# Extracting Medication Information from Typewritten Philippine Medical Prescriptions Using Optical Character Recognition (OCR) and Named Entity Recognition (NER)

Gwyneth D. Ang, Dianne G.C.S. Chong, James Kevin S. Lin, and Ma. Christine Gendrano*
*Department of Software Technology, College of Computer Studies*
*De La Salle University, Manila, Philippines*
**Corresponding Author: ma.christine.gendrano@dlsu.edu.ph*

**Abstract:** Optical Character Recognition (OCR) technology has been utilized in many studies involving image processing and text extraction. This study utilizes Tesseract, an OCR tool that transforms typewritten medical prescription images into text. Information is extracted through Named Entity Recognition (NER), particularly, annotating text with the labels dosage, drug, duration, form, frequency, route, strength, #tablets, when, how, and remarks. A web application incorporating OCR and information extraction was developed to automatically extract medication information from the medical prescription images. Overall, the OCR tool had a CER score of 0.0455 and WER score of 0.0970, the information extraction model had an F-score of 0.9316, and the system as a whole had an F-score of 0.8816. The application can be used to ease and lessen human error in data entry of medical prescription information. The extracted information can also be used as input for medication management applications.

**Key Words:** optical character recognition (OCR); information extraction; applied computing

## 1. INTRODUCTION

Optical character recognition (OCR) is a field of study in Computer Science that revolves around the extraction of text from images or scanned documents. It enables the digitization of printed text so that it can be edited, searched and stored electronically (Sabu and Das, 2018). One application of OCR is using it in the medical context, such as on clinical texts or on drug prescriptions.

Drug prescriptions are an essential part of patients' lives, and these prescriptions contain information that are valuable for their overall well being. The application of optical character recognition techniques can prove to be beneficial in extracting the needed information from drug prescriptions. After extracting prescription information, it would be easier to organize and utilize for different purposes, such as keeping track of health medications, dosages, and frequency of intake, all with the convenience of digital technology. The challenge is how to implement and utilize OCR to make digitizing and organizing prescriptions as efficient as possible.

Current studies under OCR are usually limited to either text extraction or text annotation, especially in the medical context. The combination of both steps in OCR can prove to be useful in the medical field for various applications. Studies that are able to both extract and parse are limited to prescriptions that use Latin abbreviations and other formats for intake frequency that is not used in the Philippines. An example is the study by Mccarthy et al. (2020). In the Philippines, a natural language format is used.

### *1.1 Research Objectives*

The objective of the study is to develop a web application that extracts information from typewritten Philippine drug prescriptions. Specifically, this study

aims to:

(1) Create a dataset of typewritten drug prescriptions from the Philippines;

(2) Utilize an existing OCR tool, Tesseract, to transform Philippine prescription images to text;

(3) Utilize a Natural Language Processing (NLP) tool, SpaCy, to extract medication information by annotating the resulting text from the OCR tool;

(4) Incorporate OCR and information extraction techniques for the automatic transformation of prescription images into textual medication information.

### *1.2 Scope and Limitation*

This study focuses on optical character recognition and information extraction from typewritten medical prescriptions from the Philippines.

The prescriptions obtained for this study are limited to those written in the English language and are from the Philippines. These include old samples from previous studies, new samples obtained given the health protocols in place at time of writing, and synthetic data generated using collected samples. The synthetic data generation process only includes converting handwritten text to digital text and converting the formats of prescriptions that do not fit with the samples.

The prescriptions in the dataset were extracted for the following information: dosage, drug, duration, form, frequency, and strength, #tablets, when, how, and remarks. The data that will be collected from the prescriptions do not include sensitive data such as names, signatures, and license numbers.

This study was implemented as a web application using the OCR tool Tesseract and spaCy. SpaCy is an industrial-grade NLP tool that can be used to perform information extraction using its various features of NLP techniques.

## 2. METHODOLOGY

### *2.1 Data Preparation*

The dataset contains digital typewritten medical prescriptions created using real medical prescriptions from patients who voluntarily consented to having their prescriptions processed for the study given any identifying information about them is omitted. A given prescription was first digitized by manually typing the prescription information into three format templates that are based on existing typewritten prescriptions, then said prescription was digitized based on its own format. The dataset contains 166 digital and processed prescriptions in total.

A sample from the dataset was used to create a set of images that simulated a user taking a picture of their prescriptions from different angles and lighting conditions with their mobile phones to test for skewness.

### *2.2 Optical Character Recognition*

Optical Character Recognition involves six major steps: image acquisition, preprocessing, character segmentation, feature extraction, character classification, and post processing (Islam et al. 2017). The first step is having an image inputted into the OCR. Next, the image is usually preprocessed to increase the effectiveness of the OCR. Some preprocessing tested are image binarization, a skew-correction method taking advantage of four corners of a paper (Gangal et al. 2021), two rotation correction methods, one using Hough Lines (Brilenkov 2021), another using Histogram (Bagdanov and Kanai 1997) and shadow removal (K 2020). The characters in the image are then segmented by letter. The next step usually relies on either pattern recognition or feature detection (Woodford 2021). These can then be fed into a model to classify characters or words. The output is sometimes not recognizable. Thus, post processing is done by using spell-checking and correction.

OpenCV was used to preprocess the image before inputting it to the OCR module. The OCR system extracts the text present in the prescription images that will be uploaded to the web application. After which, post processing was done using regular expressions to remove minor errors read by the OCR.

The typewritten medical drug prescriptions used in this study were limited to the formats shown in Figures 1, 2, and 3. In the first format, the Sig line is in a phrase with the dosage, form, frequency, and duration indicated. The drug and strength in this format is indicated in the first line. The second format is a simpler format wherein the drug and strength are in the first line and in the second line, the frequency is written in shorthand. The third format is the same as the second format, with frequency written completely. In addition, the information in the same prescription formats were tested in a different ordering.

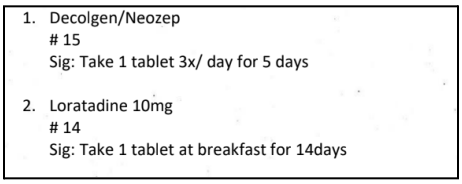


Fig.1. Prescription Format 1

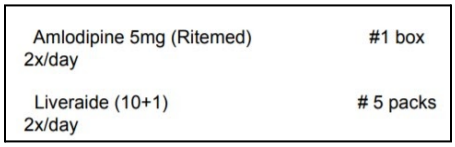


Fig.2. Prescription Format 2

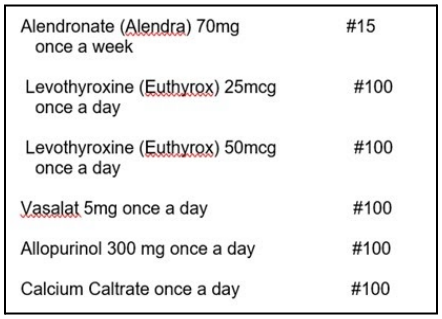


Fig.3. Prescription Format 3

## *2.3 Information Extraction*

Significant studies in information extraction of medication information in recent years can be attributed to the i2b2 Medication Challenge in 2009 which inspired the development of various annotation schemas for clinical text. In the 2018 i2b2 challenge, Med-7 was created. Med-7 is an information extraction model for clinical text that is freely accessible through a Python package for spaCy (kormitzilin 2021). Med-7 identifies seven entities namely dosage, drug, duration, form, frequency, route, and strength. The model had an average macro F1 score above 0.9. Due to the high accuracy of the model and the accessibility of the spaCy library, Med-7 is chosen to be trained with the new dataset and the new model will be utilized for this study. SpaCy, an industrial-grade NLP tool, was used to perform information extraction using its various features of NLP techniques, specifically its named entity recognition feature which uses word embeddings and a deep convolutional neural network (Honnibal 2017b). The transition based approach used by spaCy for named entity recognition is like a state machine, wherein all the words are first placed in a buffer and then the system goes through them one by one, for each word there is an equivalent action that will be determined, which is either to move to the next word or make a prediction on the entity (Honnibal 2017a). Overall, this approach for named entity recognition provides a balance in efficiency, accuracy, and adaptability to the feature (Honnibal 2017b).

Given the annotated dataset, a subset was created with only the new entities that are not part of Med-7 namely #units, when, how, and remarks. These datasets were then split in half into training and validation sets. The models were trained and the best models produced were tested with the test dataset.

## *2.4 Design and Implementation Issues*

Implementation issues for the Optical Character Recognition tool mostly stem from the preprocessing of the images. Given that the dataset created for this study is consistent in design, tests using it are very controlled. In order to counter this, tests were made on skewed images, with a sample of 16 prescriptions printed out and photographed in order to perform a user simulation. When images are skewed, the OCR tool proved to have difficulty in reading text on images too distorted, thus more preprocessing techniques were tested out before running the images through the OCR again. The preprocessing techniques used are shadow removal, getting the four corners and cropping accordingly using Gaussian blurring and canny edge detection, and lastly two skew correction techniques, one using canny edge detection and Hough lines and another using a Histogram scoring algorithm. In addition, experiments were also done with multiple techniques done together to see if performance improved.

Three main issues were faced when implementing the information extraction model. First was the difficulty in annotating the dataset manually as it required to count the character location of the target's word starting letter and ending letter. In order to simplify the process, the tool, spaCy NER annotation tool by Agate was used. Second is the difficulty in training MED-7 with the new dataset. When creating the configuration set using Med-7, the setting incurred errors during training. Training an existing NER model in spaCy without the data it was trained on before leads to catastrophic forgetting.

In implementing the OCR tool and NER model in the web application, the main issue faced was the splitting of the entities identified into rows in a table.

Since the entities within a prescribed drug can be placed in random locations there is clear division between each drug prescribed when looking at the identified entities. To resolve this issue, two solutions were implemented. One is by using image processing to split the image into each block of text before it is fed into the OCR tool. The second approach is by using a brute force approach by looping through the entities identified into a dictionary. If an entity is found twice, it assumes that the second entity is the next drug in the prescription.

## 3. RESULTS AND DISCUSSION

The dataset was validated by two pharmacists individually. Based on their comments, appropriate changes were made to the dataset, such as fixing typos and adding missing information. It was noted by both pharmacists that the entity "#TABLETS" was quite misleading as the prescribed medicine may come in the form of a capsule or tube. This lead to the change of the entity from the proposed "#TABLETS" to "#UNITS". Another notable comment made by one of the pharmacists is the fact that the dataset created is valid and usable in pharmacies, however some prescriptions did not have the generic name of the medicine listed, only the brand name, which is required under the Generics Law.

Upon testing the dataset with Tesseract, the mean character error rate (CER) and word error rate (WER) was 5.0368 and 11.6126 respectively. The dataset created for this study is a compilation of medical prescriptions digitally typed on a half letter sized white background with the format of Arial as font (size 12 ).

To further improve the performance of Tesseract OCR, initial preprocessing on the default dataset images was done, i.e. converting the image into grayscale and thresholding on that image. This resulted in a mean CER of 5.7311 and a mean WER of 13.8419. This shows that the OCR actually performs a bit worse than that of when there was no preprocessing done in the first place. This may be attributed to the fact that the original image is clearer than the pre-processed one.

Different preprocessing methods were tested on both datasets -- the default dataset that contains the digitally typed prescriptions, and the dataset that contains simulated prescription images taken with a mobile phone camera, represented as the skewed dataset. For the default dataset, the best image preprocessing method is a combination of shadow removal and skew corrections (Hough Lines) which resulted in the lowest mean CER and WER rate of 4.8591 and 10.8615, respectively (see Table 1). The same goes for the skewed dataset with mean CER and WER of 28.9828 and 31.6994, respectively, which are the lowest among the methods tested on the skewed dataset (see Table 2).

Shadow removal (K 2020) consists of splitting the image into three (3) planes for red, green and blue colors. For each of the three planes, a dilated image was created using a 7x7 pixel square. Then, blurring was done on the dilated image. An image of the absolute difference between each blurred image and the plane was created and normalized. The three resulting normalized images were then merged. The rotation correction algorithm (Brilenkov 2021) converts the image into grayscale and finds edge lines in the image using Canny (Sahir 2019). It then finds Hough lines on those edges. The angles were calculated for each Hough line. Finally, the image was rotated according to the median angle.

Table 1. OCR Preprocessing Results (Default Dataset)

| Preprocessing | Mean CER (%) | Mean WER (%) |
|---|---|---|
| Skew Correction (Hough Lines) | 4.8591 | 10.8615 |
| Shadow Removal and Skew Correction (Hough lines) | 4.8591 | 10.8615 |
| Skew Correction (Histogram) | 5.0451 | 11.1692 |
| Shadow Removal and Skew Correction (Histogram) | 5.0451 | 11.1692 |

Table 2. OCR Preprocessing Results (Skewed Dataset)

| Preprocessing | Mean CER (%) | Mean WER (%) |
|---|---|---|
| Skew Correction (Hough Lines) | 73.4642 | 76.7974 |
| Shadow Removal and Skew Correction (Hough lines) | 28.9828 | 31.6994 |
| Shadow Removal and Skew Correction (Histogram) | 21.3109 | 36.6587 |
| Shadow Removal and Skew Correction (Hough lines) with Binarization | 73.4642 | 76.7974 |
| Shadow Removal and Skew Correction (Hough lines) with Binarization | 36.0724 | 42.1139 |
| Four Corners Detection | 57.3312 | 66.8054 |

Using the default dataset, the OCR tool sometimes places the number of units at the end of the resulting text when the number of units is located far from the main content of the image. However, the placement of the number of units does not seem to affect the NER model. There were instances where the backslash was replaced with "i". The OCR was also not able to read "1/2" and convert it into "%". The rest are the

OCR misreading characters like "x" from "1x/day".

Postprocessing was able to correct minor mistakes like reading backslash with an “i” or reading “5” as “S”. On the skewed dataset, some errors were caused by the rotation preprocessing. After post processing, the results improved marginally. In the default dataset, the resulting mean CER and WER became 4.5506 and 9.6990 while on the skewed dataset it became 25.0945 and 31.7346 (see Table 3).

Two different models were trained for information extraction. The first model was trained with the complete list of entities (Model-1) while the other was trained to recognize only the entities which are not part of Med7, namely, #tablets, when, how, and remarks (Model-2). For both models, the configuration for training was set with the recommended parameters for efficiency generated in the spaCy training quickstart webpage (spaCy nd), a learning rate of 0.001 and Adam as its optimization algorithm. Adam, also known as adaptive moment estimation, is a widely used optimization algorithm that is known to be fast and requires minimal memory (Gupta 2021). Model-1 received an overall score of 0.90, while Model-2 received an overall score of 0.88. Model-1 was tested as a standalone model and as part of the Med7 pipeline, while the Model-2 is tested only as part of the Med-7 pipeline. Model-2 was not tested as a standalone model like Model-1 because it was only trained to identify four (4) entities.

Table 3. OCR Results with Shadow Removal and Skew Correction using Hough lines

| Dataset | Mean CER (%) w/out Post Processing | Mean WER (%) w/out Post Processing | Mean CER (%) w/ Post Processing | Mean WER (%) w/ Post Processing |
|---|---|---|---|---|
| Default Dataset | 4.8591 | 10.8615 | 4.5506 | 9.6990 |
| Skewed Dataset | 28.9828 | 31.6994 | 25.0945 | 31.7346 |

Model-1 was able to predict the correct entity most of the time with errors mostly having text being unlabelled. Overall, based on the precision, recall, and F-1 score for each entity, Model-1 was able to predict all the entities more than 50% of the time with the exception of Remarks (see Table 4). For the entity Remarks, its low score compared to the other entities may be due to the fact that only a small percentage of the dataset contains these entities.

The Med-7 + Model-1 pipeline performed worse than the standalone model except for the entities strength, dosage, and how, two of which are entities that are recognized by Med-7. The precision, recall, and F-1 scores of the model shows that the model is able to identify all the entities more than 50% of the time except for Remarks, similar to the standalone model. (see Table 5).

Table 4. NER Test Results for Model-1

| Entity | Precision | Recall | F-Score |
|---|---|---|---|
| Dosage | 0.9294 | 0.9518 | 0.9405 |
| Drug | 0.9221 | 0.9726 | 0.9467 |
| Duration | 0.9459 | 0.7955 | 0.8642 |
| Form | 0.9583 | 0.8118 | 0.8790 |
| Frequency | 0.9058 | 0.9398 | 0.9225 |
| Strength | 0.9697 | 0.9897 | 0.9796 |
| #Units | 0.9722 | 0.9790 | 0.9756 |
| When | 0.9821 | 0.833 | 0.9016 |
| How | 0.9600 | 0.9600 | 0.9600 |
| Remarks | 0.9167 | 0.500 | 0.6471 |
| Overall | 0.9440 | 0.9194 | 0.9316 |

Table 5. NER Test Results for Med-7 + Model-1

| Entity | Precision | Recall | F-Score |
|---|---|---|---|
| Dosage | 0.8333 | 0.9639 | 0.8939 |
| Drug | 0.3728 | 0.5822 | 0.4545 |
| Duration | 0.9722 | 0.7955 | 0.8750 |
| Form | 0.8280 | 0.9059 | 0.8652 |
| Frequency | 0.5972 | 0.6466 | 0.6209 |
| Strength | 0.4384 | 0.9897 | 0.6076 |
| #Units | 0.8309 | 0.4126 | 0.5514 |
| When | 0.9286 | 0.1970 | 0.3250 |
| How | 0.9231 | 0.9600 | 0.9412 |
| Remarks | 0.7778 | 0.3182 | 0.4516 |
| Overall | 0.5991 | 0.6659 | 0.6308 |

Finally for Med-7+Model-2, the poor performance may be due to the first model being more fitted to the test dataset and the second model being trained with more data for the Med-7 entities. The precision, recall, and F-1 score of the model are very

similar to the previous model, except for the entity "When", which showed a significantly lower score when compared to the two other configurations (see Table 6).

Five-fold cross validation was also conducted on the three model configurations to test for overfitting. The results from the cross validation shows that Model-1 had an f-score of 0.9792. For both Med-7+Model-1 and Med-7+Model-2, the F-1 score is 0.7050.

To further test the overall performance, both the OCR and NER modules were combined and tested. This resulted in an F-score of 0.8816, which means the OCR and NER modules were able to extract, read, and categorize the information fairly accurately. As seen in Table 7, the overall result is almost similar to the result from the standalone model with complete entities when tested individually.

Table 6. NER Test Results for Med-7 + Model-2

| Entity | Precision | Recall | F-Score |
|---|---|---|---|
| Dosage | 0.8333 | 0.8434 | 0.8383 |
| Drug | 0.4000 | 0.4521 | 0.4244 |
| Duration | 1.000 | 0.7955 | 0.8861 |
| Form | 0.8148 | 0.7765 | 0.7952 |
| Frequency | 0.5798 | 0.5188 | 0.5476 |
| Strength | 0.5680 | 0.9897 | 0.7218 |
| #Units | 0.9524 | 0.4196 | 0.5825 |
| When | 0.1333 | 0.1818 | 0.1538 |
| How | 0.9231 | 0.9600 | 0.9412 |
| Remarks | 0.8750 | 0.3182 | 0.4667 |
| Overall | 0.5998 | 0.5983 | 0.5991 |

Furthermore, four (4) sets of distorted or skewed samples of the same 16 images each are collated and tested with the model – (1) under dark lighting, (2) under normal lighting, (3) slanted images under normal lighting, and (4) four corners visible in the frame. Each set contains the same prescriptions captured differently each time using a smartphone camera to simulate real users using the app. All of these sets are combined and tested in the OCR and NER module in one go. The overall f-score came out to be 0.9215 which means that the model can read from distorted and skewed prescription images fairly accurately. Table 7 shows the results per entity. This can also be attributed to the preprocessing done to the images in the OCR module.

Table 7. NER Results for OCR and NER Tests Combined

| Entity | Precision | Recall | F-Score |
|---|---|---|---|
| Dosage | 0.6549 | 0.9180 | 0.7611 |
| Drug | 0.9279 | 0.9371 | 0.9317 |
| Duration | 0.9152 | 0.8429 | 0.8714 |
| Form | 0.9297 | 0.9061 | 0.9168 |
| Frequency | 0.9308 | 0.9288 | 0.9289skewed |
| Strength | 0.9020 | 0.9634 | 0.9304 |
| #Units | 0.7711 | 0.9193 | 0.8352 |
| When | 0.7479 | 0.7875 | 0.7530 |
| How | 0.8489 | 0.8864 | 0.8588 |
| Remarks | 0.9214 | 0.8914 | 0.9048 |

The system also suggests medicine names which are similar to the medicine names read from the OCR in case the OCR misreads the medicine names. The suggestions are checked for its correctness. The OCR was able to get the medicine names 108 out of 345 (31.30%) medicine names. The suggestions were able to correct 153 out of 237 (64.56%) incorrect medicine names while 84 out of 237 (35.44%) of the incorrect names were not able to be corrected.

## 4. CONCLUSION AND RECOMMENDATION

For image preprocessing, it was found that shadow removal in addition to skew correction using Hough lines produced the lowest error rates among the different preprocessing techniques. While for the information extraction model, based on the results of the different configurations tried, the best by far is the standalone model trained with complete entities having a score for almost all entities of 80%. This significantly better result is due to the model being better fitted to the test dataset.

For the OCR tool, it is suggested that tests be done on a variety of different fonts to further test the capabilities of the OCR tool. In addition, more metrics other than error rate can be tested to gain an even deeper understanding of the tools. Other image preprocessing techniques can be explored to further improve the accuracy of the OCR tool. Based on preliminary tests on the web application, improvements can be done for the recognition of prescriptions with

multiple medications. With multiple medications, the website sometimes cannot recognize which entities belong together in a single row. Improvements in OCR preprocessing, may help reduce errors for this.

Improvements can be made to the accuracy of the spaCy named entity recognition model given more training data. The performance of the model can also be improved with further testing and trials of different training parameters, such as different learning rate, different optimizers, and trying to train the model for accuracy over efficiency. Further improvements can also be made to the method in which the result of the NER model is split into rows, given that the used method is a brute force approach and is prone to errors. Lastly, the speed in which the whole system works can also be improved, with better algorithms or approaches.

For future work, this system can be implemented in medication management applications as a feature. This can help users to easily add prescription information into their medication management application without having the need to manually type down the information.

## 5. REFERENCES

A. Bagdanov and J. Kanai. 1997. Projection profile based skew estimation algorithm for JBIG compressed images. In Proceedings of the Fourth International Conference on Document Analysis and Recognition, Vol. 1. 401–405 vol.1. https://doi.org/10.1109/ICDAR.1997.619878

Ruslan Brilenkov. 2021. Document Scanner from Scratch with Python. https://medium.datadriveninvestor.com/document-scanner-from-scratch-with-python-6a55c7ce1423

spaCy. nd. Training Pipelines & Models. https://spacy.io/usage/training#quickstart

Ayush Gupta. 2021. A Comprehensive Guide on Deep Learning Optimizers. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/

https://doi.org/10.1136/jamia.2010.003855 arXiv:https://academic.oup.com/jamia/article-pdf/17/5/528/2477130/17-5-528.pdf

Ayushe Gangal, Peeyush Kumar, and Sunita Kumari. 2021. Complete Scanning Application Using OpenCv. CoRR abs/2107.03700 (2021). arXiv:2107.03700 https://arxiv.org/abs/2107.03700

Matthew Honnibal. 2017a. SPACY'S ENTITY RECOGNITION MODEL: incremental parsing with Bloom embeddings residual CNNs. https://www.youtube.com/watch?v=sqDHBH9IjRU

Matthew Honnibal. 2017b. spaCy's NER model. https://spacy.io/universe/project/video-spacys-ner-model#gatsby-noscript

Noman Islam, Zeeshan Islam, and Nazia Noor. 2017. A Survey on Optical Character Recognition System. arXiv:1710.05703 [cs.CV]

Ravi K. 2020. Shadow removal with open-CV. https://medium.com/arnekt-ai/shadow-removal-with-open-cv-71e030eadaf5

A. Kormilitzin, N. Vaci, Q. Liu, and A. Nevado-Holgado. 2021. Med7: A transferable clinical natural language processing model for electronic health records. Artificial intelligence in medicine 118 (2021).

kormitzilin. 2021. Med7. Retrieved August 15, 2021 from https://github.com/kormilitzin/med7

T. Mccarthy, L. Pineda, and A. Reyes. 2020. Medication Management Application to Assist Older Adults with the Indications and Contraindications of Prescribed Drugs.

Abin M Sabu and Anto Sahaya Das. 2018. A Survey on various Optical Character Recognition Techniques. In 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). 152–155. https://doi.org/10.1109/ICEDSS.2018.8544323

Sofiane Sahir. 2019. Canny edge detection step by step in python-computer vision. https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

C. Woodford. 2021. Optical character recognition (OCR). https://www.explainthatstuff.com/how-ocr-works.html