Improvements to Offline Versus Online OCR

Bachelor's Project Thesis

Martin Cupic, s3736601, m.cupic@student.rug.nl,
Supervisors: Dr C.P. Lawrence & Prof Dr H. Jaeger

Abstract: This project is focused on improving an offline (on-device) and open-source OCR software, called Tesseract, and an online and closed-source OCR software, called Adobe Scan. Firstly, testing was done on benchmark datasets, kleister-nda dataset for high resolution scanned documents and RVL-CDIP dataset for low resolution scanned documents. This paper will focus more on OCR accuracy with relation to low resolution scanned documents since testing showed that high resolution scanned documents achieve an accuracy of 100% with both OCR software. The testing also determined that the OCR detects shapes/icons as characters and the OCR has great difficulty extracting much text from low resolution scanned documents. This is important as it highlights how to increase accuracy and in which areas OCR is especially bad. This was improved by adding a minimum confidence value to remove any icon/shape confusion and a boost to contrast and sharpness was made to increase the quality of low resolution documents. After these two improvements were made the overall accuracy for Tesseract went from 34% to 40% and for Adobe Scan the overall accuracy went from 54% to 58%. There is still a large contrast compared to the state of the art OCR's using deep convolution neural networks which achieves an overall accuracy of 92.21% on this dataset.

1 Introduction

Optical Character Recognition (OCR) is a machine learning field which is specific to recognizing characters inside images such as printed documents, scanned documents, and photos. Even though this technology has been around for a while now, no one has been able to perfect it for all types of text like formulas, analytical, normal text or cursive. Often times the OCR is actually quite inaccurate when it comes to certain types of text but very accurate when it comes to others.

Many businesses are starting to incorporate this technology into their business. A famous example is self-driving cars like Tesla using this technology to read traffic signs. Business and individuals would use this technology to improve processing speed as OCR minimizes the manual process of digitizing work. They could also use it to optimize the workforce by reducing the time spent on redundant work and instead completing it automatically therefore increasing time spent on higher valued tasks, improving productivity and receiving higher customer satisfaction. Lastly, it also reduces costs by eliminating redundant labour which can be done automatically.

OCR is used in many applications across society, such as scanning textbooks, house numbers and car license plates or speed signs for self-driving cars, Forms/Cheque/loan processing and many more. This is why it is important to have an accurate system which works most of the time. This project will help identify how to use the OCR to get the most accurate results.

When developing OCR, developers are looking for a way to educate artificial intelligence on how to read. Firstly, the algorithm does prepossessing to improve the results of the OCR by manipulating the document which it was fed (such as alignment), then the algorithm must be able to see the text which is in the text recognition step, after which it must process the text by double-checking if they are words and finally convert it into a different type of file mainly .txt.

1.1 Research Topic - Online Versus Offline OCR

In this study, I will be testing high resolution scanned documents versus low resolution scanned documents on two OCR software. After which, improvements are made to the OCR software and the OCR accuracy is tested on both software. This study will be attempting to reach the state-of-the-art results of 92.21% accuracy on the low resolution scanned documents dataset (Das et al., 2018).

To understand which is the best image to text software, researchers have to determine what they deem the most important and test out the different software based on these factors. The best Image to text software might be different for different researchers so for this study we will be focusing on software with high accuracy which is reliable and specific for typewritten text. The software decided on was Adobe Scan for the online and closed-source system and Tesseract for the offline and open-source system. These Softwares will be tested against each other to see which performs better on high resolution scanned documents, low resolution scanned documents and the low resolution scanned documents after improvements are made.

2 Background

The origins of OCR can be traced back to telegraphy. Emanuel Goldberg, a scientist, devised a computer that could read texts and turn them into telegraph code. He went much further in the 1920s, developing the first electronic system for document retrieval.

2.1 Types of OCRs

A decade later there are different types of OCRs for different use cases. Below are the 5 types of OCR methods each working for a different purpose (Barve, 2012):

Intelligent Word Recognition - IWR works by recognizing a whole word instead of a single character at a time. It is used for cursive or handwritten text.

Intelligent Character Recognition - ICR works by recognizing a single character at a time. It is used for cursive or handwritten text as well.

Optical Word Recognition - OWR looks at a whole word at a time, it is used for typewritten text.

Optical Character Recognition - OCR looks at a single character at a time, it is also used for typewritten text.

Optical Mark Recognition - OMR works by capturing input data from humans by recognizing patterns or marks on a document.

This paper will be using the most prevalent technology, which is optical character recognition(OCR).

3 Method

As mentioned in the introduction the first step to an OCR is preprocessing which can be done in seven different ways. After this comes text recognition and finally post processing.

3.1 Pre-processing

Pre-processing of the image is done to make it easier for the OCR to get better results. It improves overall accuracy and tries to counteract the negative effects of a low quality image. These are the methods that can be used during the pre-processing step (Springmann, 2015):

De-skew: This technique helps to find the correct alignment of the document so that the characters are not crooked and causing unnecessary errors.

Binarisation: This step is crucial for making text recognition as easy as follow. It separates the text from the background by making the image black and white.

Despeckle: This step is important specifically for images that have a lot of speckles typically caused when scanning or faxing. It cleans up the document by removing any spots on the document.

Zoning: This creates different zones on the document so that it can separate the different attributes on the page. A zone is identifiable as alphanumeric, non-text graphic, or numeric.

Line removal: It removes any lines which run through the text as well as removing empty spaces and lines. This works particularly well if the image is black and white.

Segmentation: The purpose of segmentation is to segment every possible portion of the document into individual characters.

Script recognition: The different fonts must be separated and the correct font must be identified so that the correct font is invoked by the OCR during data capture.

3.2 Text Recognition

Then text recognition is executed on the preprocessed data. This should be done using the following methods:

Matrix matching: This method works by comparing a glyph to a character image through pattern recognition. This works best when simple fonts are used therefore recognition of fancy or artistic fonts is always less accurate.

Feature Extraction: It makes the recognition of characters more efficient and accurate by recognizing all other parts of the document such as figures, intersections, direction, loops and lines.

3.3 Post-processing

After which all the data has been processed, the accuracy can be increased using logical methods. One method is comparing the extracted data to a lexicon which helps increase the accuracy of the extracted data. A Lexicon is a large list of words that is compared to the extracted data and in this way checks whether the extracted words are words or if they should be something else. Processing of the data can potentially be very difficult without the help of a lexicon to steer it in the right direction. Other methods which improve the accuracy are Database Lookups which reads data from a specified database table to check the words and Natural Language Processing (NLP) which processes, understands, analyzes, manipulates, and can potentially generate natural language data from the extracted text.

3.4 Data Pipeline

The following image is a pipeline of the possible functions which can be implemented into an OCR. Often times, not all the functions are used.

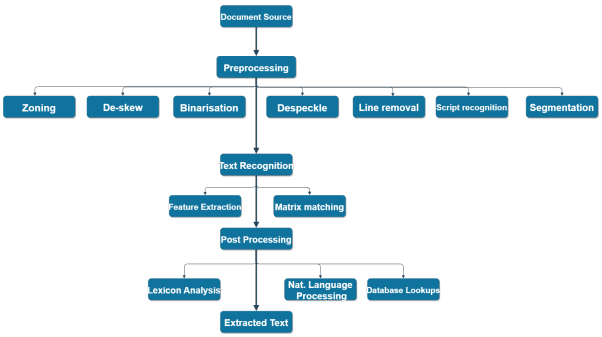


Figure 3.1: Pipeline with all possible OCR methods

3.5 Tesseract

Tesseract is an offline and open-source software which means that it does not need an internet connection to run and the code is freely available for everyone to view. As Tesseract is open-source it

is easy to find out all the methods which it performs when running its OCR. Binarisation is the first step. The result of binarizing a picture looks as follows.

Figure 3.2: Image Before Binarisation Versus Image After Binarisation

Tesseract does not automatically detect the language of an image, but it uses orientation and script detection (OSD) to detect what language it is extracting and most importantly whether that language is in Latin, Cyrillic, Han, Korean or so on. It is able to recognize over 100 languages.

In the pre-processing step, Tesseract uses OpenCV to localize the text in an image. It also uses all the other methods discussed in the prepossessing section. The first step in the recognition process is to attempt to recognize each word in turn. This process is carried out through a two-pass process. After making a satisfactory initial attempt, the word is then passed to an adaptive classifier as training data, which then tries to improve its accuracy. This in turn helps increase the accuracy of words extracted lower in the page.

Tesseract also uses long short term memory (LSTM) which is a form of recurrent neural networks (RNN). When the LSTM fails on a character or character sequence, it can revert to its generic static shape classifier to make the decision. The LSTM which Tesseract uses is therefore actually two OCR classifiers. Feature extraction helps the LSTM to detect where the text is that needs to be extracted by highlighting everywhere that the text can not be. After adding a training tool and increasing the number of data and fonts which Tesseract is trained on, the model's performance improves. However, it's not good enough to work with strange and handwritten text. This is why it's recommended to try out different approaches to improve the model's performance and why I will only be using typewritten text in this study.

This is the pipeline for Tesseract which shows all the methods used by it(underlined in light blue):

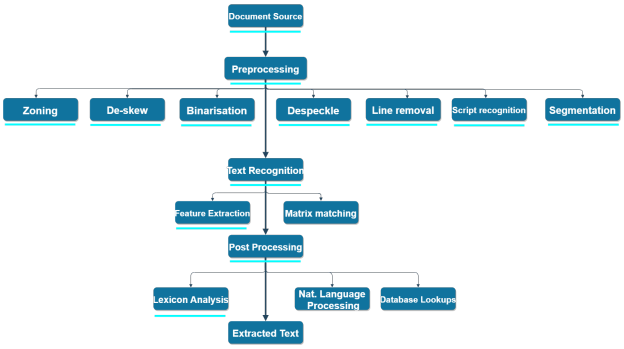


Figure 3.3: Pipeline with methods used by Tesseract (underlined in blue)

3.6 Adobe Scan

Adobe Scan is a closed source and online OCR which means that it can only function with an active internet connection. There is not a lot of information about it supplied by Adobe. However, the document is sent to the cloud where the OCR is performed on it. It should therefore have more computational power than Tesseract which runs solely on the person's computer without an internet connection. The following methods are still some which are used by Adobe Scan:

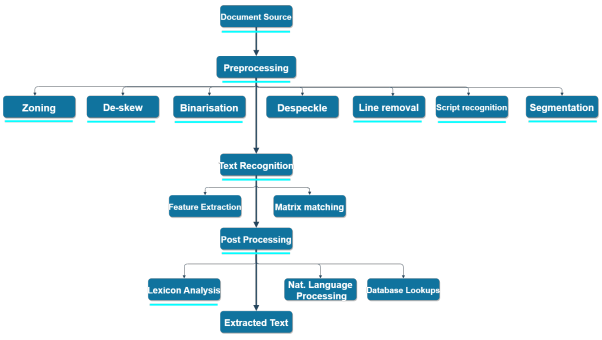


Figure 3.4: Pipeline with methods used by Adobe Scan (underlined in blue)

4 Experiment

On-device and open-source software used in this paper:
Tesseract OCR Engine

Off device (online) and closed-source software used in this paper:
Adobe Scan

Software is chosen based on reliable software which can output raw text from a simple scanned document. The software is also chosen based on the popularity and capability to extract text excluding structured data and success rate.

The testing will be done based on the following three categories:

Category 1 – Low resolution Scanned documents.

Category 2 – High resolution Scanned documents.

Category 3 – Low resolution Scanned documents after improvements.

All of the scanned documents will be provided to the software in .jpeg file types. The output of the text in the images will be given back in a .txt file type. The extracted word accuracy of all categories will be tested on both Tesseract and Adobe to establish which has the highest extraction accuracy and which has the largest impact from the improvements. The accuracy will be tested as the number of words extracted divided by the total number of words in the document. This will be done for 100 scanned documents of each category and the mean accuracy will be taken.

The High resolution scanned documents will be taken from the "kleister-nda" benchmark dataset (Gralinski et al., 2020) which only contains high resolution scanned documents. The low resolution scanned documents will be taken from the "rvl-cdip" benchmark dataset (Harley et al., 2015) which also only includes low resolution scanned documents. The low-resolution scanned documents come in many different types and have markings, stamps, blur, glare from scanning, and specks. The different types of documents tested in the low resolution category are:

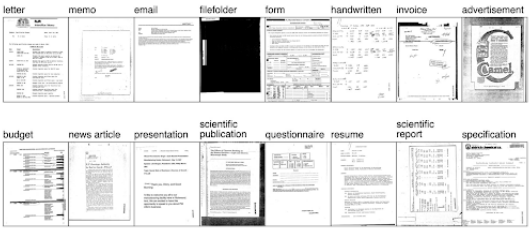


Figure 4.1: Types of low resolution documents.

4.1 Document Resolution

Document resolution is a key factor affecting the OCR's accuracy. However, not all images can have a perfect resolution as some people have poor scanners and OCR's must adapt to this. One way of increasing the resolution is the preprocessing step of the document. However, to increase resolution this preprocessing step must be adapted.

4.2 Hypothesis

Recognition of high resolution scanned documents will likely out perform in terms of accuracy the other two categories. Also, the Tesseract program used will likely underperform state-of-the-art software which works only with online access for all categories. However, it will also likely get very similar levels of accuracy as Adobe Scan for the high-resolution documents category as they are easily readable. Additionally, Improvements made to the Tessaract program in this paper will likely have a beneficial effect in one area such as accuracy but might slow down the processing speed.

5 Testing

The following scanned documents are examples of the 100 scanned documents for each category which were used on Tesseract and Adobe Scan to test their accuracy. In this section, two examples of feeding OCR high and low resolution scanned documents will be shown. For the image in figure 5.1, Tesseract was able to extract most words

but in a more jumbled manner than Adobe Scan. It also added some jargon to the extracted text which makes reading the extracted text difficult. However, it was able to extract all words even though it was in an unorganized manner. Tesseract also recognizes the image next to "LOGO" as a "Q". Adobe Scan does not encounter this problem with the image and manages to accurately extract all words which are selected since you can highlight text and extract it directly from the image in Adobe Scan. It manages to extract all text properly and the only problem is seldomly combining words such as "IndianCreek" in this example. Both software manage to extract all words.

EX-99.(D)(7) 15 d701401dex99d7.htm AMENDMENT TO NON-DISCLOSURE AGREEMENT

Exhibit (d)(7)

EXECUTION VERSION

LOGO

April 6, 2014

PRIVATE AND CONFIDENTIAL

Vocus, Inc.
12051 Indian Creek Court
Beltsville, MD 20705

Re: Amendment to Non-Disclosure Agreement

Gentlemen:

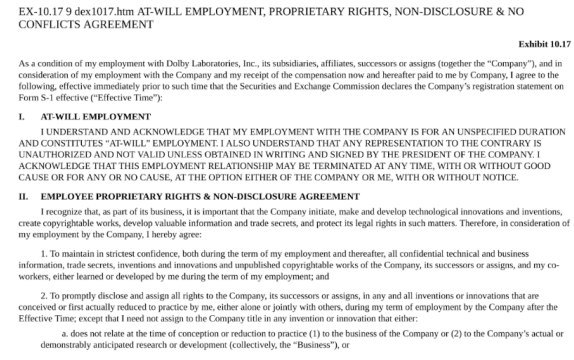
Reference is made to that certain Non-Disclosure Agreement, dated December 20, 2013 (the "Non-Disclosure Agreement"), by and between Vocus, Inc. (the Company") and GTCR LLC ("GTCR"), a copy of which is attached hereto as Exhibit A. Capitalized terms used but not defined herein shall have the meanings given to such terms in the Non-Disclosure Agreement.

GTCR and the Company hereby agree to amend and restate in its entirety the final sentence of Section 3 of the Non-Disclosure Agreement as follows:

"Without limiting the generality of the foregoing and for purposes of clarification, except (a) with the prior written consent of Vocus or (b) for Jefferies Finance LLC, you agree that you shall not enter into any exclusivity agreement or arrangement with respect to a Transaction with any bank or other debt financing source. Notwithstanding anything in this Agreement to the contrary, you and your Representatives shall be permitted at any time prior to the termination of that certain Agreement and Plan of Merger, dated as of April 6, 2014, by and among GTCR Valor Companies, Inc., a Delaware corporation, GTCR Valor Merger Sub, Inc., a Delaware corporation and wholly owned Subsidiary of Parent, and Vocus (the "Merger Agreement") pursuant to Article X thereof (the "Termination Date") to disclose Evaluation Material, hold confidential discussions and negotiations, and disclose Transaction Information (only to the extent relevant to such discussions and negotiations), with those persons identified on Annex 1 to this Agreement (each an "Initial Permitted Disclosure Party") and, with the prior written consent of Vocus, the giving or withholding of which consent shall be in the sole discretion of Vocus, any other person (each an "Additional Permitted Disclosure Party" and, together with each Initial Permitted Disclosure Party, a "Permitted Disclosure Party") and, in each case, their respective Representatives regarding any equity or co-investment participation by such person with you in the Transaction or any transaction to acquire a portion of the equity or assets of Vocus or its subsidiaries; *provided*, that consent of Vocus shall not be unreasonably withheld or delayed in the case of the first three (3) Additional Permitted Disclosure Parties; *provided, further*, that each Permitted Disclosure Party is bound by a confidentiality agreement with you restricting the disclosure and use by such Permitted Disclosure Party and its Representatives of the fact and content of such discussions and negotiations, the Evaluation Material and the

Figure 5.1: High Resolution Scanned Document

For the next scanned document, Tesseract was fed an image of another high resolution scanned document with slightly different text sizes. It manages to achieve 100% accuracy for extracting the text. Adobe Scan also manages to extract all the text with 100% accuracy as well as still combining some words where there should be a space between them.



EX-10.17 9 dex1017.htm AT-WILL EMPLOYMENT, PROPRIETARY RIGHTS, NON-DISCLOSURE & NO CONFLICTS AGREEMENT

Exhibit 10.17

As a condition of my employment with Dolby Laboratories, Inc., its subsidiaries, affiliates, successors or assigns (together the "Company"), and in consideration of my employment with the Company and my receipt of the compensation now and hereafter paid to me by Company, I agree to the following, effective immediately prior to such time that the Securities and Exchange Commission declares the Company's registration statement on Form S-1 effective ("Effective Time"):

I. AT-WILL EMPLOYMENT

I UNDERSTAND AND ACKNOWLEDGE THAT MY EMPLOYMENT WITH THE COMPANY IS FOR AN UNSPECIFIED DURATION AND CONSTITUTES "AT-WILL" EMPLOYMENT. I ALSO UNDERSTAND THAT ANY REPRESENTATION TO THE CONTRARY IS UNAUTHORIZED AND NOT VALID UNLESS OBTAINED IN WRITING AND SIGNED BY THE PRESIDENT OF THE COMPANY. I ACKNOWLEDGE THAT THIS EMPLOYMENT RELATIONSHIP MAY BE TERMINATED AT ANY TIME, WITH OR WITHOUT GOOD CAUSE OR FOR ANY OR NO CAUSE, AT THE OPTION EITHER OF THE COMPANY OR ME, WITH OR WITHOUT NOTICE.

II. EMPLOYEE PROPRIETARY RIGHTS & NON-DISCLOSURE AGREEMENT

I recognize that, as part of its business, it is important that the Company initiate, make and develop technological innovations and inventions, create copyrightable works, develop valuable information and trade secrets, and protect its legal rights in such matters. Therefore, in consideration of my employment by the Company, I hereby agree:

1. To maintain in strictest confidence, both during the term of my employment and thereafter, all confidential technical and business information, trade secrets, inventions and innovations and unpublished copyrightable works of the Company, its successors or assigns, and my co-workers, either learned or developed by me during the term of my employment; and

2. To promptly disclose and assign all rights to the Company, its successors or assigns, in any and all inventions or innovations that are conceived or first actually reduced to practice by me, either alone or jointly with others, during my term of employment by the Company after the Effective Time; except that I need not assign to the Company title in any invention or innovation that either:

a. does not relate at the time of conception or reduction to practice (1) to the business of the Company or (2) to the Company's actual or demonstrably anticipated research or development (collectively, the "Business"), or

Figure 5.2: High Resolution Scanned Document

As can be seen, the high resolution scanned documents are scanned very well as it is almost impossible for the OCR softwares not to extract all the text due to it being extremely clear. For these examples the extracted accuracy was 100%, as well when running the OCR's on many more samples, the accuracy is consistently 100%.

With the following low quality scanned document, Tesseract is unable to extract some of the text and shows clear weakness around the stamps and where there is a substantial blur. However, Adobe Scan manages to extract most of the text while making many small mistakes such as leaving out letters, combining words, reading characters incorrectly and random capitalization but it does manage to extract almost every word while some are extracted incorrectly. For this case, Tesseract manages a word extraction accuracy of 62% while Adobe Scan managed to extract 84% correctly.

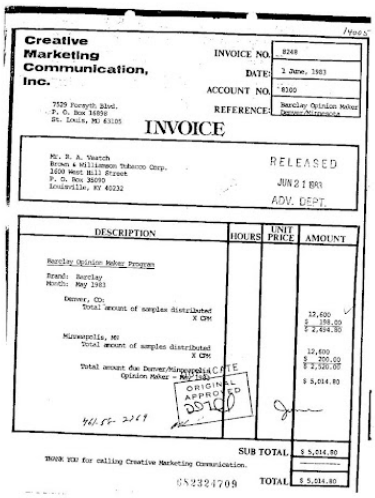


Figure 5.3: Low Resolution Scanned Document

The following scanned document is very challenging for the OCR's. Tesseract has difficulty reading the text and identifying where the text is. It manages to extract 27% of the words correctly. Adobe Scan however manages to substantially outperform Tesseract while still only extracting about half of the words or an accuracy of 53%.

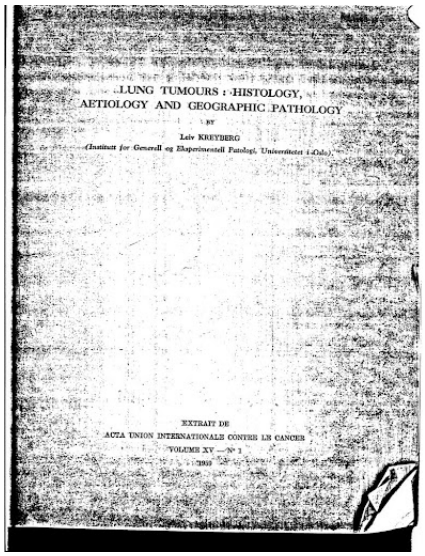


Figure 5.4: Low Resolution Scanned Document

6 Improvements and Results

6.1 Improvements

The Tesseract system clearly has a lower performance as compared to the Adobe Scan. This could be explained due to Adobe Scan being an online system which is able to improve its accuracy through cloud computing. However, Tesseract performs very well with scanned documents that have a high resolution but very poor when it comes to images with low resolution. The difference between Adobe Scan and Tesseract when high quality images are fed into their OCR is very small in what they extract, however, the amount of words extracted is exactly the same, so it is 100% accurate for both cases. When low resolution images are fed into the OCR's then the difference becomes very large, with an accuracy of 33% for Tesseract as compared to an accuracy of 54% for Adobe Scan. This shows that Tesseract is especially poor when dealing with low quality images. Another problem specific to Tesseract was the classification of random objects or shapes as letters. The following images show what can happen when there is a shape that Tesseract assumes to be a letter. Here, Tesseract is confusing the leaf above the apple with an "a":

Figure 6.1: Image from https://pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/

Tesseract sets a confidence value to all of its extracted text. Therefore, an improvement to remove such an error would be to set a minimum confidence value for which Tesseract extracts text. This can be done by adding "–min-conf #" when running Tesseract where the "#" is the minimum confidence percentage. In this case, it is satisfactory to set this number to 50 in order for

Tesseract to not extract the leaf as an "a" since its confidence value is lower than 50% while the other text has a confidence value which is higher than 50%.

Figure 6.2: Image from https://pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/

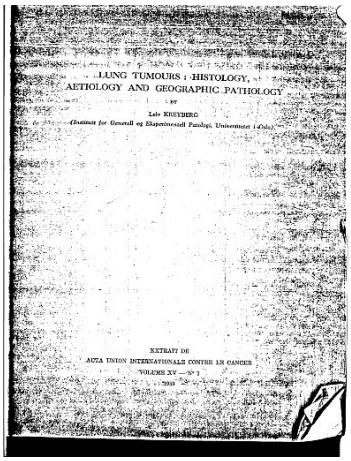

LUNG TUMOURS : HISTOLOGY, AETIOLOGY AND GEOGRAPHIC PATHOLOGY

Figure 6.3: Low resolution image after increasing contrast and sharpness

The next improvement focuses on improving both systems' accuracy when dealing with low quality images. This is far more challenging since there needs to be a perfect balance between removing the noise in the image while not breaking the characters. After testing many alterations to the images, contrast and sharpness were found to have the most significant effect. Therefore, contrast and sharpness are increased for all low resolution images using a python script. Pillow library was used in the script which is a free and open-source library for manipulating images. This improves the results and produces slightly better images for extracting text.

Some low-resolution images could have a higher contrast due to being darker while other images of higher brightness should have a lower contrast. Also if contrast is set too high it can sometimes have a counter effect where it decreases the accuracy due to being more difficult to read. Therefore, a contrast level of 10% is set so that it does not ruin high brightness document results by making them too bright and unreadable. As well as not being too low that it does not help illuminate the text in low brightness images.

A sharpness of 50% is set due to it being the perfect balance of removing the blur around characters but not transforming very unreadable characters such as "a" into different characters such as a "u" for this case. These alterations to the image are shown in figure 6.3.

The text in figure 6.3 is slightly clearer now for an OCR software to read. This change helps Tesseract manage to extract slightly more text. After the improvements were made in figure 6.3, the Tesseract OCR managed to extract exactly one extra word. Its accuracy went from 27% to 31%. Adobe Scan also saw improvement as it extracts slightly more text, does not combine words as much and extracts more words accurately. Adobe Scan accuracy went from 53% to 63% after the improvements.

The OCR pipeline now looks as follows after the improvements were added to the OCR systems:

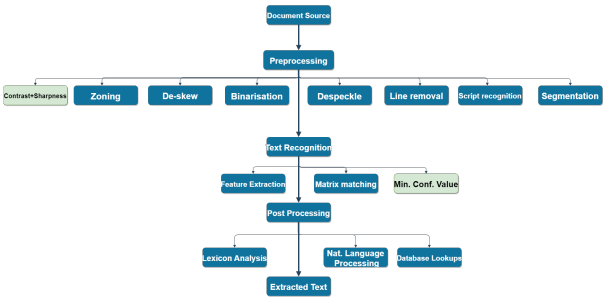


Figure 6.4: Improved Pipeline

6.2 Final Results

The following figures, 6.5 and 6.6, represent the final results after running both OCR software on 100 high resolution scanned documents, 100 low resolution scanned documents and 100 low resolution

scanned documents with the previously mentioned improvements. The overall accuracy for high resolution scanned documents was 100% for both OCR software. Tesseract performed slightly worse for low resolution scanned documents with an overall accuracy of 33% in contrast with Adobe Scan which reached an overall accuracy of 54%. However, the improvements added to the OCRs had a much more significant effect on Tesseract rather than Adobe Scan. Tesseract's overall accuracy for low resolution scanned documents went from 33% to 40% while Adobe Scan improved from 54% to 58%. This is likely due to Adobe Scan already performing better so having less room for improvement as well as already running some sort of image-enhancing in its cloud which Tesseract does not.

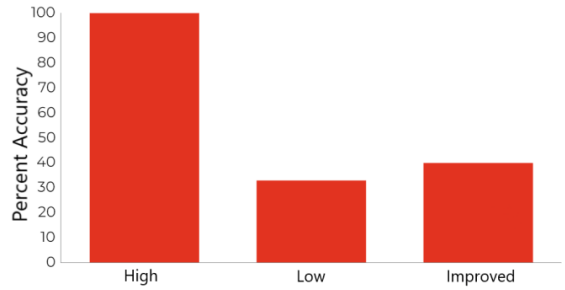


Figure 6.5: Final results for Tesseract

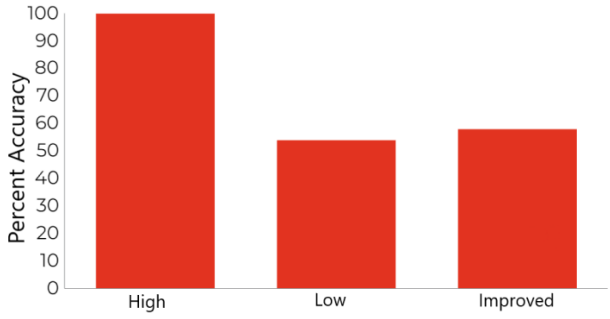


Figure 6.6: Final results for Adobe Scan

7 Discussion

7.1 Conclusion

What was evident is that the OCR systems, Tesseract and Adobe Scan are especially bad at dealing with low resolution images. This makes sense as their confidence value is way lower for low resolution images so sometimes it must guess the characters. Many times the characters also look like other characters due to the blur around them. Therefore wrongfully extracting characters that look similar to other ones(such as "a" and 'u").

Tesseract is especially terrible when dealing with low resolution documents while being equivalent to Adobe Scan when dealing with high resolution documents. The reason Tesseract is worse is due to it not having the computational power and intelligence obtained by Adobe Scan from running on the cloud. My hypothesis that Adobe Scan will outperform was correct as it only makes sense that the system which has access to more resources will perform better. However, the improvements made to Tesseract led to a substantially bigger increase than the improvements made to Adobe Scan. This is likely due to Adobe Scan already running some sort of document enhancer in its OCR and Tesseract having more room to improve.

The improvements to either software were not good enough to let them compete against state-of-the art online OCR software. The state-of-the-art results are 92.21% on the rvl-cdip benchmark dataset (Das et al., 2018). This is substantially higher than the 58% and 40% achieved by the improved software used in this study.

7.2 Reflection

Working with OCRs was an enjoyable process as well as educational as I ran into many problems along the way. Discussing OCRs with others made my experience more motivated and led me to set ambitious goals for this project which could not all be achieved. One which was not achievable was testing formulas on these OCR systems which was a waste of time as they were virtually incapable of extracting anything useful out of the formulas. It seems that it would have been better to start the project off with the simplest approaches and then move to more difficult approaches instead of the other way around. In this way, I could have focused more time on the simple approaches and tried to perfect them by attempting the highest accuracy achievable for the OCR systems used. At the beginning of the project I had a vague idea of OCRs and neural networks but as I was taking the neural networks course at the same time it greatly

evolved my knowledge and made my grasp of the topic much more firm.

7.3 Future Research

For future research, there are many ways to increase accuracy in terms of word extraction. One easy improvement is to just use better OCR software, this way a higher accuracy will be attained from the start. To increase the accuracy through improvements though is a complicated process as trying to increase the accuracy of an OCR through changing its code is quite difficult due to the OCR already being close to perfect if it is widely used. Therefore, The biggest way to improve accuracy is in the preprocessing step of the OCR where the image is manipulated. Here one can attempt to improve the image quality, therefore, increasing the accuracy as the biggest weakness in an OCR system is the resolution of the documents it is being fed. Therefore, the best improvement for future research which I have discovered is to implement super-resolution which uses convolutional neural networks (Dong et al., 2015). In this way, the accuracy of a given image is drastically increased. Example of super-resolution in Figure 7.1:

Figure 7.1: Super-Resolution (Photo: Weizmann Institute of Science)

However, the state-of-the-art OCR for this dataset has a different approach. They have tested many different meta-classifiers for stacked generalization and found that a multi-layer neural network gives the highest accuracy. Fast training of region-based deep convolutional neural networks (DCNN) was explored with several levels of transfer learning. Intra-domain transfer learning was also used in addition to the general inter-domain variant due to the similarity between region-based and holistic input spaces. This makes the feasibility of region-based models and fast convergence possible. In this way, the researchers managed to achieve a state-of-the-art accuracy for the rvl-cdip dataset (Das et al., 2018).

References

(Rosebrock, A. (2021, October 25). Tesseract OCR: Text localization and detection. PyImageSearch. https://pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/).

A. (2021). github - applicaai/kleister-nda [dataset]. https://github.com/applicaai/kleister-nda. (n.d.).

Barve, S. (2012). Artificial neural network based on optical character recognition. *International Journal of Engineering Research & Technology (IJERT)*, *1*(4), 2278–0181.

Cash, G. L., & Hatamian, M. (1987). Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, *39*(3), 291-310. Retrieved from `https://www.sciencedirect.com/science/ar\ticle/pii/S0734189X87801834` doi: https://doi.org/10.1016/S0734-189X(87)80183-4

Chaudhuri, A., Mandaviya, K., Badelia, P., & Ghosh, S. K. (2017). Optical character recognition systems. In *Optical character recognition systems for different languages with soft computing* (pp. 9–41). Cham: Springer International Publishing. Retrieved from `https://doi.org/10.1007/978-3-319-50252-6_2` doi: 10.1007/978-3-319-50252-6_2

Das, A., Roy, S., & Bhattacharya, U. (2018). Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. *CoRR*, *abs/1801.09321*. Retrieved from `http://arxiv.org/abs/1801.09321`

Dong, C., Zhu, X., Deng, Y., Loy, C. C., & Qiao, Y. (2015). Boosting optical character recognition: A super-resolution approach. *ArXiv*, *abs/1506.02211*.

Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013). Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(2), 311-325. doi: 10.1109/TCSVT.2012.2203741

Glasner, D., Bagon, S., & Irani, M. (2009). Super-resolution from a single image. In *Iccv*. Retrieved from `http://www.wisdom.weizmann.ac.il/~vision/\SingleImageSR.html`

Gralinski, F., Stanislawek, T., Wróblewska, A., Lipinski, D., Kaliska, A., Rosalska, P., ... Biecek, P. (2020). Kleister: A novel task for information extraction involving long documents with complex layout. *CoRR*, *abs/2003.02356*. Retrieved from `https://arxiv.org/abs/2003.02356`

Harley, A. W., Ufkes, A., & Derpanis, K. G. (2015). Evaluation of deep convolutional nets for document image classification and retrieval. *CoRR*, *abs/1502.07058*. Retrieved from `http://arxiv.org/abs/1502.07058`

Kae, A., Huang, G., Doersch, C., & Learned-Miller, E. (2010). Improving state-of-the-art ocr through high-precision document-specific modeling. In *2010 ieee computer society conference on computer vision and pattern recognition* (p. 1935-1942). doi: 10.1109/CVPR.2010.5539867

Mori, S., Suen, C., & Yamamoto, K. (1992). Historical review of ocr research and development. *Proceedings of the IEEE*, *80*(7), 1029-1058. doi: 10.1109/5.156468

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Nips workshop on deep learning and unsupervised feature learning 2011.* Retrieved from `http://ufldl.stanford.edu/housenumbers/n\ips2011_housenumbers.pdf`

Plamondon, R., & Srihari, S. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 63-84. doi: 10.1109/34.824821

Rvl-cdip-i dataset. (2019, august 18). [dataset]. https://www.kaggle.com/datasets/nbhativp/first-half-training. (n.d.).

Springmann, U. (2015). *Ocrocis: A high accuracy OCR method to convert early printings into digital text – A Tutorial.* Retrieved from `http://cistern.cis.lmu.de/ocrocis/tutorial.pdf`

Vamvakas, G., Gatos, B., Stamatopoulos, N., & Perantonis, S. (2008). A complete optical character recognition methodology for historical documents. In *2008 the eighth iapr international workshop on document analysis systems* (p. 525-532). doi: 10.1109/DAS.2008.73

A Appendix

https://github.com/MartinCupic1/Bachelors_Project_S3736601