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 a wide variety of uses, with one common application being to digitize scanned documents. OCR has three main categories of challenges that reduce accuracy when applied to scanned documents, stemming from page layouts, the alphabet used, and visual noise. By intentionally expanding the documents used to train modern OCR models, we can increase the range of capabilities of this technology. This paper looks at some of the ways that OCR models have adapted to address these challenges, and at some examples of datasets which are made to cover some of these document variations.

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

1 Introduction

Over time, American institutions have accumulated a tremendous 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 requirements, 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 record of the content found on the original document. 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) models*, programs made to extract text from images.

there's a little more to it, but generally some software that takes and input, applies all stages, then outputs something

The Background 2 of this paper covers the three main steps in OCR. Challenges 3 looks at how layout, alphabet, and visual noise impact OCR output accuracy. Results 4 section looks at two examples of datasets designed to address these challenges and looks at how the OCR model Tesseract has adapted to address some of these challenges. Conclusion 5 discusses the importance of specialized datasets and their role in increasing accuracy of OCR.

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, where the model will go through the three stages of OCR. Figure 1 shows a document, with images, in Persian, going through each of the stages of OCR. The output of the model is the text identified from the document. This is generally returned as plain text, but can also include meta-information [1].

As pictured in Figure 1, the first stage of OCR is *Document Layout Analysis* (DLA) which breaks down the page into sections of text. The second stage is *Text Line Detection* (TLD), which then further breaks down the sections into individual lines of text, or into individual words. The final stage is Classification and Recognition which identifies the text and outputs it as a search-able text document.

2.1 Document Layout Analysis

DLA 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, which is non-obvious for multi-column documents, documents with table, and other documents with more complex structure. Without this step, OCR models are effectively limited to simple single-column text inputs.

2.2 Text Line Detection

TLD is the second step in the OCR process. 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.

mention the pixel dimensions?

A common technique used in this step includes rotating the individual lines of text to create a baseline. This can be seen in Figure 1, where the text entering the TLD step is at an angle, but the output is rotated so each line is horizontal.

[1] Modern OCR models can use this meta-information and text to construct new output formats.

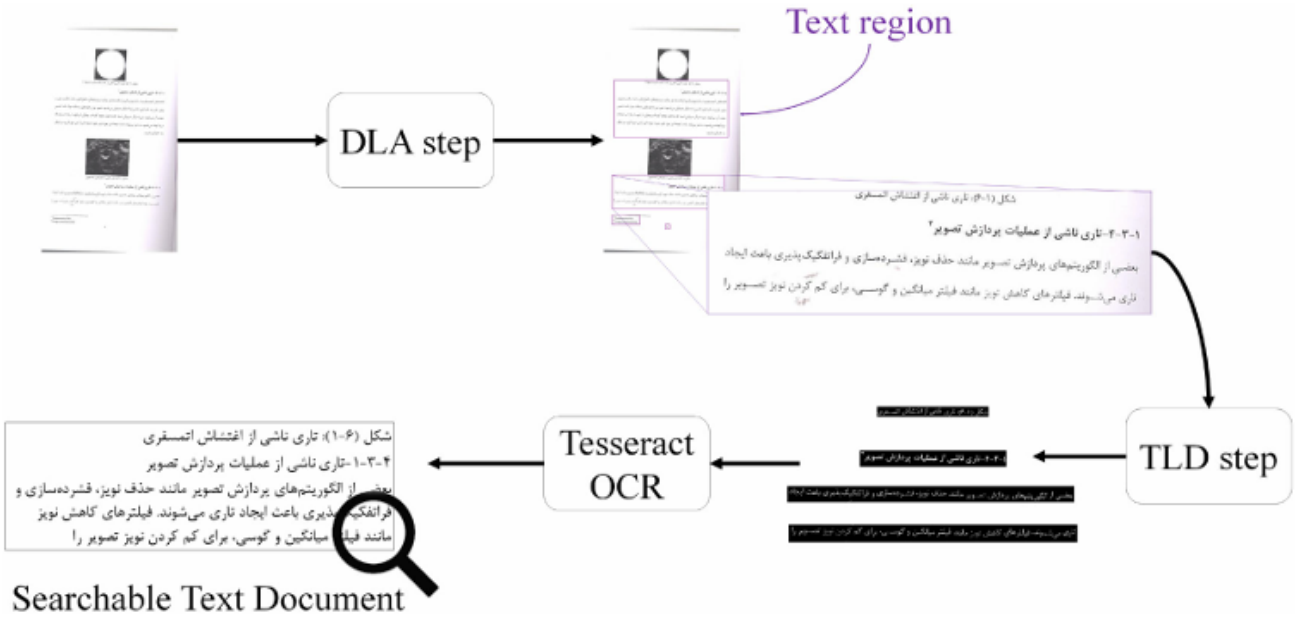

Figure 1: Stages of OCR pictured on a Persian Document[1].

By creating this baseline and somewhat standardizing the characters, accuracy in defining characters is improved [1].

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. [3] Another popular technique, called feature extraction uses measurements like character height, width, and presence of loops to identify a character. There are a lot of variations in classification techniques, but the key thing to note is that all methods use reference material as a basis to classify characters [3]. The output of this recognition step is inherently limited to characters the model has been trained on.

3 Challenges

There are three main categories of issues 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. Images, figures, number of columns, and similar aspects 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.

Some OCR models are intentionally made to only handle one layout, such as a specific tax form, or a job application for a specific company. A specialized OCR model, when applied to the layout it is made for, yields higher accuracy. Layout specific-OCR strategies are not directly applicable to other layouts, and can not be easily combined with other layout-specific models.

they can be combined, but they require some human input to say what document it is.

One related challenge to OCR accuracy is curved lines of text. As seen in Figure 1, the text on the source document was angled. In this figure you can also see that the DLA and TLD stages use rectangles to section off portions of text. By intentionally rotating the lines of text, the boxes are a tighter fit to the text, creating more consistency between the characters and increasing accuracy [1].

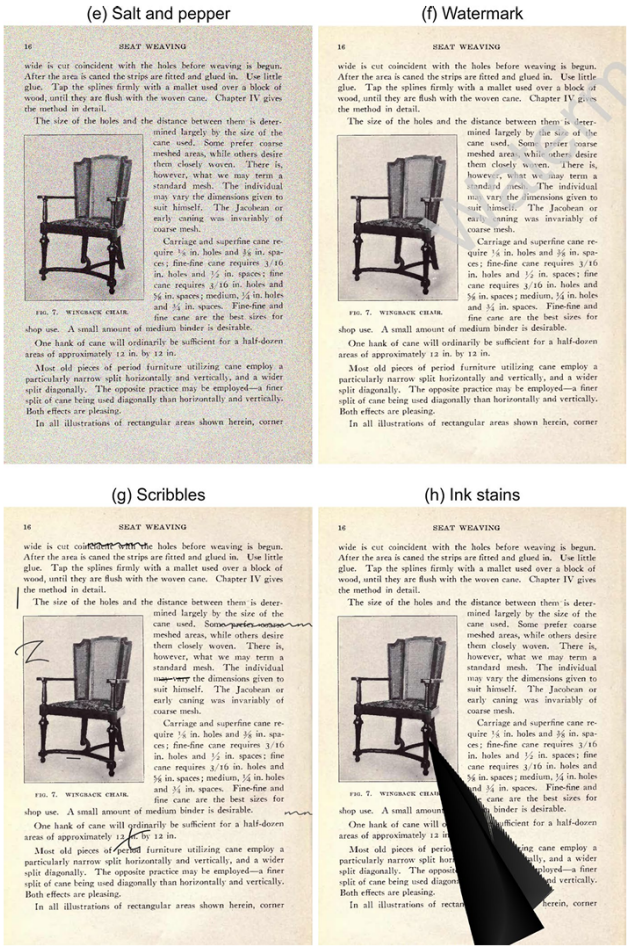

Figure 2: Examples of visual noise from Hegghammer [2]: Salt and Pepper, Watermark, Scribbles, and Ink Stains

3.2 Visual Noise

One big factor in the accuracy of OCR is the quality of the initial image. Marks on the physical document, damaged pages and low image resolution all add additional complexity to the process. There are two main ways that visual noise impacts OCR output, obscuring characters and adding characters.

In figure 2 cases f, g, and h are examples of visual noise obscuring the original text. In these cases, part, or all, of a character is unidentifiable. Some OCR models include an additional post-processing step where the OCR output is checked for spelling errors. This can help address potential missing letters. This post-processing step comes at the cost of potentially overwriting correctly identified text from the source document, such as the author misspelling a word.

Cases e and g from Figure 2 are examples of where the OCR may identify regions of non-text as text. If a section of visual noise is large enough, it can be identified as a character, frequently as punctuation.

أبجدية رومانية

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

3.3 Writing Systems

The most commonly used writing systems, by number of users worldwide, are Latin, Chinese, then Arabic [4]. The majority of OCR models are trained to recognize characters from the Latin alphabet. Most, but not all, documents from American institutions are written in English, a language which uses the Latin alphabet. As mentioned in Recognition 2.3, 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 writing systems. This weakness in OCR models is most easily seen in non-Latin language documents, but can also be seen when using these models on documents with a variety of fonts, or documents with handwritten text.

The best writing system to highlight this weakness in OCR is Arabic. The Arabic alphabet has two main features, that are not common in the Latin alphabet. The first is the use of connected characters, the second is the use of diacritics. Arabic is a cursive language, where a character is connected to the characters before and after it. A diacritic is a small graphic symbol added to a letter. Connected characters often appear in English handwritten documents and diacritics appear in other Latin-based alphabets, such as Spanish and German.

During the TLD step, to better account for connected characters, the boxes drawn around each character must be larger. When the size of boxes around characters increase, curved text becomes a larger problem [1].

In figure 3, diacritics can be seen both above and below the main line of text. Written Arabic does not include vowels, and instead relies on the reader to use context clues to place them. Diacritics are especially important to the Arabic alphabet because they can be used to indicate the necessary vowel when the context is ambiguous [5]. 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 Hegghammer performed a bench-marking experiment. Hegghammer made a dataset of English and Arabic documents artificial visual noise applied and used three popular OCR models on them. “While not an exhaustive list of possible noise types, they represent several of the most common ones

found in historical document scans." Hegghammer found that certain types of visual noise such as blur and salt and pepper specks, reduced output accuracy more than types like ink stains and watermarks. [2] While this doesn't directly provide a way to address these challenges, it highlights them and provides resources, such as the dataset and his noise generator, which can be used to train future models.

Fateh et al [1] look at new methods to improve accuracy of the TLD and DLA steps when applied to Persian text. The Persian script derived from the Arabic script. In this paper, the authors discuss the use of separate datasets to test their proposed TLD and DLA methods. This paper highlights several DLA-specific datasets which utilize newspapers and magazine collections to provide a variety of layouts. When it came to testing their TLD method, they specifically note: "TLD in complex scripts like Persian and Arabic presents unique challenges, and the availability of suitable standard datasets is limited. Unlike English or other widely studied languages, Persian and Arabic require specialized datasets and approaches to tackle text line extraction effectively." In response to the lack of a suitable TLD dataset, the paper introduces a specific Persian dataset.

Some OCR models, notably Tesseract, have adapted to use machine learning to recognize text. Machine learning, in this case, works like matrix matching, discussed in Recognition 2.3, 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 its access to a variety of documents. Tesseract has also removed the Text Line Detection Step, instead of identifying text by individual characters, it uses full lines of text. This reduces errors introduced when segmenting the text and minimizes the steps needed to reconstruct the text to output it.

cite

5 Conclusion

To compare 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 every alphabet is similarly difficult, but to a lesser degree.

Serious consideration to storage constraints need to be had when making these data sets. Hegghammer's "Noisy OCR Dataset" (NOD), consists of 422 original documents with 43 variations of each, for a total of 18,568 documents. NOD, is about 26 GB when compressed and 193 GB uncompressed. Hegghammer recognizes in his paper that there is limited variation in the layouts of the Arabic documents he included.

horrible news, i think this might be more, one noise type for a color version and one for black and white...

All that said, these individual datasets are an important step in progress towards improving OCR accuracy. The overall discussion of weaknesses in current OCR technology guide improvements to the technology. These specialized datasets made to address these challenges, while not all-encompassing, make space for developments in those specialized areas. Similarly, the techniques used to construct these datasets can be used to make more datasets which can cover more of these cases. By pushing the bounds of what OCR models can do, we can strengthen their current capabilities.

Acknowledgments

Thank you.

References

[1] 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

[2] 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

[3] 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.

[4] 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

[5] Wikipedia contributors. 2025. Arabic diacritics — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Arabic_diacritics&oldid=1317677291 [Online; accessed 26-October-2025].