Diverse Documents: Challenging 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 increasing the number of document collections that contain these variations, 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. Additionally, this paper looks at the role these collections can play in training neural-network-based OCR approaches.

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

1 Introduction

Physical documents are used frequently and for a variety of purposes. 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*.

For many reasons such as research and record keeping, the contents of a scanned document may be important. Scanned documents do not contain machine-readable text. While text on a scanned document may be readable visually, no record of the contents is made when the image is taken. 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 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. This paper focuses on an open-source OCR model called *Tesseract*. This model is commonly used in OCR research because it is designed so the techniques used can be easily changed.

OCR began as a tool to read messages for blind people and was later used to convert messages to Morse code. As the technology developed, the purpose shifted to more commercial applications such as record keeping. The technology grew dramatically in popularity because it could 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. This paper covers the impact of page layout and the presence of visual noise on OCR output, in the context of English documents. A large barrier to using OCR on scanned documents comes from the language of the document. Arabic is the third most used writing system. This paper uses examples of documents in languages, such as Persian, which use the Arabic writing system to discuss this challenge.

Section 2 of this paper covers the three main stages of OCR, along with several techniques used in each of the steps. 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 introduces additional steps which can be used to minimize the challenges. Section 4 goes deeper into the techniques covered in Sections 2 and 3, specifically into the datasets that were used to research these areas. Section 5, Discussion, introduces some ethics regarding datatsets and 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. Tesseract, an open-source model discussed in this paper, accepts a variety of file formats, mainly, PNG, JPEG, and TIFF files. Some models will both extract text from images and record where the text was found. This additional information about the location of the text can be used to construct more complex output formats. Tesseract currently can format the output as plain text, HTML files, several types of PDFs, and more [5].

The process of OCR can be split into a number of stages, but in this paper we discuss the main three: *Document Layout Analysis (DLA)*, *Text Line Detection (TLD)*, and *Recognition.* [1] 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,

[1] Two common stages which are not recognized in this list are a preprocessing 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.

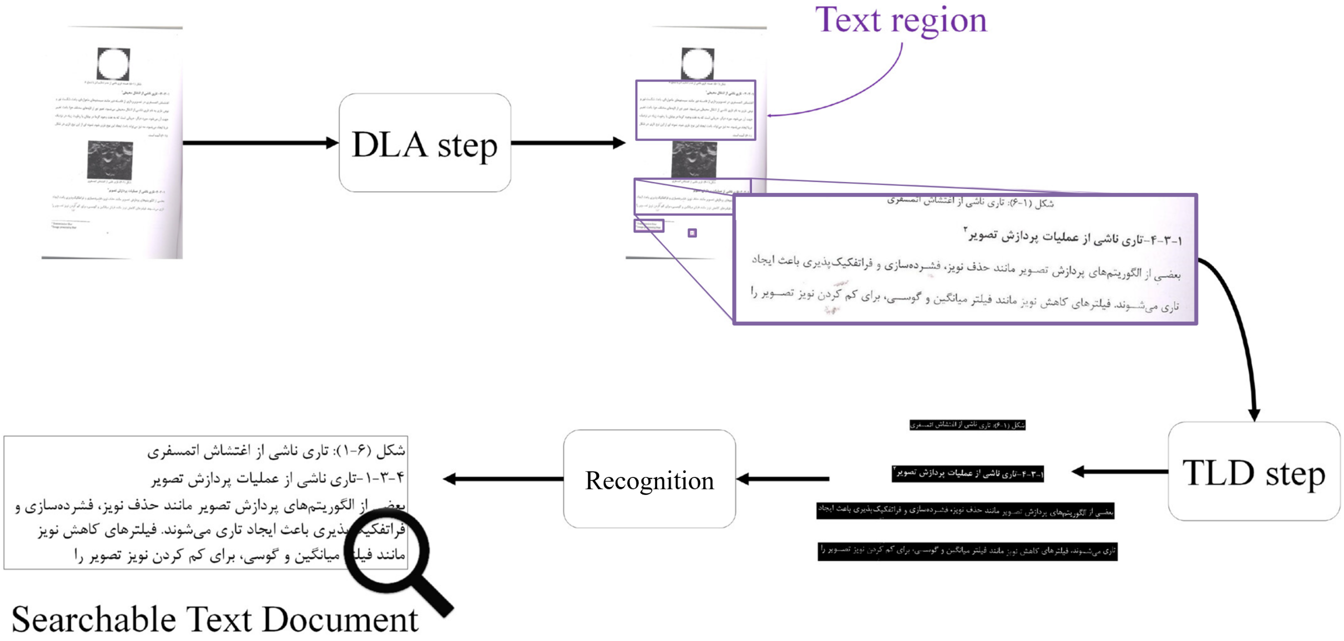

Text region
DLA step
TLD step
Recognition
شکل (۶-۱): تاری ناشی از اغتشاش اتمسفری
۴-۳-۱-تاری ناشی از عملیات پردازش تصویر
بعض از الگوریتم‌های پردازش تصویر مانند حذف نویز، فشرده‌سازی و
فراتفکی ذیری باعث ایجاد تاری می‌شوند. فیلترهای کاهش نویز
مانند فیل میانگین و گوسی، برای کم کردن نویز تصویر را
Searchable Text Document

Figure 1: Stages of OCR pictured on a Persian Document: Document Layout Analysis, Text Line Detection, and Recognition. Modified from Fateh et al. [2].

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 subsections. The final stage, Recognition, identifies the text in each subsection and combines the results into one searchable text document.

2.1 Neural Networks

Neural networks are a method of machine learning meant for pattern recognition. These networks are important to the field of OCR because they can be used in all stages of the process. With the exception of neural networks used for the recognition stage, how these networks are being used is not unique to OCR. They are part of the larger field of computer vision, so they will not be covered in depth in this paper.

The one important thing to know about neural networks, is that they need to go through a training process to make any reasonably accurate prediction. In an abstract way, a neural network is a large mathematical function which uses many parameters to make predictions. To choose the parameters, a neural network must be exposed to a large amount of examples of each possible outcome. In the case of training a neural network for the recognition stage, these examples could be individual characters from a writing system. Collections of these examples are called *datasets*. Part of the training process is to use the proposed parameters to make predictions about examples that the network has not been exposed to previously. The accuracy of the predictions is then used to modify the existing parameters to reduce the overall error. This process of exposure and adjustment are repeated until the accuracy has met some pre-defined threshold, or is no longer changing significantly between rounds.

To ensure the neural network can be applied in a variety of cases, it is important that the dataset it was trained on was diverse. By intentionally including examples of the challenges discussed in this paper in a dataset, a neural network can slowly learn and adapt to these cases. To train neural networks on these challenges, there needs to be a large amount of diverse and publicly-available document datasets.

2.2 Document Layout Analysis

The purpose of DLA is to identify what part of the input document 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 many dimensions and can contain different amounts of text, depending on the OCR model and documents used. The regions of the document which are not put in a text box during this stage are ignored for the following 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] propose a DLA method which uses a voting system on the output of four neural networks trained to recognize regions of text. Each of the four voting neural networks use coordinates to record where on the original image the text was found. This voting method was added to the base version of Tesseract, which inputs and outputs full lines of text in the recognition stage. Output formats of Tesseract, such as HTML and PDFs, do not recombine text from multi-line regions like paragraphs. Instead, the text is left in individual text boxes. The human labor needed to combine text boxes of individual lines of text is less than the labor needed to combine individual characters. By keeping the OCR output in full lines of text, the models need to make fewer decisions about the reading order. This method reduces the human labor needed, which is often the end goal with increasing OCR accuracy, but does not directly address the challenge The need to preserve the document order is partially hidden when OCR is used on full lines of text, like Tesseract does.

2.3 Text Line Detection

The input to the Text Line Detection stage is the previously identified regions of text. The output of TLD is determined by the technique to be used in the Recognition stage. Modern recognition techniques, such as neural networks 2.1, can be used on full lines of text. In those cases, the output of TLD is the lines of text from the provided region. For other recognition techniques, the lines may be broken down further into individual words or 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, the TLD stage may include rotating, centering, or scaling the text.

By using lines instead of words or characters as the final unit of text OCR methods can largely bypass errors introduced from dividing overlapping or connected characters. One downside of using full lines of text is that, as the length of the box increases, it is impacted more by the rotation or bend of the paper. This can be seen in Figure 1, where the text entering the TLD step is at an angle. Fateh et al. [2] propose a TLD technique which uses the dimensions of characters in the document to rotate and standardize full lines of text. The result of this technique can be seen as the output of the TLD step in the figure. To develop this technique, the researchers built a special dataset made to include curved lines of text and lines with very little space in-between. By somewhat standardizing the characters, the researchers were able to correctly identify 96.11% of the regions of text in the dataset they created. The researchers compared their technique to four existing DLA techniques: Layout Parser, SSD, YOLOv3 and Faster R-CNN. The second best method to identify regions of text in this dataset was Faster R-CNN which was 93.67% accurate.

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. One technique which was popularized when OCR moved from sound to text output is known by many names, one being *Matrix Matching*. This technique 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. The templates and unknown character are represented as series of pixels, either with binary values or color codes that represent the shape of the character. The difference between the pixels in the template and the unknown 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 parameters add a level of flexibility when it comes to identifying characters. Neural networks, especially when trained on a diverse dataset, are more accommodating to types of fonts and variations in character rotation and placement. This makes neural networks better suited for OCR applications involving inconsistent characters, such as handwritten documents, than Matrix Matching.

3 Challenges

There are many factors of scanned documents and OCR models which impact the accuracy of the resulting OCR output. These factors can be loosely categorized as 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.

The placement of lines of text, specifically the distance between lines, can be considered a feature of the layout. While the other variations in layout have an impact on the reading order, line spacing impacts the content of text boxes. 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 curve or are at an angle. If a box contains part of a neighboring unit of text, that text will influence the

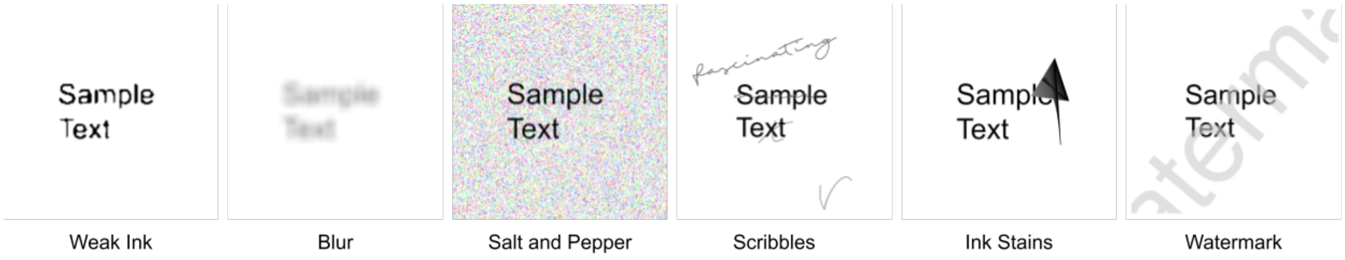


Figure 2: Examples of visual noise generated using code written by Hegghammer [3]: Weak Ink, Blur, Salt and Pepper, Scribbles, Ink Stains, and Watermarks.

results of the recognition stage, resulting in a higher rate of incorrectly identified characters.

Some OCR models are intentionally made to only identify documents in a specific group of layouts, such as receipts or a tax form. These specialized models are given information about the layouts they should expect, which reduces the complexity of DLA and reconstructing the reading order, thus improving the quality of the OCR output. 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. In 2022, Thomas Hegghammer [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. Hegghammer used Tesseract as one of the models he compared. This paper only discusses the results related to the Tesseract output accuracy. 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. One of the datasets was made of old books in English, and the other was a collection of Arabic articles put together for the purpose of OCR research. Documents from both collections began with some degree of noise, but because the Arabic documents were synthetic, they contained less initial noise than the English documents. To each document, Hegghammer added combinations of one and two layers of noise. He then made a greyscale version of each initial document and repeated the proccess.

Hegghammer came to two conclusions about the impact of noise on the OCR model's accuracy. First, documents with several layers of added noise produced OCR output with higher percentages of incorrect words than those with fewer layers of noise. When Hegghammer used the Tesseract model on the English books with no added noise, he got a mean word error rate of 2.4%. This means that in the OCR output, 2.4% of words had at least one misidentified character.[2] For the same documents with one and two layers of noise, the mean word error rates were 23.3% and 41.4% respectively.

Hegghammer's second finding was that *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[3] are the two types of integrated noise. Scribbles, Ink Stains, and Watermarks, are types of superimposed noise. Weak Ink does not neatly fit into either category. Of the English documents with one layer of noise applied, the noise type which resulted in the lowest mean word error rate was Weak Ink with a value of 10.6%. The second smallest value came from the documents with Ink Stains applied, with a mean value of 16.1%. The highest mean word error rate of this subsection of documents was 70.2%, which came from documents with Salt and Pepper applied.

Integrated noise impacts 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 and 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 characters can be identified and inserted throughout the output. Noise types like Ink Stain can partially or fully obscure text. In these cases, a human reader would use context clues to attempt to identify any missing characters. Without this logic in the OCR model, obscured text can result in missing text and can disrupt the reading order.

The errors resulting from noise can be addressed in two parts of the OCR process: pre-processing and post-processing. Avyodri et al. [1] performed a literature review of 29 OCR-related studies to identify techniques used in pre-processing, post-processing, and the three main OCR stages. Three out of the four papers related to pre-processing that they highlight are focused on image rotation. Additional techniques used in this step were to remove any borders, and to sharpen or blur

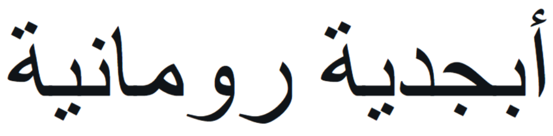
[2] Hegghammer uses the ISRI word accuracy tool, which does not count errors resulting from capitalization, or excess words.

[3] Salt and Pepper is a unique noise type in this context because it adds color, similar to television static, to the image.

regions of the image. Five of the papers reviewed by Avyodri et al. were focused on post-processing. All of the post-processing approaches consisted of removing spelling and grammatical errors. Some techniques used external spelling tools, such as Google’s online spelling suggestions.

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 writing system.[4] As mentioned in Section 2.4, OCR models are generally limited in what characters they can recognize to ones they have been exposed to previously. For Matrix Matching, this limit is the collection of templates; for neural networks, it is the training dataset used. 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 to documents with a variety of fonts, or documents with handwritten text.



أبجدية رومانية

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

Because of its prevalence, and how different it is from Latin, Arabic is a relevant example to discuss this weakness in OCR. The Arabic writing system has three main differences from Latin: the use of printed cursive characters, the use of diacritics, and the reading order. Figure 3 shows an example of Arabic text. Cursive characters are when a character is connected to the characters before or after them. Diacritics are marks around a main line of text. In Figure 3, diacritics appear above and below the main line of text. Arabic is read from right to left, this is visible in Figure 1, but not Figure 3.

3.3.1 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]. To OCR models, diacritics can be mistaken as separate characters and as visual noise. When a diacritic is removed from its associated text, important information about the content is lost. Diacritics appear in other Latin-based languages such as German and Spanish, but to a lesser degree.

3.3.2 Cursive Characters. Cursive characters often appear in handwritten documents, regardless of writing system. Arabic is notable here because typed characters are also cursive, where typed Latin characters are not. In a study by Fateh et al. [2] on improving TLD accuracy for Persian text, they found that, to better account for connected characters, the boxes drawn around each character must be larger. The study specifically put forth a TLD technique which uses box sizes determined by the font size. The study compares the basic implementation of Tesseract and a version of Tesseract which was modified to use their proposed techniques. The researchers tested their TLD technique on three datasets, one they created, an existing Arabic dataset, and one existing Persian dataset. They measured accuracy as the percentage of characters, dots, and diacritics correctly identified. They found that when they added their TLD technique to Tesseract, the total error rates for each of the three datasets went from 6.235% to 3.431%, from 13.38% to 1.52%, and from 6.22% to 3.88%, respectively. This shows that accuracy of this OCR model on Arabic and Persian documents was greatly improved with the addition of this specialized TLD technique.

3.3.3 Direction. The Arabic writing system is read from right to left, whereas 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. Languages such as Chinese and Japanese are read top to bottom. In cases such as these, the techniques used in OCR must be adjusted further, to change how the regions of text are drawn to begin with.

4 Datasets

Thomas Hegghammer’s effort to understand the effect of noise on OCR accuracy was limited by the datasets he used to compare models on, specifically in the variety of layouts present in the documents and in the types of noise. His limited layouts are a result of the source datasets he used. Hegghammer’s old English books contained a moderate variety of layouts, but his Arabic articles were relatively uniform. Hegghammer says, “While not an exhaustive list of possible noise types, they represent several of the most common ones found in historical document scans.” Hegghammer remarks that if not restricted by dataset size and computation costs, the dataset could have been expanded to cover up to 10-20 total noise types, instead of the six he used. The total dataset consists of 422 documents with 43 variations of each for a total of 18,568 documents. When compressed, these files are about 26 GiB and are about 193 GiB uncompressed. The number of documents impacts the time needed to run the

[4] This includes many major languages like English and Spanish. These models generally also recognize the numbers and punctuation included in the American Standard Code for Information Interchange (ASCII) list of printable characters

experiment and introduces a challenge when it comes to storing the dataset. Thomas Hegghammer made the contents of his dataset publicly available, along with the noise generator he made and the output from the models he evaluated.

Fateh et al. [2] look at new methods to improve accuracy of OCR for Persian text, which uses the Arabic writing system. 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." In total, this paper used three TLD datasets and five DLA datasets. Of the three TLD datasets, one was specifically created for this study. That dataset, which is made of images of Iranian newspapers, was made to include rotated lines of text, regions with closely spaced lines, and to have a large amount of diacritics.

5 Discussion

5.1 Accuracy

It is hard to have a discussion about the state of OCR accuracy because there are several ways to measure accuracy. Different accuracy measures are relevant to different applications of OCR. If the OCR output is going to be used for sentiment analysis, overall word accuracy would be a sufficient measurement. If the OCR output is for something like data entry, character accuracy would be a better measure.

In a literature review of OCR techniques by Avyodri et al. [1], the authors identified several studies which resulted in OCR character accuracy rates above 90% for scanned documents. Two notable accuracy rates achieved by these studies are a 99% average character accuracy rate for 20,000 English financial documents and a single instance of 99.8% accuracy for Japanese characters.[5] Neither paper lists the exact formula used to calculate character accuracy, but from this we can still say that by some metrics, the OCR output produced by these methods are very similar to the known contents of the starting documents. While this is impressive, when taken in the context of larger bodies of work, this still leaves a large amount of errors. In projects, where truthful representation of the source documents is a high priority, these errors are often fixed using human labor. For projects with limited resources or large counts of content, any change in the human labor required to digitize the documents can have a significant effect.

5.2 Ethics

As rapid development of neural networks expand the abilities of OCR, we need to consider the ethics of how people get the documents used to train these models. Some models of artificial intelligence are trained on source material which was taken without the permission of the content creators. This can come with legal ramifications such as copyright infringement. There are some differences between historical and modern documents which come from the development of the printing press and the evolution of language. These differences can impact the frequency of certain characters and the presence of certain document layouts. With the exception of neural networks trained for specific layouts and time periods, there is no explicit benefit to using exclusively modern documents in a training dataset.[6] By utilizing public domain content and by generating documents to be used as datasets, researchers can support a more ethical model of training this application of neural networks.

5.3 Conclusion

Specialized datasets help researchers to better understand and develop techniques for these scenarios. These developments address a need for accuracy in practical applications of OCR. Additionally, some of the emerging techniques can be applied to documents outside of their specific context. For example, the techniques designed to recognize cursive characters in Arabic also work for many handwritten documents in other languages such as English.

The studies performed by Hegghammer and Fateh et al. built upon existing datasets. While the datasets they used did not perfectly fit the specific characteristic of documents they were studying, they served as foundation. Accuracy improvements for these real applications of OCR are made possible by pre-existing, publicly available, specialized datasets. By distributing the labor required to build these datasets, and by reducing the resources needed to access and store them, researchers support the development of more specialized datasets.

There is a need for specialized datasets to increase Optical Character Recognition accuracy. For this technology, datasets serve a dual purpose: to develop new techniques and to train neural networks. At this point it is unrealistic to cover every possible document variation, but making these specialized datasets helps to gradually broaden the application of this technology.

Acknowledgments

Thank you to my adviser, Elena Machkasova, to my alumna reviewer, Skatje Myers, and to my peer reviewer, Josie Barber.

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.

[5] These are references number 23 and 26 in the Avyodri et al. review.

[6] One example of a modern layout is a tax form. Because these forms often contain sensitive information, this would be a good area to use synthetic documents.

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].