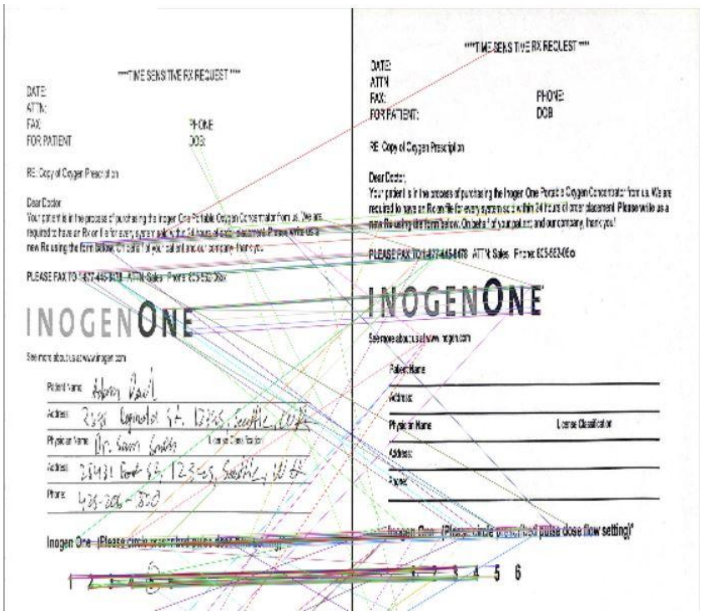
UCSB Data Science Capstone - Individual Summary

Abstract

Inogen, a medical technology company, aims to optimize patient care by accessing and analyzing data from physical concentrator patient prescription forms. The main focus of this project was to develop a custom Python script to accurately extract relevant information from the forms, achieving improved accuracy in text extraction and automating the data analysis process. Our project was able to achieve a moderately high accuracy, while simultaneously automating the entire data extraction process from when the form is scanned until the data is deposited into a usable database format. My role on the project was to focus on formatting scanned prescription forms, running computer vision techniques in order to optimize accuracy, and using open source optical character recognition (OCR) softwares to extract the text from the prescription forms.

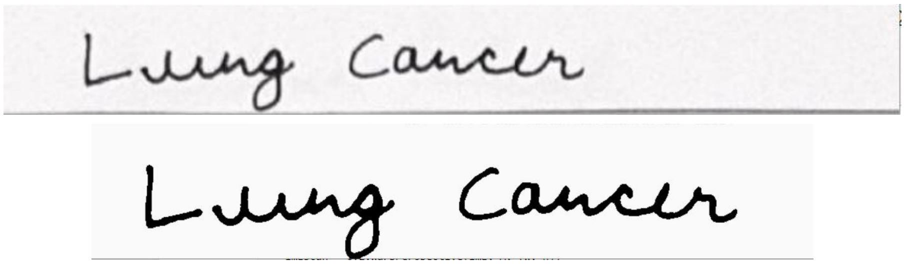
Methodology Overview

The dataset that we used during the course of this capstone project was made up of prescription forms that contain crucial information such as patient name, address, phone number, and flow setting (the prescribed level of oxygen concentration that Inogen's portable oxygen concentrator (POC) provides). It is important to note that the sample prescription forms did not contain patient information; instead the forms were filled out by volunteers. The form information that we had the volunteers fill out was also set as to maintain statistical significance when running the accuracy analysis. After we scan the prescription form, we need to ensure that all of the forms are the correct dimensions, skew and zoom. We do this by incorporating a key-point matching technique using OpenCV in python. The python script draws key-points on a control form as well as the scanned forms. After this it will attempt to match all of the points and take the top 25 best fit points. Once it has a collection of matched points, we again use OpenCV to morph the scanned form to fit the parameters of the control form.



OpenCV key-point matching

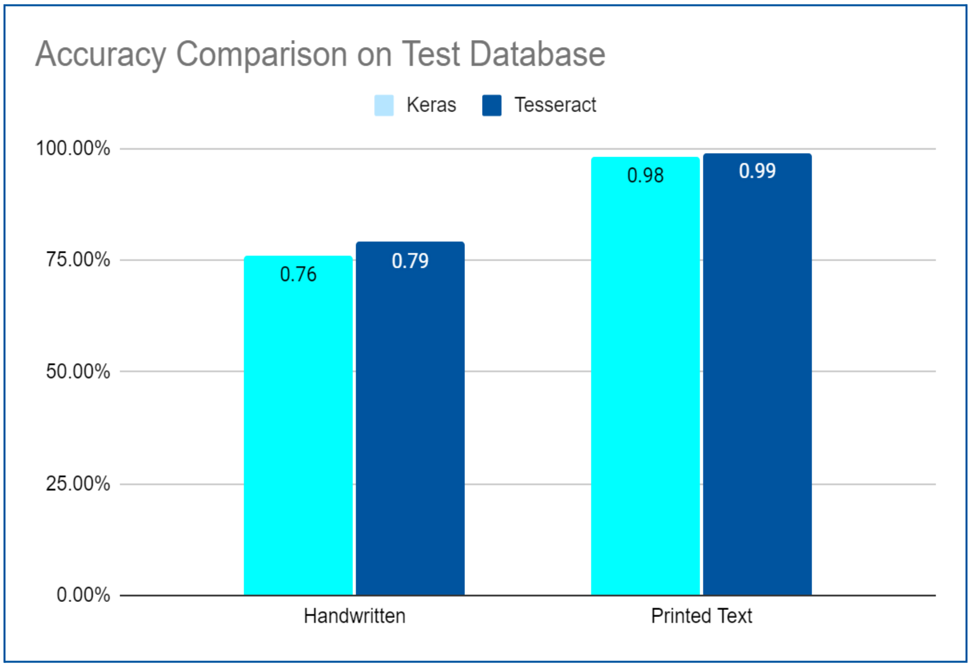
As we can see from this figure, this is what all of the points look like when they are matched. Note that as I previously stated, Python does not use all of these matched points, only the top 25 matches. After we complete this process we are then able to crop out specific segments of the form to prepare them for the OCR softwares. Using OpenCV, we use image preprocessing on all of the necessary information. The goal of implementing preprocessing techniques is to improve the accuracy of the OCR softwares. The preprocessing techniques that we will be using are image gray-scaling, enhancement, and thresholding.



This is an example of what a patient name would look like before and after image preprocessing.

Project Outcomes

To extract the text off of the preprocessed images, we used two different OCR softwares. The first is Tesseract, which uses a pre-trained OCR model that was developed by Google. The second of the two softwares is called Keras OCR, which is a pre-trained OCR model that was developed by Tensorflow. The reasoning for including both of these models was that the accuracy comparison between the two was negligible. It also gives us an understanding of what is considered to be hard-to-read text, as if both of the models had trouble deciphering the data, then we can deduce that it is not an edge case for the specific OCR software, but rather a limitation on OCR as a whole.



This figure displays the accuracy comparison between Keras and Tesseract.

The last step in this process was an automated data analysis which analyzes extracted patient data to generate insightful recommendations and strategies for Inogen in Excel Macro will flag incorrect inputs such as dates, names, incorrect addresses. This also automates data modeling, visualization, and predictive analytics with patient datasets. The Excel macro will generate histograms of various patients over time with their flow setting, a forecast model to predict what flow setting patients will require, and a scatter plot to determine trends within patients.

Individual Contributions

I was primarily responsible for the bulk of the coding solution for this project. This included developing the python script for morphing scanned prescription forms, image preprocessing on cropped fields, and running the optical character recognition softwares on our extracted data. Additionally, I created the sample patient information that was used to fill out the forms in the database. Lastly, I was responsible for giving updates on the progress of the technical aspect of the project to our corporate partners as well as our UCSB capstone mentor.