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 writing system 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. By increasing the number of document collections that cover these challenges, we make it easier to develop OCR techniques to target them. This paper looks at document collections for evaluating the extent of these weaknesses and researching methods to address them. These collections have the added advantage that they can be used to train future OCR models.

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

1 Introduction

Physical Documents are used frequently and in a variety of ways. These can be pages of note, tax forms, and recipes. A common way to save and share these documents is to photograph them. This creates a *scanned document*. For many reasons such as research and record keeping, we care about the specific content of those original documents. Scanned documents do not contain information about the contents of the paper. This can be problematic if you want to go back and search or edit the text from the image. One alternative to manually typing up the contents, is to use *Optical Character Recognition (OCR) models*. OCR models are programs and techniques specifically made to identify text from images.

OCR was popularized as [a tool for banks], but has since expanded to work on a large variety of documents. As we have increased the use cases of OCR, we have encountered several aspects of documents that current OCR models are less accurate on. One of the big weaknesses, or challenges, is the language of the document. [This paper primarily looks at two studies of Arabic documents, but that include other elements of document variety]

This paper looks at the Stages of OCR and some of the emerging Techniques for addressing document variation.

Section 2 of this paper, covers the stages and techniques used in OCR. The three challenges related to document variations are the page layout, the visual noise of the image, and the writing system used. Section 3, Challenges, looks at how they specifically impact the proccess, the stages the relate to, and some additional steps to address them. Section 4, goes into the techniques covered in Sections 2 and Section 3, specifically into the datasets that were used to reaserch these areas. Conclusion 5 discusses the importance of specialized datasets and their role in increasing accuracy of OCR.

2 Stages and Techniques

Scanned documents can come in many different file types, such as .tiff, .pdf, and .jpg. OCR models take the file as an input and performs 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 include meta-information, such as the location on the source image the text came from [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 Neural Networks

One technique which can be used in each of the stages of OCR is to use Neural Networks. Neural Networks, in this context, are a way to take an image and to make prediction about the content of the image. [The most popular use is for recognition, so I will discus it there, but will otherwise brush past it.]

A Neural Network takes the pixels of an image and maps them to a series of *nodes*. These nodes represent a possible pattern. As the input pixels are mapped to the nodes, they are adjusted with a *weight*, to assign a value. [assign a value to what?] In the context of images, one possible representation of the input pixels can be a RGB value of the pixel.

[1] Modern OCR models can use this meta-information and text to construct complex output formats like HTML files and can be used in combination with a graphic user interface to display the text output to highlight any potentially incorrectly identified characters.

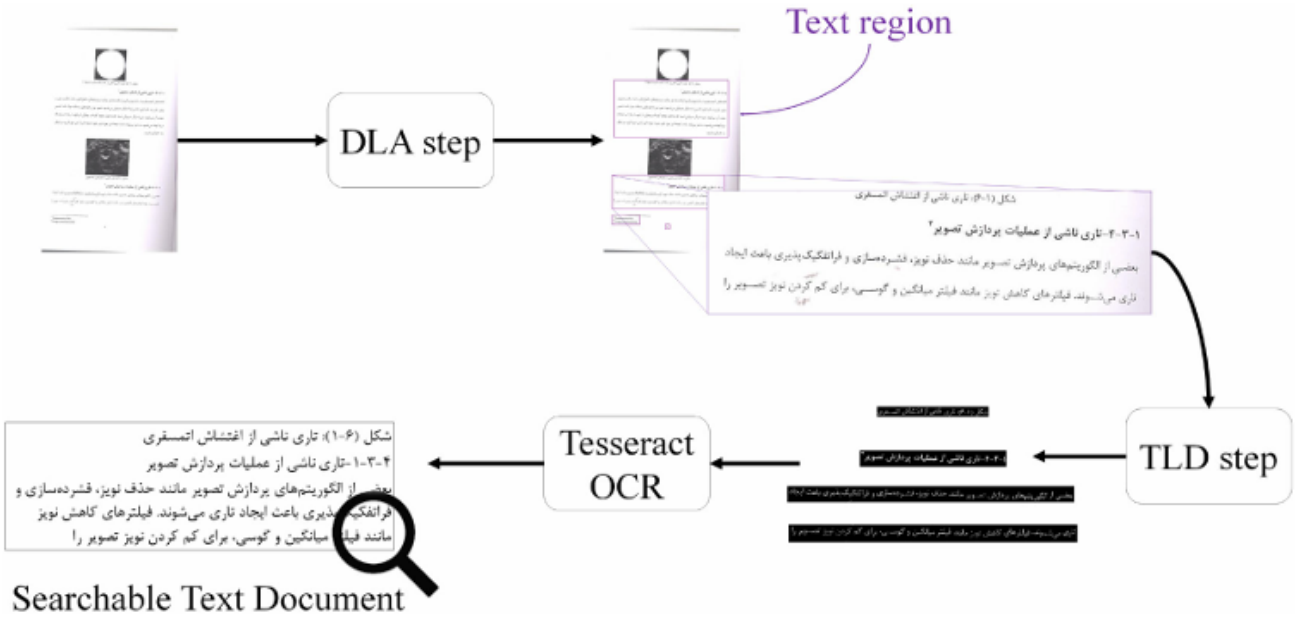

Figure 1: Stages of OCR pictured on a Persian Document[2].

[bad example] Weights are a series of values independent to the input image. At each node, the value of the input is modified with an *activation function*. A Neural Network can contain many layers of nodes. An activation function is used at each layer to introduce complexity and further control the significance of certain patterns.

To determine the weights, Neural networks going through a process called *training*. In this process, the neural Network is applied to a variety of documents with known content, a*dataset*. The output of the OCR stage is judged and the weights are adjusted to minimize error. Some examples of error in the the context of documents, could be part of the background that was marked as text, or could be a misidentified character.

2.2 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 regions of text. A region can be many things, such as an isolated character, full paragraphs, or tables.

An important step of DLA is preserving the reading order of the document, which can be non-obvious when it comes to documents beyond a single column of text. Features like the number of columns of text, and the presence of components like images and tables disrupt a traditional top-down reading order.

Fateh et al [2] put forward a DLA method which used a voting system on the output of four Neural Networks trained to recognize regions of text. Each of the four voting Neural Networks, used coordinates to record where on the original image the text was found. The overall OCR model they used, Tesseract, takes full lines of text and returns the text as text boxes placed over the original image. The need to preserve the document order is hidden when OCR is used on full lines of text and placed in this way. DLA becomes apparent when the model is used on individual characters or when the output is constrained to one column, like a plain text file.

2.3 Text Line Detection

TLD takes the previously identified regions of text and further breaks them down into lines, words, or individual characters. The size of the unit of text is determined by the technique used in the Recognition stage 2.4. Modern techniques such as Neural Networks 2.1, can be used on full lines of text, instead of individual characters. The output of TLD is each identified unit of text in its own defined box of pixels. Traditionally, these boxes have a uniform pre-set size. To best fit the text into the boxes, TLD may rotate, center or scale the text.

It is important for the boxes in question to be large enough to contain the intended character(s), but not to include characters from the neighboring lines or words. This results in a long discussion of what is the proper size box.

By the segmentation process at lines of text, instead of further breaking the text into individual characters, we can prevent introducing errors related to character overlap. One downside of using full lines of text, is that because they cover more of a page, the line has a larger capacity to be curved. Fateh et al [2] propose a TLD technique which uses font size

to rotate and standardize full lines of text. This can be seen in Figure 1, where the text entering the TLD step is at an angle, but the output is rotated, to create a baseline. By creating this baseline, and somewhat standardizing the characters, the accuracy when defining the characters is improved [2]. To develop this technique, they built a special dataset made to include curved lines of text and lines with very little space in-between.

2.4 Recognition

The final step in the OCR process is referred to as Classification, or Recognition. This step takes sections of text, either characters or lines, and tries to identify the contents.

One of the foundational techniques used in character recognition is known by a variety of names, one being Matrix Matching. The technique uses templates of known characters. In this process, an unknown character is compared to all of the templates. The output of Matrix Matching is a list of the templates and how similar they were to the unknown character. The identity of the template with the highest similarity, is chosen as the identity of the unknown character.

An increasingly popular technique for identifying characters, uses Neural Networks. This technique is similar to Matrix Matching, where the input is the unknown character, and the output is the full list of possible characters and a measure of confidence for each option. In Matrix Matching, each mismatched pixel has a uniform impact on the resulting similarity metric. In a Neural Network, each pixel and possible character outcome is given an associated weight. This means that the presence or absence of pixels can have a larger impact on the confidence in a certain character outcome. This makes it so the resulting confidence measure is a percent between 0 and 100.

Neural Networks are popular because the weights add a level of flexibility when it comes to identifying characters. Neural Networks are more accommodating to types of fonts and variations in character rotation and placement. Neural Networks can also include a condition, where if the confidence is low for all of the possible outcome characters, the unknown character will be added to the dictionary of possible outcomes. This feature makes it easier to identify unknown characters, but it is important to know that Neural Networks, like Matrix Matching, are still limited in some capacity by the characters they have been exposed to previously. Neural Networks can be trained to identify a larger number of characters, by intentional exposure.

3 Challenges

There are three main categories of issues that decrease accuracy of OCR for documents. These categories are the layout of the original document, the presence of visual noise, and the writing system used.

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 DLA stage, and when the model outputs the final result, to be accurate, the model must have some method to record and replicate 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 readily combined with other layout-specific models.

3.2 Visual Noise

One big factor in the accuracy of OCR is the quality of the initial image. Marks on the page or noise acquired when the image was captured, add additional complexity to the OCR process. Thomas Hegghamer[3], a historian, performed a bench-marking experiment to better understand how OCR models of the time were impacted by the types of noise found in historical documents. Figure 2 shows the six different types of visual noise he studied. Hegghammer used 422 documents from two existing datasets. Hegghammer kept a color version of each of the document and made a copy in black and white. To each of the versions, he then applied layers of each of the noise types, so each version had additional versions with zero, one, or two layers of noise. In total, each starting document had 43 variations of each, for a total of 18,568 documents.

Hegghammer had two main noise-related findings from his experiment. First, that documents with more noise layers, produced OCR output with higher percentages of incorrect words, than those with less layers of noise. Second, certain types of noise had a larger impact on accuracy. Specifically, noise types that were built into the document had a larger impact. In Figure2, these are Blur and Salt and Pepper.

There are many ways that noise actually effect accuracy. These integrated types impact how the model’s ability to tell the boundaries and lines of characters. Two additional ways that noise can impact the output is by being mistaken for a character, or by covering up characters. In instances like, scribbles and watermark, from Figure 2, the noise is made up of text, but is frequently not an indented part of the output. Because of their placement the characters inside of the watermark may be identified then inserted throughout the output. Noise types like Ink Stain can partially and fully obscure text. In these cases, a human reader is left to fill in the missing text, by guessing or attempting to counteract

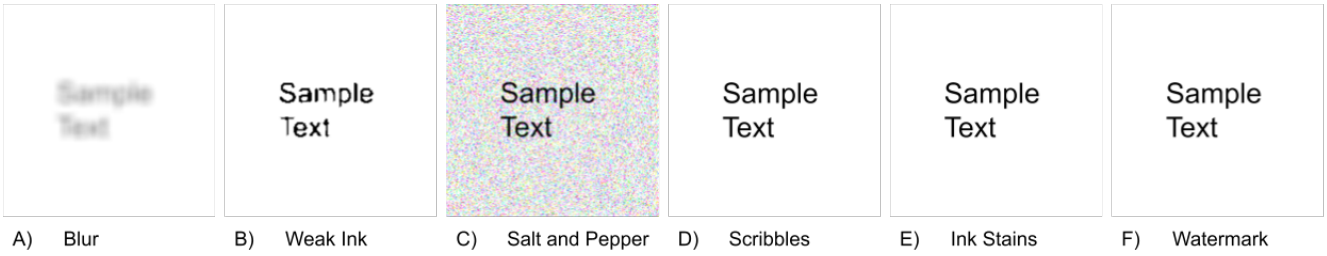

Figure 2: Examples of visual noise: Blur, Weak Ink, Salt and Pepper, Scribbles, Ink Stains, and Watermarks.

the stain. Obscured text in OCR results in missing text and disrupts the reading order.

The errors resulting from noise, can be addressed in two parts of the OCR process: pre-processing and post-processing. The techniques which are used in pre-processing are sometimes included in the DLA stage. This can include rotating the full image, removing any borders, and sharpening or blurring regions of the image [1].

3.3 Writing Systems

The most commonly used writing systems, by number of users worldwide: are Latin, Chinese, and 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 based on the Latin alphabet. As mentioned in Recognition 2.4, OCR models are generally limited in what characters they can recognize, to ones they have been exposed to previously. Additionally, the techniques used to identify Latin characters do not automatically extend to characters from other writing systems. The current limitations in identifying a variety of text types is most easily seen in non-Latin language documents, but also apply for 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 three 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, and the third is the reading order. Arabic is a cursive language, where a character is connected to the characters before and after. 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.

In a study by Fateh et al [2] on improving TLD accuracy for Persian text, they found, to better account for connected characters, the boxes drawn around each character must be larger.

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.

The Arabic writing system is read from right to left, where the Latin writing system is read from left to right. During the DLA stage 2.2 and when the identified characters are assembled into the output, parts of the process must be reversed, to accommodate this.

4 Results

In an effort to better understand the impact of visual noise and the Arabic alphabet on popular OCR models, a historian named Thomas Hegghammer, performed a benchmarking experiment. To perform this experiment Hegghammer crafted a dataset called the “Noisy OCR Dataset”, or NOD. The source documents in NOD come from two existing English and Arabic datasets. Each image was converted to black and white. Hegghamer chose six types of visual noise and applied every combination of noise, up to three types of noise per document, to both the color and black and white versions of the documents. The six types of noise are pictured in figure 2. ‘While not an exhaustive list of possible noise types, they represent several of the most common ones found in historical document scans.”

Hegghammer acknowledges in his paper that the variety of layouts and noise were limited. Hegghammer found that of the types of noise he researched, noise that was built into the page had a larger effect on accuracy than noise that affected targeted areas. Built in noise, referred to blurred documents and documents with multicolored speckling applied. Superimposed noise referred to things like scribbles, ink stains, and watermarks.[3] The contents of NOD are publicly available, along with the noise generator Hegghamer made, and the output from each of the models evaluated, for each of the documents. The documents in NOD, are about 26 GB when compressed and 193 GB uncompressed.

Fateh et al [2] look at new methods to improve accuracy of the TLD and DLA steps when applied to Persian text.

Figure 3: The heading of the Latin Alphabet Wikipedia page, in Arabic.

Persian is 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 total, this paper used three TLD datasets, and five DLA datasets. Of the three TLD datasets, one of them was specifically created for this study. The Official Iranian Newspapers (OIN) dataset, which is made of images of Iranian newspapers, was made to have rotated lines of text, regions with closely spaced lines, and to have a large amount of diacritics.

5 Conclusion

We know that there are existing weak spots within the field of OCR. Through studies like Hegghamer’s bench-marking experiment [3], we know that these weaknesses can be identified and measured. Research like Fateh et al [2] shows us there is more progress to be made in TLD and DLA stage techniques. By understanding the process of character recognition, we know the role of training documents, in increasing the number of characters an OCR model can be used to identify.

All of these findings were made possible because of specialized datasets. Both Hegghamer and the Fateh research team utilized and built off of existing datasets. Because of researchers prior, these studies exist.

While it may be ideal to create one dataset which contains some collection of documents which covers every one of the challenges from this paper perfectly, it is not realistic. Due to the nature of layout and visual noise, it is inherently impossible to cover all scenarios. Covering every writing system is similarly difficult, but to a lesser degree. As discussed in Results 4, Hegghammer’s dataset, which he admitted lacked layout variation and did not cover all noise types, is a total of around 193 GB. This is a large amount of storage for a dataset which covers a mere fraction of the challenges presented. Specialized datasets fill a niche needed for this technology, while working within storage and processing constraints.

By increasing the number of datasets made to cover the current challenges to OCR, and by improving the tools used to make these datasets, we can increase the energy spent to actually address these problem areas.

The challenges identified in this paper are discussed in contexts where they perform notably poorly, but there are also areas where OCR performs well, but still has room for improvement. Handwriting OCR, a subsection of OCR specifically for identifying handwritten text, also has impacted accuracy due to connected characters. The techniques used to identify Arabic can also be applied to cursive English.

A 2022 literature review of OCR research papers found that several implementations and techniques for English scanned documents resulted in accuracy rates above 90%. In some cases, techniques employed resulted in accuracy rates as high as 98 and 99% [1]. While this is impressive, when taken in the context of larger bodies of work, this still leaves a large amount of errors. In cases like digital accessibility, we care a lot about truthful representation of the original document’s text, so continuing to increase overall OCR accuracy is a priority.

While the specialized datasets introduced in this paper may seem unnecessary and irrelevant on first glance, they address the needs of real people and the research used to improve accuracy in those cases has a ripple effect on the larger field.

Acknowledgments

Thank you.

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] 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].