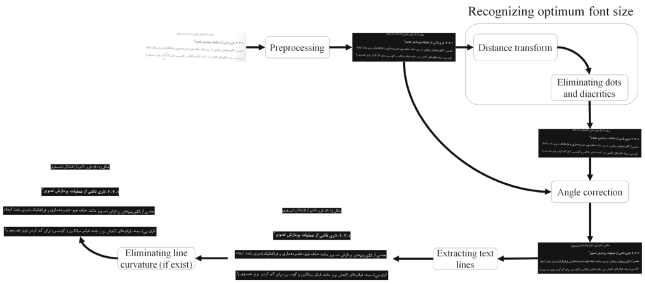

Figure 1: stages of TLD

2.2 Text Line Detection

TLD builds off of how DLA rotates the image, by then rotating the individual lines to create a baseline for each line of text.

This step consists of breaking up blocks of text into lines of text, then into individual words, then letters.

After the individual characters have been identified, we go through and look at each of the boxes we have drawn around the characters and mark which pixels we think are text and which are not.

2.3 Recognition

We then take the mega-cleaned characters and try to identify them. There are many ways to do this, but it boils down to matching your letter to a whole bunch of other letters and seeing which one is closest.

One very common method for character recognition is to use a Deep Learning. (woo magic). The notable thing about this method is that it must be intentionally fed examples of text to train.

2.4 Post-Processing

Post-processing is an optional, but very common step for OCR, consisting of spellcheck and formatting. Running spellcheck on the output generally improves accuracy, but will write

over spellings from the source document(misspellings and unconventional spellings included.) The additional formatting step is used to return the document to a human readable form. Some OCR will place the output directly over the input image, some will make webpages with the output and some will –.

2.5 Comparison

The main considerations when judging OCR models is accuracy and speed.[4]

[Speed is measured with a timer]

A popular way to measure the accuracy of OCR output is to run the OCR program on a document where the page content is know. The OCR output is then compared to the known content and is measured by [the formula below], where x is a unit of measurement, like a line, word, or character.

$$\frac{\sum_{i=1}^{\text{num of pages}} \text{number of x correctly indentified}}{\sum_{i=1}^{\text{num of pages}} \text{number of x on the page}}$$

3 Challenges

There are three main categories of things that make scanned documents harder to digitize.

3.1 Layout

[There exists many ways to format a paper.]

3.2 Alphabet

Most OCR are trained on the Latin Alphabet.[2] Latin is a remarkably simple alphabet (in this context)due to its lack of connected letters[3], dots and diacritics[4]. Connected characters, especially, require extra consideration when performing text line detection. Fateh et Al[2] looks at TLD for Arabic text and found that increasing the size of the boxes drawn around each character increased OCR output accuracy(for the model(s) they tested)

These models are not automatically applicable to other alphabets. As this paper[2] says about Arabic, "Finally, character and TLD approaches cannot be universally applied to scripts with different languages. This challenge is particularly pronounced in languages such as Persian and Arabic, where text characters can take the form of connectors or non-connectors. Connector letters are affixed to both pre- and post-letters to form words, and diacritic marks are commonly used in Arabic text—both of which can introduce complexity to TLD techniques." This inherently gives all other languages a bit of a disadvantage. Other alphabets also have nuances that make it harder to directly apply OCR to. A common example is Arabic. (there's dots) [2, 3]

3.3 Visual Noise

There's so many ways to scan a document badly. [3]

4 Results

Results - some specialized data-sets exist [2, 3]

5 Conclusion

Conclusion - While it's ideal to have one golden benchmarking set, that's hard (because of the reasons outlined above), so now we have specialized ones. Acknowledgments

Acknowledgments

Thanks.

References

[1] Ridvy Avyodri, Samuel Lukas, and Hendra Tjahyadi. 2022. Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review. In *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*. 1–6. doi:10.1109/ICTIIA54654.2022.9935961

[2] Amirreza Fateh, Mansoor Fateh, and Vahid Abolghasemi. 2024. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. *Engineering Reports* 6, 9 (2024), e12832. doi:10.1002/eng2.12832

[3] Thomas Hegghammer. 2022. OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment. *Journal of Computational Social Science* 5 (05 2022). doi:10.1007/s42001-021-00149-1

[4] Ravi Raj and Andrzej Kos. 2022. A Comprehensive Study of Optical Character Recognition. In *2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*. 151–154. doi:10.23919/MIXDES55591.2022.9837974

[5] Chhanam Thorat, Aishwarya Bhat, Padmaja Sawant, Isha Bartakke, and Swati Shirsath. 2022. A Detailed Review on Text Extraction Using Optical Character Recognition. In *ICT Analysis and Applications*, Simon Fong, Nilanjan Dey, and Amit Joshi (Eds.). Springer Nature Singapore, Singapore, 719–728.

[2] why?

[3] an English equivalent would be cursive fonts

[4] such as accent marks