Medical Forms OCR

GitHub repository link: https://github.com/Kawaeee/medical_forms_ocr/

Introduction

Each person has own handwriting style that unique and different. Normally, we can read other people's handwritten text easily. But when we look at doctor handwriting style, we realized that a doctor's handwritten text is the most difficult thing to understand. So, we would like to develop a web platform that can recognize handwritten text and transform all of them into digital form using Optical Character Recognition (OCR).

Purpose

The purpose of this project was to solve the pain point. Since Ramathibodi Hospital they have a pain point in recognizing handwritten text from a doctor and their registration section seems to be inefficient. By their existing workflow is shown in Figure 1, we can see nurse need to work as a messenger to request form from registration section and hand in the form to doctor, then wait for doctor's diagnosis result and translate doctor handwritten text to easier to understand and go back to registration section to update patient data. So, we proposed to apply the OCR platform to serve better workflow as shown in Figure 2. We tried to reduce workload and get rid of unnecessary tasks to help nurses can spend more time with their patients and main tasks.

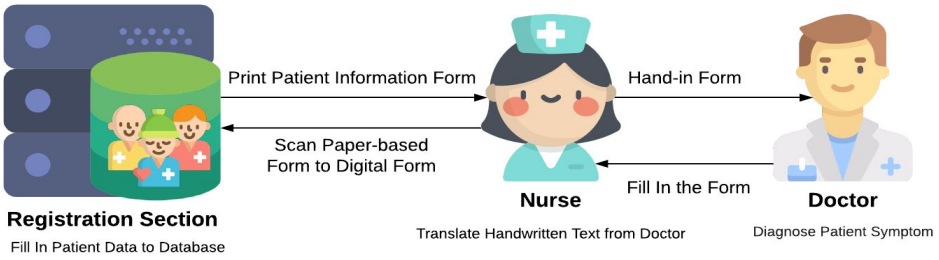

Figure 1: Existing Workflow

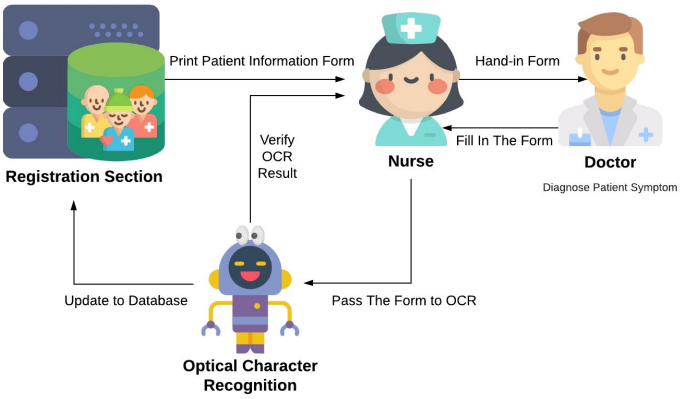

Figure 2: Proposed Workflow

Methodology

We divide the tasks into two sub-tasks which are OCR and web platform. First, we work on OCR by tried to locate all of the handwritten text bounding in Infliximab form since we know the medical form will not rapidly change every day. Then, we tried to use different API on the form such as Google Cloud Vision AI, Microsoft Azure Computer Vision, PyTesseract, and also tried on free OCR online web application to see how API performs. Then we chose PyTesseract since it can customize the model we run in each bounding box. In addition, it is free and we can train the tesseract model by ourselves.

Algorithm

We use Optical Character Recognition (OCR) to transform a two-dimensional image of text that contains digital or handwritten text from the image into machine-readable text.

Techniques

We use these three main libraries to develop this project which are:

PyTesseract (Python-tesseract) is a wrapper for Google's Tesseract-OCR which is an optical character recognition (OCR) tool for python that has the ability to recognize and read the embedded text in images. We use it to recognized handwritten text and digits.

Pillow is the Python Imaging Library that easily uses to manipulate images and apply image processing. We use it to locate and crop the entire bounding box in the form.

Flask is a micro web framework python library that wraps the Werkzeug toolkit and Jinja2 template engine. We use it as a front-end web platform that allows a nurse to upload a file and running OCR directly from the web.

Model

We use the default model from Google Tesseract to recognize handwritten text and use a pre-trained digit model from "tessdata_shreetest" GitHub repository.

Problem Encounter

After we tried a different type of OCR platform, we found that most of them still cannot be able to efficiently recognize handwritten text. Even though we use a pre-trained model on PyTesseract. The result of the model still far from human recognition.

Limitations

Currently, this OCR web platform is available for Infliximab form only. The pre-trained OCR model performs poorly with our medical form. In addition, it requires locating the bounding box manually in each form.

Future Improvement

In order to improve the OCR platform, we can improve it by developing our own tesseract model that fits with our medical form data to achieve a more accurate result. Also, we can develop an automatic update feature which adds OCR result to the new form and updates data to the registration section (database) directly.

References

A comprehensive guide to OCR with Tesseract, OpenCV and Python
https://nanonets.com/blog/ocr-with-tesseract/

Pre-trained tesseract model: digits1.traineddata
https://github.com/Shreeshrii/tessdata_shreetest/

PyTesseract: https://pypi.org/project/pytesseract/

Microsoft Azure Computer Vision:
https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/

Google Cloud Vision AI: https://cloud.google.com/vision/

Google Tesseract: https://opensource.google/projects/tesseract

Flask: https://pypi.org/project/Flask/

Pillow: https://pypi.org/project/Pillow/