# Challenges of Optical Character Recognition

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

## Abstract

Optical Character Recognition (OCR) is technology used to extract text from images. OCR has three main weaknesses when applied to scanned documents stemming from page layouts, the alphabet used, and visual noise. By intentionally expanding the documents we use to train modern OCR models, we can increase the accuracy of this technology. This paper looks at the Tesseract, Amazon Textract, and Google Document AI models, as applied to English and Arabic documents.

***Keywords:*** optical character recognition, scanned documents, layout, languages, visual noise, datasets

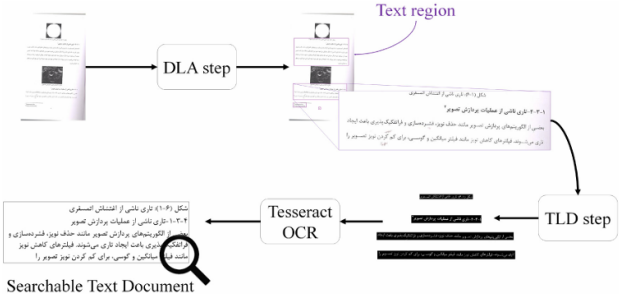


Figure 1: stages of OCR

## 1 Introduction

Over time, American Institutions, have accumulated a ginormous amount of scanned documents. In April of 2024, the Department of Justice, updated the Americans with Disabilities Act to include access to digital content such as scanned documents. Among other things, all scanned documents made publicly accessible by state and local governments must now be usable by a screen reader. When a document is scanned, it becomes an image and looses all information about the text present. The first step to make a scanned document screen-readable is to recognize the text lost when the document was scanned. This process can be done manually, but that isn't well suited for large numbers of documents. Instead we look to Optical Character Recognition (OCR), a technology made to extract text from images.

the first step is to consider if the page is necessary, the second is to try to find the source document

This paper looks at how OCR works, specifically for scanned documents, and looks at some of the specific weaknesses in applying OCR in the context of Public Universities.

## 2 Background

The process of OCR starts with an existing scanned document. These documents can come in many file types, some common ones are .tiff, .pdf, and .jpg. This file is then input to an OCR model, such as Tesseract, Amazon's Textract, or Google's Document AI, where the model will go through the three stages of OCR. The model will then return the text identified in the document as plain text, sometimes with meta-information. [1]

Pages to Reference[1, 4]

### 2.1 Document Layout Analysis

The first step in OCR is Document Layout Analysis (DLA), and is a general pre-processing step. The purpose is to identify what part of the image is text and what is not. This effectively draws a box around each paragraph and table on the page. This step frequently outputs the result as a binary image, where each pixel is marked as a text or non-text pixel, to reduce computation costs.

An important step of DLA is preserving the reading order of the document. Without this step, OCR models are effectively limited to simple single-column text inputs.

### 2.2 Text Line Detection

Text Line Detection (TLD) is the second step in OCR. TLD takes the blocks of text from the previous step and further breaks them down, into lines, words, and then individual characters. The output of this step is each identified character in its own defined box of pixels.

A common technique used in this step includes rotating the individual lines of text to create a baseline, so counter any rotated text. This extra action improves the accuracy of identified characters. Two methods employed by Fateh et al[2] used font size and correcting line rotation, to improve overall accuracy.

[1]Modern OCR models can return use this information to make differnt output formats, but they all use the text and meta-information to do that.

### 2.3 Recognition

The final step in the OCR process is referred to as Classification or Recognition. This step takes the boxes of individual characters from the last step and tries to identify the character inside of them.

One common technique to identify an unknown character uses a process known as matrix matching. In this process, the unknown character is compared to an existing collection of known characters, and the character with the most similarity is chosen. There are a lot of variations in classification techniques, but the key thing to note here is that all methods use some sort of reference material as a basis to classify characters. The output of this step is inherently limited to characters the model has been trained on.

## 3 Challenges

There are three main categories of things that decrease accuracy in the OCR process. Each of these challenges tie back to the key concept that OCR models work best when applied to what they were made to recognize.

### 3.1 Layout

Documents come in many different layouts. Things like images, figures, and number of columns add a layer of complexity to documents. In the Document Layout Analysis step and when the model outputs the final result, to be accurate, the model must have some method to understand the reading order. A paper formatted with two columns, such as this, is meant to be read left column, then right column. Unless otherwise instructed, an OCR model will take the first line from each column and treat them as one line.

There exists OCR specifically trained to handle documents like forms. For example a model can be trained to specifically digitize job applications for a specific company, or one specific tax form. By making a specialized model, there is an increased accuracy when working with documents of that specific layout, but a decrease in accuracy for other layouts.

One related challenge to OCR accuracy is curved lines of text. The DLA and TLD stages of OCR[2] use rectangles to section off portions of text and do not automatically rotates lines/words/characters to be [horizontal?]. Identifying a character when it is rotated is less likely to be accurate then if the character was horizontal. [2]

### 3.2 Visual Noise

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. Marks on the page can both obscure letters, and can also be recognized as letters.

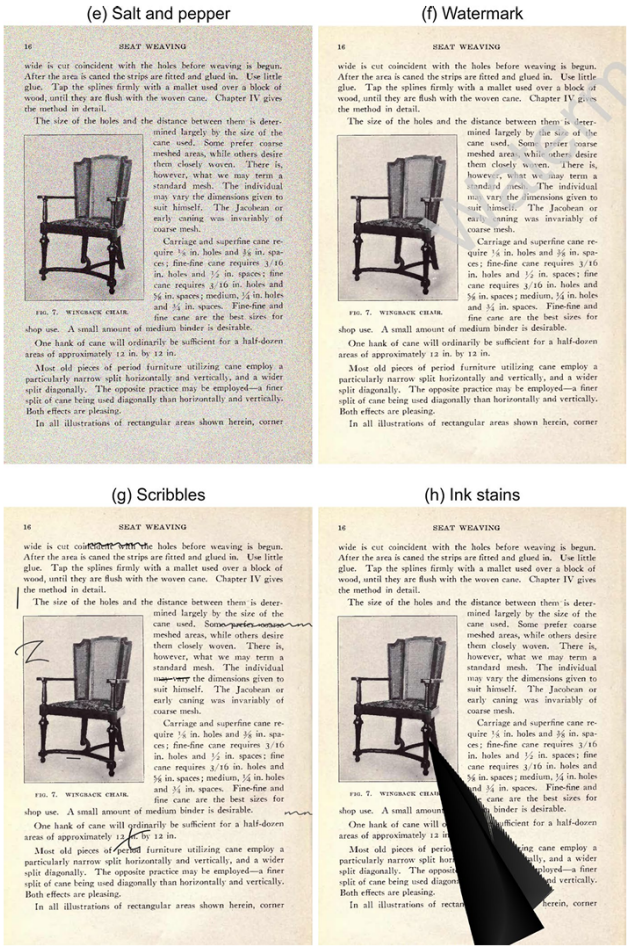


Figure 2: Examples of visual noise from Hegghamer:2022, Salt and Pepper, Watermark, Scribbles, and Ink Stains

### 3.3 Alphabet

The most commonly used writing systems, by users, are the Latin alphabet, Chinese characters, then the Arabic alphabet.[5] [3] The majority of OCR models are trained to recognize characters from the Latin alphabet. As mentioned in section 2.4 Comparison, OCR models are limited in what they can recognize, to the characters they were trained on. The ability to recognize a Latin character, does not automatically extend to characters from other alphabets. While in the context of American universities, most but not all documents are in English. Some of the features discussed in this section are also applicable to English. Improving recognition of connected characters can help with handwritten documents and recognizing various fonts. This weakness in OCR models is most easily seen in non-Latin language documents, but can

[2] don't you dare use another acronym in this paragraph

[3] Arabic is actually an abjad, not a "true alphabet" because of how it treats vowels.


Figure 3: Example of Arabic text, The heading of the Latin Alphabet Wikipedia page

also be seen when using these models on documents with a variety of fonts, or documents with handwritten text.

The best alphabet to highlight this weakness is Arabic. The Arabic alphabet has two main features, that are not common in the Latin alphabet, which impact the OCR process. The first is the use of connected characters, the second is the use of diacritics.

During the Text Line Detection step, to better account for connected characters, the boxes drawn around each character must be larger. Curved text becomes a larger problem when using a larger box around characters.[2] Diacritics are marks that are added around the character.[4] In written Arabic, diacritics are especially important because there are no vowels. These diacritics can be mistaken for visual noise.

## 4 Results

In an effort to better understand the impact of visual noise and the Arabic alphabet on popular OCR models Thomas Hegghamer performed a bench-marking experiment. Hegghamer made a dataset of English and Arabic documents with artificial visual noise applied and used Tesseract, Amazon Textract, and Google Document AI on them. [He found —]. While this doesn’t directly work to address these challenges, it highlights them and provides resources, the dataset and his noise generator, which can be used to train future models. While not an exhaustive list of possible noise types, they represent several of the most common ones found in historical document scans.”[3] Hegghamer’s ”Noisy OCR Dataset”, consists of 422 original documents with 43 variations of each, for a total of 18,568 documents.

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)

Some OCR models, notably Tesseract, have adapted to use machine learning to recognize text. Machine learning, in this case, works like matrix matching2.3, with the advantage that the existing collection of known characters is updated as the model is used on more documents. This method is better suited to recognize a variety of fonts and alphabets, but is still limited in some capacity by it’s exposure to documents. Tesseract has also removed the Text Line Detection Step, instead of identifying text by individual characters, it uses full lines of text.

[4]Accents are an example of a diacritic.

what impact does going by full lines have?

## 5 Conclusion

To compare and evaluate the accuracy of OCR models, the models must be run on the same collection of documents, a dataset. If the goal of the model is to better handle the challenges identified in this paper, the dataset should include as many edge cases as possible. Due to the nature of layout and visual noise, it is inherently impossible to cover all scenarios. [covering all alphabets is still unrealistic at this point, but it is much more feasible than layout and noise].

Serious consideration to storage constraints need to be had when making these data sets. The Hegghamer dataset, NOD, is about 15BG of data when compressed, and – when uncompressed. Hegghamer recognizes in his paper that there is limited variation in the layouts of the Arabic documents he included.

All that said, I think that ideal data set is worth perusing. Progress towards a goal we may never meet is still progress. By pushing the bounds of what OCR models can do, we can strengthen their current capabilities.

## 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] 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.

[5] Don Vaughan. 2025. The World’s 5 Most Commonly Used Writing Systems. https://www.britannica.com/list/the-worlds-5-most-commonly-used-writing-systems