

This work is licensed under a [Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International”](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.



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, it is possible to 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. Some examples are pages of notes, tax forms, and recipes. A common way to save and share these documents is to photograph them. This creates a *scanned document*. The text on the original document may still be significant for users of the scanned document, for purposes such as research and record keeping. Scanned documents do not contain information about the contents of the physical document. While text on a scanned document may be readable visually, the contents are not inherently machine-readable. As a consequence, it is not possible to edit or search within the text of a scanned document. One method to digitize a scanned document is to manually read and re-type the contents. An alternative method is to use an *Optical Character Recognition (OCR) model*. OCR models are programs specifically made to identify text from images.

OCR began as a tool to read messages for Blind people. The technology grew dramatically in popularity because it could be applied in corporate settings to reduce the human labor needed for data entry [4]. As applications for OCR have grown, the technology has been applied to increasingly diverse types of documents. Several aspects of documents play a role in the effectiveness of OCR on them. There is an increased need for OCR that works on a wide variety of

documents. One of the big challenges to OCR accuracy is the writing system of the language of the document. As the third most used writing system, this paper uses examples of Arabic documents to discuss this topic. The other two challenges, page layout and visual noise are covered, but to a lesser degree.

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 layout, noise, and writing system specifically impact OCR. It covers how these challenges relate to the stages of OCR and covers some additional steps which can be used to minimize the effect. Section 4, goes deeper into the techniques covered in Sections 2 and Section 3, specifically into the datasets that were used to research these areas. Section 5, Conclusion, discusses the importance of specialized datasets and their role in increasing accuracy of OCR.

2 Stages and Techniques

The input and output of the OCR process depends on the model used. In general, the input is an image and the output is a text file. Some models, in addition to extracting text from the image will record the location of where the text was found to construct more complex output formats. Tesseract, an open-source model discussed in this paper, accepts a variety of file formats, mainly, PNG, JPEG, and TIFF files. Tesseract currently can format the output as plain text, HTML, as several types of PDFs, and more [5].

The process of OCR can be split into a number of stages, but for the purpose of this paper is made of three: *Document Layout Analysis (DLA)*, *Text Line Detection (TLD)*, and *Recognition*. Two common stages which are not recognized in this list are a pre-processing and post-processing stage. Because these stages are not necessary to perform OCR, they are instead mentioned in Section 3.2 as noise-reduction techniques.

Figure 1 shows a document, in Persian, going through each of the three stages of OCR, where the output of the model is the text identified from the document. The first stage, DLA, breaks the image up into regions of text. For the example document, this is two paragraphs, a page number, and some isolated words. The second stage, TLD, breaks down the sections of text into individual lines of text. Some TLD methods will further break the lines of text down into

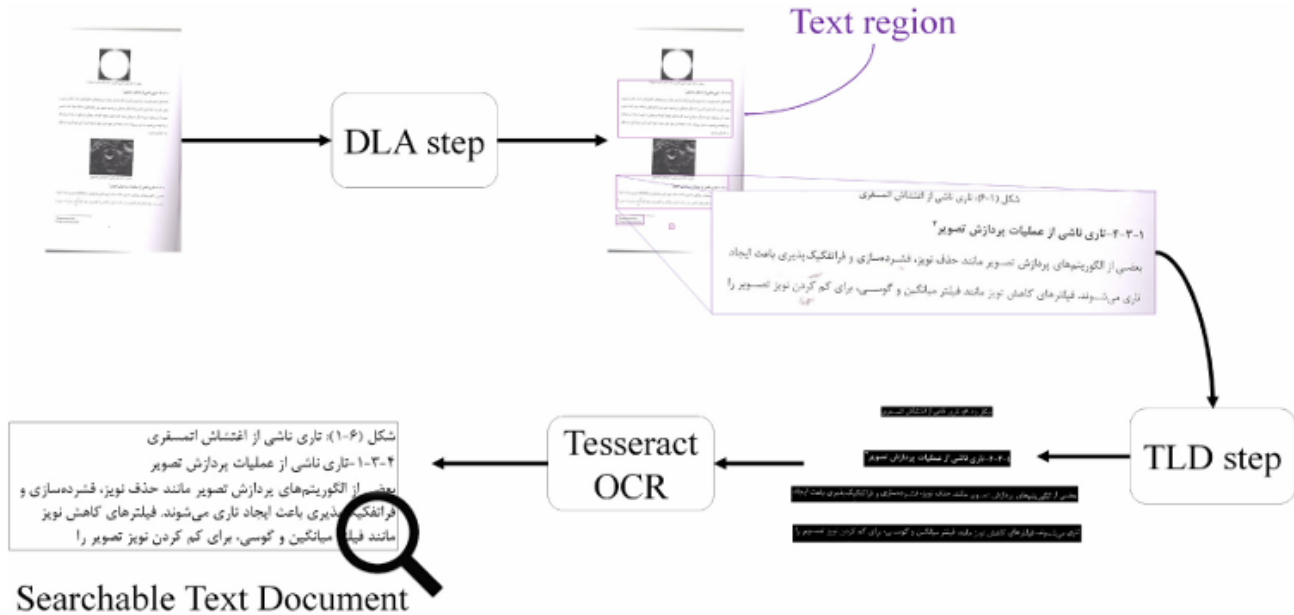


Figure 1: Stages of OCR pictured on a Persian Document[2].

into individual words or characters. The final stage, Recognition, identifies the text in each line and combines the results into one search-able text document.

2.1 Neural Networks

One technique which can be used in each of the three OCR stages, is to use *Neural Networks*. Neural Networks are a subsection of Machine Learning meant for pattern recognition. These networks use large functions with many parameters to make predictions. In the context of OCR and scanned documents, these predictions are often if a region contains text, or are about the value of some text.

To choose the parameters, or *weights*, in the Neural Network, it must go through a training process. This process begins with a large collection of examples, a *dataset*. Datasets can be anything from full documents to individual characters, depending on which stage the Neural Network will be used for. It is important for the training process that there is an existing machine-readable copy of the contents of the examples. The dataset is divided into two or more groups. The Neural Network is exposed to the first group of examples, to get a sense of patterns within the possible outcomes. In this step the Neural Network will set initial values for the weights. After exposure, the network is then exposed to the second group of examples. The network is meant to use the weights chosen in the earlier step to predict the values of the second set. The accuracy of the predictions are then used to adjust the weights in a way that minimizes errors. This process repeats as needed. If the dataset was divided up into more than two groups, the remaining examples can be

used as a final measure for accuracy of the Neural Network's predictions.

2.2 Document Layout Analysis

The purpose of DLA is to identify what part of the input image is text and what is not. To do this, a model can use a variety of techniques to draw rectangular boxes around the text. These boxes can be a variety of sizes and can contain varying amounts of text. The areas not captured by the boxes are ignored for the rest of the stages, to reduce the workload.

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 output of TLD, is each identified unit of text in its own defined box of pixels. To best fit the text into the boxes, TLD may rotate, center or scale the text. Traditionally, these boxes have a uniform pre-set size. 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.

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. By ending 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 to attempt to recognize the text identified in the document. Because OCR was originally developed to convert printed characters to sound, many of the initial recognition techniques are no longer used. When the technology moved from noise to text output, some techniques arose which are still relevant. One such technique is known by many names, one being Matrix Matching. Matrix Matching is important to understand because it is the foundation of many modern recognition techniques and can help explain the common restrictions in recognizing text.

Matrix Matching uses templates of known characters. In this process, a single unknown character is compared to all of the templates. For each template, the number of non-matching pixels between it and the single character are recorded. 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.

Neural Networks are a popular technique in the Recognition stage. The networks work similarly to Matrix Matching, but 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. Because of the training process for Neural Networks, through repeated exposure, it is also possible to train the model to predict previously unknown characters.

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.

One related feature of document text, is line spacing. If lines of text are close together, or overlapping, the boxes drawn around the lines, will overlap. This can also occur if the lines of text are not straight or are at an angle. This results in the offending overlap being taken into consideration when identifying the otherwise unrelated text.

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. In 2022, 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 he found in historical documents. Figure 2 shows the six different types of visual noise he studied. To perform this experiment Hegghammer created a dataset based on 422 documents from two existing datasets. Hegghammer kept a color version of each of the documents 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.

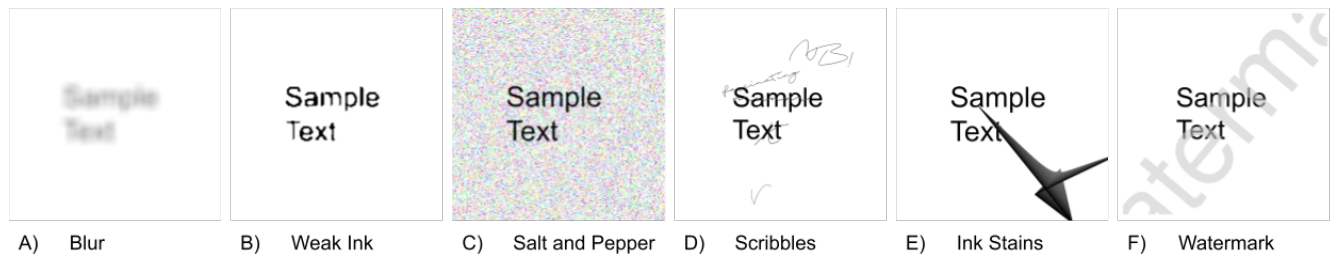


Figure 2: Examples of visual noise: Blur, Weak Ink, Salt and Pepper, Scribbles, Ink Stains, and Watermarks.

Hegghammer came to two conclusions about the impact of noise on the OCR model's accuracy. First, documents with several layers of noise, produced OCR output with higher percentages of incorrect words, than those with less layers of noise. Second, *integrated noise*, noise that was built into the document, had a larger impact on accuracy than *superimposed noise*. In Figure 2, Blur and Salt and Pepper are the two types of integrated noise.

give some numbers, maybe just some Tesseract ones?

Integrated noise impact the ability to distinguish the boundaries and lines of characters. Two additional ways that noise can reduce OCR accuracy, is by being mistaken for characters, 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 the placement of Watermark, the superimposed character can be identified and 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 the stain. Obscured text in OCR can result 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].

mention post-processing more

3.3 Writing Systems

The most commonly used writing systems, by number of users worldwide: are Latin, Chinese, and then Arabic [6]. The majority of OCR models are trained to recognize characters from 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.

Because of its popularity, and how different it is from Latin, Arabic is a relevant example to discuss this weakness in OCR. The Arabic alphabet has three main differences from Latin: the use of connected characters, the use of diacritics, and the reading order.

3.3.1 Connected Characters. Arabic is a cursive language, where a character is connected to the characters before and after. 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.

go back to

Connected characters often appear in handwritten documents, regardless of writing system. Arabic is notable here because it includes it, where printed Latin characters are not cursive.

3.3.2 Diacritics. A diacritic is a small graphic symbol added to a letter. 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 [7]. These diacritics can be mistaken for visual noise.

Diacritics appear in other Latin-based languages such as German and Spanish, but to a lesser degree.

3.3.3 Direction. 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. For writing systems such as [] and [], where they are read top to bottom, the techniques used in OCR must be adjusted further, to change how the regions of text are drawn to begin with.

أبجدية رومانية

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

enter writing systems

4 Datasets

Thomas Hegghammer’s effort to understand the effect of noise on OCR accuracy was limited in the variety of layouts present in the documents and in the types of noise. “While not an exhaustive list of possible noise types, they represent several of the most common ones found in historical document scans.” Even with this limitation, his study gives insight into the scale of impact noise had on the models he tested and gives insight into what noise has a more pronounced impact. Thomas Hegghammer made contents of his dataset publicly available, along with the noise generator he 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. 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 Hegghammer’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 Hegghammer 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 Datasets 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 to my adviser, Elena Machkasova, and to my proofreader, Skatje Myers.

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] Herbert F Schantz. 1982. *The History of OCR, Optical Character Recognition*. Recognition Technologies Users Association.
- [5] tesseract ocr. 2025. tesseract-ocr/tesseract. <https://github.com/tesseract-ocr/tesseract>
- [6] 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>
- [7] 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].