OPTICAL CHARACTER RECOGNITION ON MEDICAL RECORDS

G. Naga Jaswanth, B. Hari Charan

Project Report November 2023
Course: DS250 - Data Analytics and Visualization
Semester: 2022-23M Instructor: Dr. Rajesh Kumar Mundotiya

Abstract

This paper focuses on development of workflow of an OCR (Optical Character Recognition) project in order to extract textual data from Medical or Patient Records. The primary objective is to facilitate the digitization of medical documents, including patient records, doctor prescriptions, and reports. This helps in providing methodology to the hospital management in order to digitize their records without the help of extensive manpower. This further aids in improving accessibility of important information related to the hospital patients. The report aims to explain the project, including with all the functions involved in it. This will help anyone using the workflow provided better insight on how to improve it and fine-tune it towards their requirement.

The main challenge here is that most of the records available at a hospital is hand-written, and in this case, many standard OCR models fail to perform with reasonable accuracy. To tackle this problem, we leverage a different techniques comprising broadly of Image Processing, Text Extraction and Natural Language Processing, and thus making it a hybrid workflow. These three subroutines form basis of the project's three main verticals. At the end, this report uses the aforementioned workflow on the patient records data set present in the Project Directory, to perform basic exploratory data analysis. At this stage of workflow, this can be a common application, in order to explore the data.

With the help of this project, if the management is the audience, taking it as a workflow they can improvise on these methods and develop their own procedure in order to digitize the patient records and if the audience is an individual he/she can use these to simply extract text from a record in order to gain insight on the particular patient's health. Beyond facilitating record digitization, successful implementation of this workflow promises enhanced efficiency in managing patient information, quicker access to crucial medical data, and opportunities for personalized patient care through insightful data analyses.

Contents

Chapter 1

Problem Statement

1.1 Problem

The problem imposed as a question asks us to provide a workflow via which the user is enabled to detect text from an image or series of images with the help of important procedures including image preprocessing, text extraction and word processing or post processing. This project's underlying goal is to extract different words from the given input image and then generates a tuple for each word extracted consisting of two elements, first one being the closest English word in English dictionary to the given word and the second element in the tuple contains list of atmost 5 closest medical terms to the given word. The output in this format aids a person performing using this module to get better clarity on the word mentioned and make the right selection.

Chapter 2

Description

2.1 Background and Motivation

The problem statement here is that, we are given a patient's health record in form of image, and we need to extract the text from it. Although this may seem a very basic problem with the kind of AI tools available in this generation, it is not an inherent part of any hospital's data storage architecture.

The main inspiration behind the problem is that, it is surprising to note that in order to do so even in this era of extensive digitization, most of the hospitals do not employ an infrastructure built on the foundations of technology. They stick to traditional input taking methods. This is due to the fact that many of these healthcare institutions still take patient records into account only via hand. When we admit someone at the hospital, we have to fill a form regarding the details of the patient, and when a medical practitioner takes information regarding patients condition, he/she does so by hand. This causes great difficulty in digitizing the records of a patient, hence making these records virtually inaccessible. Now, if we hire humans, manual data entry involves the use of human operators to input data into a computer system or database, and this process can be time-consuming and error-prone.[1] This requires the need of an automation technology that can allow the practitioners to take input data in the traditional way and yet with the help of technology, we can convert this input into data that can be stored in the hospital records for further usage, mostly by management.

2.2 OCR

OCR which stands Optical Character Recognition, implies collectively to the algorithms that convert an image to text, either readable by humans or machines in order to store it. Anyone who has used any OCR model in order to obtain text from image understands that these models, even today are not very accurate. More often than not it is the case that the model just generates gibberish upon calling it. The simple fact is that OCR is hard.[2]

The OCR technology can be broadly classified into two types:

- Machine Printed OCR
- Hand-written OCR

It is easy to understand that generally, hand written OCR is more tough process compared to printed letters in order to achieve good accuracy. The reason is simply that there are many different ”stroke” styles, in which humans write and it is not possible to train a model based on all possible calligraphies, whereas there are limited number of font styles for printed text to exist on. This poses a great challenge and hence requires careful preprocessing and additional post processing steps. Despite their differences, the workflow doesn’t differ much for both, certainly, some functions can be relaxed when moving from hand written OCR to printed text OCR.

2.2.1 Brief History

The first instances of OCR that were developed date back to 1914, when ”Emanuel Goldberg developed a machine that could read characters and then converted them into standard telegraph code” .[3] The current most popular OCR library ’Tesseract’ was developed first by HP Labs in 1980s and then it was later made open-source in 2005. But the fact remains that these OCR engines still take a knowledgeable computer vision practitioner to operate.[2]

2.2.2 Important tools

The following libraries are most widely used in the field Computer Vision, of which OCR is a subdomain.

(1) **Tesseract** Hewlett-Packard (HP) Labs initially created the Tesseract OCR engine as closed source software in the 1980s. With most of the Tesseract software written in C, making it incredibly fast. Google later started sponsoring the Tesseract development from 2006, once it was open sourced in 2005. This library now uses deep learning techniques based on long short-term memory (LTSM) networks. In this project, we do not play much with "pytesseract" which is the python package that is based on tesseract.

(2) **OpenCV** Major part of the project revolves aroud using this library as it provides many image manipulation techniques that play an integral part in the Image preprocessing and segementation procedures. This is done to improve our OCR accuracy, and in this way it fits into the OCR workflow of any experienced practioner.

2.2.3 OCR in HealthCare

Healthcare OCR is basically OCR with an emphasis on healthcare, which aims to transform the way that medical records, prescriptions, reports, and other forms are managed in the industry. To enable the digitization of medical records for simpler access, analysis, and storage, it comprises converting handwritten or printed documents into machine-readable text.

By automating the extraction and interpretation of medical data, OCR is essential to improving the productivity of administrative operations in the healthcare industry. OCR simplifies patient information management by transferring paper-based records into electronic formats. This guarantees prompt access to vital medical information while lowering transcription errors that arise from manual transcription. Faster patient admissions, prescription filling, and diagnostic procedures are made possible by this technology, which benefits patients and helps doctors make more informed decisions.[4]

2.2.4 HealthCare OCR in India

OCR in healthcare makes use of sophisticated algorithms and neural networks currently. The most popular use case of OCR model is ABBYY FlexiCapture, an OCR software that can help healthcare establishments digitize paper-based patient records.[1] But as we know in a country like India, where there are thousands of hospitals and countless number of patients and hence their records, it can be quite a challenging task to digitize even a single hospital's records. We are far from thinking about centralizing it. Since most of these hospitals are Government hospitals, it is not economically feasible for them to maintain a department of manual labour who actively digitize records. Therefore, there arises a need for an OCR system, that efficiently and accurately converts these records into text using their images. This data can then be stored in a proper database, in the hospital's infrastructure. This not only helps management to retrieve patient data easily and even remotely, but also helps National analytics in gaining much better realization of current medical situation, even geographically.

2.3 NLP

NLP stands for Natural language processing, a field of artificial intelligence focused on enabling computers to understand, interpret, and generate human language. NLP techniques help machines comprehend language patterns, semantics, and context to perform tasks like language translation, sentiment analysis, text summarization, and speech recognition. Any good OCR project, involves post processing of text obtained through the process of OCR. This is due to the fact that most of the obtained string is usually not easily understandable by humans, so it has to be cleaned in such a way that we can understand them better.

Chapter 3

Contribution

3.1 Project Working

3.1.1 Structure

——Patient_Records

— ——Ache

— ——Ache_page-0001.jpg

— .

— .

— .

——Samples

— ——kohli.jpeg

— ——Walmart.jpeg

——README.md

——east.py

— .

— .

——page_detection.ipynb

——page_detection.py

——ocr_processing.ipynb

——ocr_with_east.py

——word_processing.ipynb

——word_processing.py

——ocr_complete.py

——analysis.ipynb

3.2 Image Preprocessing

This is implemented using module: ”page_detection.py” Input: Image Output: Converted Image to extract only page.

This Python script operates as an all-encompassing tool tailored for refining images to facilitate efficient text extraction. Leveraging libraries like OpenCV (cv2), NumPy, and Pandas, it presents a series of operations aimed at augmenting image clarity and effectively extracting textual content.

The code begins by introducing various functions, including ’convert_to_portrait’ and ’resize_image,’ meant for adjusting image orientation and size while preserving their original proportions. ’Edge_detection’ assumes a pivotal role, leveraging the Canny edge detection algorithm to precisely isolate and extract image edges. Canny Edge detection algorithm is a hybrid multi-stage algorithm, it includes:

(1) Applying Gaussian Pass Filter to reduce noise in the image by smoothening out low intensity edges

(2) Finding Intensity Gradients using four filters in horizontal, vertical and two in diagonal directions to detect edges.

(3) Applying Non-maxima suppression in order to filter out suspicious edges

(4) After NMS, applying double threshold will determine potential edges

(5) Finally. edge tracking by hyteresis.

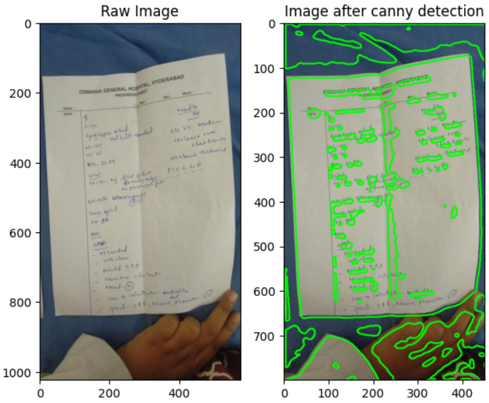


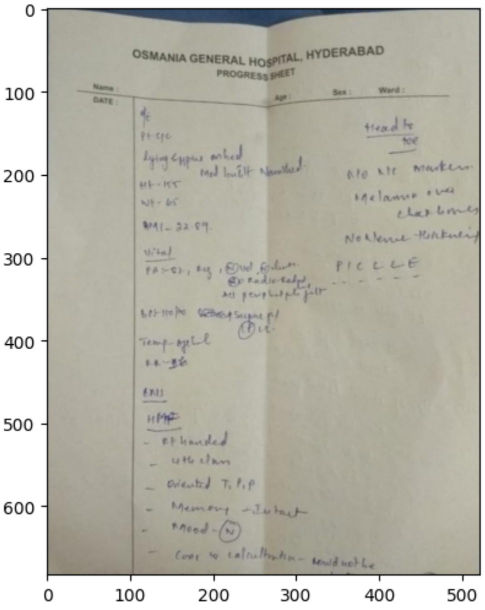
Figure 3.1. Images before and after canny edge detection

Another crucial function, 'get_contour,' concentrates on extracting contours from processed images, with 'four_corners_sort' meticulously organizing these contours to accurately represent document corners. 'Contour_offset' further refines the corner positions for a precise delineation.

Embedded within the script are functions like 'find_page_contours' and 'persp_transform,' dedicated to pinpointing document edges and rectifying perspectives. This correction ensures that the image resembles an overhead capture, facilitating accurate text or content extraction.

The 'cleaned_image' function streamlines this process by orchestrating edge detection, contour extraction, and perspective correction. This harmonized workflow produces optimized images suitable for subsequent text extraction or analysis.

Image formed after applying perspective transform is without any extraneous borders as shown.



OSMANIA GENERAL HOSPITAL, HYDERABAD
PROGRESS SHEET

Name :
Age : Sex : Ward :
DATE :

c/c
Pt-c/c
lying supine on bed
Mod built Nourished.
Ht-155
Wt-65
BMI-22.89.
Vital
PR-82, reg, [illegible]
All peripheral pulse felt
BP-110/70 [illegible]
Temp-afebrile
RR-18
CNS
HMF
- Rt handed
- 4th class
- Oriented T, P, P
- Memory - Intact
- Mood - (N)
- Cons & calculation - could not be

Head to toe
No IV markers
Melanin over chest bones
No Nerve thickening
PICCLE

FIGURE 3.2. Image after perspective transform

In summary, this script shows a comprehensive workflow for image preprocessing, meticulously identifying edges, contours, and rectifying perspectives. These steps fortify the images for efficient text extraction or further analysis, amplifying the efficacy of Optical Character Recognition (OCR) and similar text extraction algorithms reliant on superior image quality.

3.3 OCR Processing

This is implemented using module: ”ocr_with_east.py” Input: Image of page Output: Perform OCR on extracted RoIs using EAST

This procedure contains only one generic function call of the function ’image_to_string’ from the ’pytesseract’ module. However, in order to obtain more accurate localized results, there is also an implementation of EAST (Efficient and Accurate Scenic Text Detection) algorithm, with the help of which, we can efficiently extract text from an image having non-textual data in comparable amount.

The implementation is done with the help of 'detect_localized_text' function. This Python script is built to localize and extract text from images using the EAST (Efficient and Accurate Scene Text) detection model, Pytesseract OCR, and OpenCV functionalities. The detect_localized_text function serves as the core mechanism, resizing the input image for compatibility with the EAST model, predicting text regions, and filtering these regions using Non-Maximum Suppression. For each identified text box, the script computes a padded region to enhance OCR accuracy and utilizes Pytesseract to extract text content. Visualizing the detected text regions on the image, it outlines the boxes and annotates the text. This module, hence, exemplifies the usage of the text detection and extraction routine on an image, demonstrating its capabilities in identifying and extracting text content from images with visual representations for each detected text area. This process is also know as extraction of Regions of interest (RoIs). It is nothing but focusing our attention to a particular region and perform OCR to that region independently.

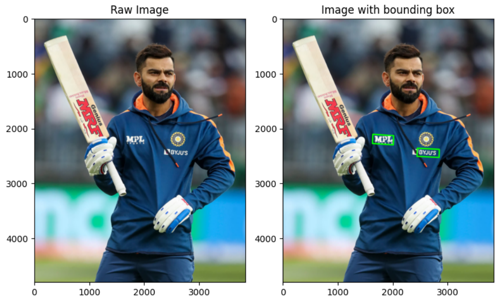


FIGURE 3.3. Image before and after finding bounding boxes using EAST

3.4 Text Processing

This is implemented using module: ”word_processing.py” Input: Text Output: Tuple Containing Closest English Word and upto 5 closest medical terms(if any) unless specified.

This part of the workflow first refers to 'MedicineNet.com' and safely scrapes the website in order to make a medical terms Glossary, and stores it alphabetically in a file 'med_terms.txt' for further usage in program. This Python script aims to enhance Optical Character Recognition (OCR) results by correcting OCR text, validating against English words, and suggesting medically relevant alternatives if the recognized word appears to be incorrect or misspelled. The editDistance function calculates the Levenshtein edit distance between two strings, serving as a metric for identifying similar words.

The get_med_dict function generates a dictionary of medically relevant terms from an external file, organizing them by their first letter for efficient lookup. Utilizing this dictionary, the get_closest function retrieves the closest medically relevant terms to a given OCR word within a defined edit distance threshold.

The cleaned_words function processes the OCR text by first extracting and lowercasing alphabetic words. It then employs the Speller from the autocorrect library to correct potential misspellings within the OCR text. Next, it validates these corrected words against the WordNet corpus to filter out non-English words. For each OCR word, it identifies medically relevant alternatives using get_closest and constructs a dictionary associating each word with its corrected English form and potential medical alternatives.

In the main function, which is currently empty, one could integrate this script by providing OCR-generated text as input. The script would then apply these functions to the text, generating a comprehensive dictionary associating each recognized word with its corrected English version and potential medically relevant alternatives, thereby refining OCR results for further analysis or application.

Chapter 4

Analysis

4.1 Comparative Analysis

Performing comparative analysis between the proposed workflow with general data science workflow shows that both the processes have a lot more in common than expected along with some key distinctions.

While both involve diverse data, the OCR workflow is uniquely tailored to decipher handwritten medical entries characterized by varying styles and intricate medical language. Specific pre-processing techniques, such as image enhancement and noise reduction, edge detection, etc., are emphasized to optimize OCR accuracy for healthcare-specific content. Domain-specific considerations, including the focus on medical terminology and abbreviations, which were extracted from internet, underscore the specialized nature of the OCR workflow.

Integration challenges involve customization for healthcare systems, and a user-friendly interface is crucial for healthcare professionals to verify OCR results. In contrast, a general data science workflow lacks the specificity and domain-focused nuances embedded in the proposed OCR workflow, demonstrating its distinct adaptation to the challenges posed by handwritten medical records. This is because of the vast domain of data science applications. Hence with the help of domain abstraction and reducing its width we can increase the specificity of the problem and thus optimize the workflow.

4.2 Data Analysis

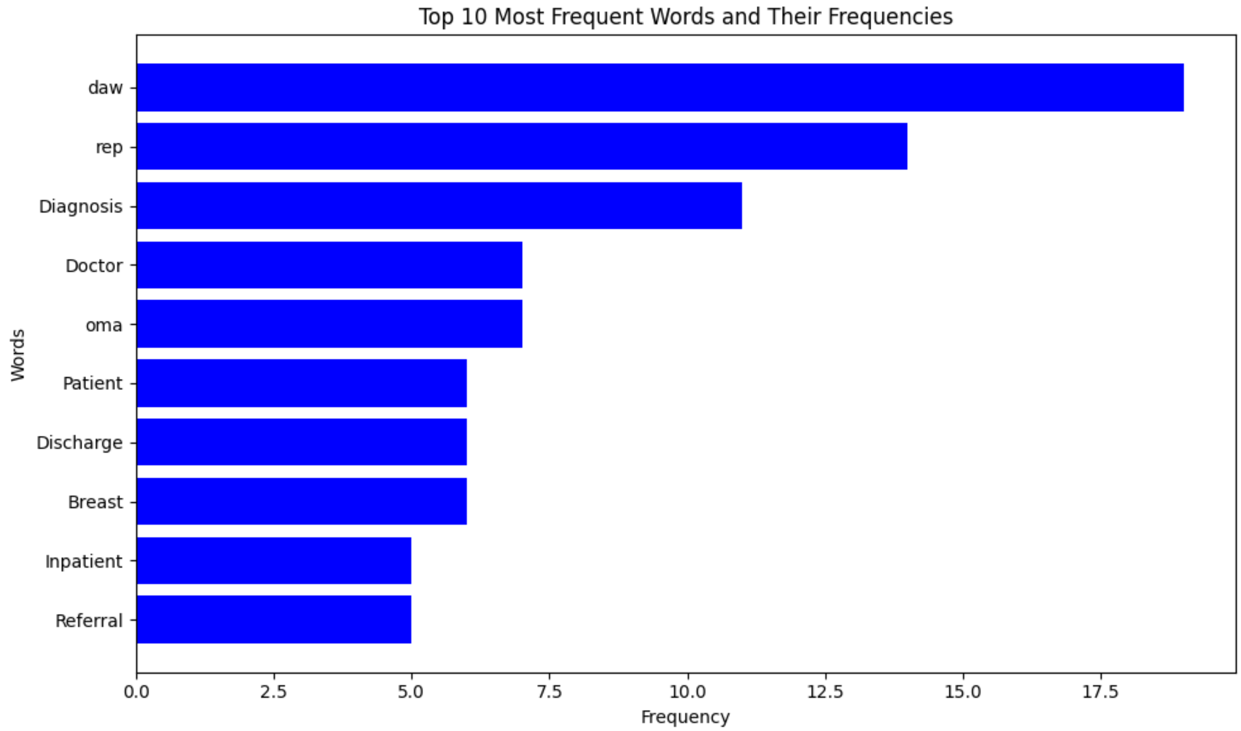


Figure 4.1. Histogram of medical words with their frequencies

Chapter 5

Findings and Summary

5.1 Findings and Summary

5.1.1 Data Analysis

Developing a workflow for Optical Character Recognition (OCR) on handwritten medical records is a complex yet pivotal endeavor, aiming to streamline the digitization of crucial healthcare information. Through rigorous exploration, several key findings emerged, guiding the project toward practical and effective solutions.

The foremost challenge encountered was the diverse nature of handwritten medical records. Physicians, nurses, and other healthcare professionals exhibit a wide range of handwriting styles, making it challenging for traditional OCR systems to accurately interpret the text. Additionally, handwritten entries often include medical jargon, abbreviations, and sketches, introducing further complexity. To address this, the project highlighted the significance of robust pre-processing techniques.

One major finding emphasized the critical role of pre-processing in enhancing OCR accuracy. Techniques such as image enhancement and noise reduction proved instrumental in preparing the handwritten text images for optimal recognition. By systematically addressing image quality issues, the project significantly improved the OCR system's ability to decipher intricate handwritten characters.

Model selection emerged as another crucial aspect of the project. Generic OCR models were found to be less effective compared to those specifically tailored for handwritten text recognition, e.g., usage of EAST model. The experimentation with various models related too EAST and related algorithms underscored the importance of choosing a model that aligns with the unique characteristics of medical handwriting. Models trained on diverse medical datasets demonstrated superior performance, showcasing the significance of domain-specific training.

The project also shed light on the need for continuous training of OCR models, as the text extraction poses a huge problem in the workflow, due to low accuracy of the model when the picture is not properly cleaned. Handwriting styles and medical terminologies evolve over time, necessitating regular updates to maintain the system's accuracy. The dynamic nature of healthcare documentation underscores the importance of an adaptive OCR model capable of learning from new examples and adjusting to emerging patterns.

Furthermore, domain-specific training emerged as a key factor in achieving accurate OCR results. Fine-tuning OCR models with medical data, extracted from net proved effective in recognizing specialized medical terminology, as we had to use NLP techniques to further optiimze textual findings. This finding emphasized the importance of tailoring OCR systems to the unique linguistic and contextual nuances of the healthcare domain.

Integration challenges were also addressed in the project's conclusion, since the project three major verticals which are not heavily correltaed. While OCR technology offers substantial benefits, seamless integration into existing healthcare systems requires careful consideration. Customization and compatibility adjustments may be necessary to ensure a harmonious workflow that complements the established processes of healthcare professionals.

The user interface was identified as a pivotal component in the successful implementation of OCR in healthcare settings. The findings underscored the need for a user-friendly interface that allows healthcare professionals to verify and correct OCR results, so that patient records can be utilized by them easily 'on-the-go' with the help of applications that use the workflow.

Lastly, the project concluded with a pragmatic evaluation of the cost-benefit aspect. Assessing the time saved through OCR implementation versus potential errors and scalability considerations provided a comprehensive understanding of the technology's economic viability in the context of healthcare digitization.

5.1.2 Future Application

In the rapidly advancing field of healthcare, this OCR workflow presents a wealth of potential for digitizing medical data in the future. With its advanced correction and validation capabilities, it has the power to greatly improve the accuracy of digitized medical records and ensure the precise extraction of patient information from scanned documents. By cross-referencing against a database of English words and suggesting medically-relevant alternatives, this system has the potential to streamline the entry of medical data, resulting in more accurate and standardized records. Even more exciting is the prospect of integrating machine learning models into the system, allowing it to continuously improve its accuracy through learning from corrections. This unprecedented approach to medical data digitization has the ability to revolutionize the process, ultimately leading to faster and more reliable extraction of patient information for crucial purposes such as diagnosis, treatment, and research.

Bibliography

[1] "Healthcare OCR - OCR Automation for Medical Sector — nanonets.com," https://nanonets.com/blog/ocr-for-healthcare/#introduction, [Accessed 21-11-2023].

[2] A. Rosebrock, "What is Optical Character Recognition (OCR)? - PyImageSearch — pyimagesearch.com," https://pyimagesearch.com/2021/08/09/what-is-optical-character-recognition-ocr/., [Accessed 20-11-2023].

[3] "Optical character recognition - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Optical_character_recognition, [Accessed 23-11-2023].

[4] D. Gifu, "Ai-backed ocr in healthcare," *Procedia Computer Science*, vol. 207, pp. 1134–1143, 2022.