Mahidol University

Department of Biomedical Engineering, Faculty of Engineering

Optical Character Recognition Project Report

EGBI443 Image Processing

1st Semester of Academic year 2023

Submitted to

Assoc. Prof. Dr. Panrasee Ritthipravat

Members

6313407 Anunda Chuenlerssakul

6313413 Rinrada Chongsomsuk

6313419 Thanawut Timpitak

6313423 Chonthicha Thipkaew

6313473 Napasara Asawalertsak

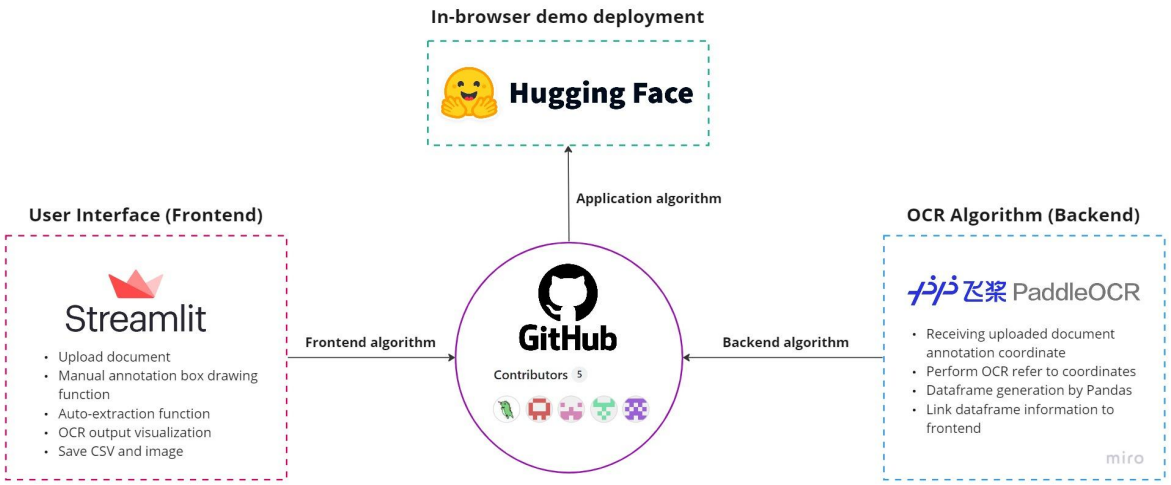
Date of Submission

15th December 2023

Table of Contents

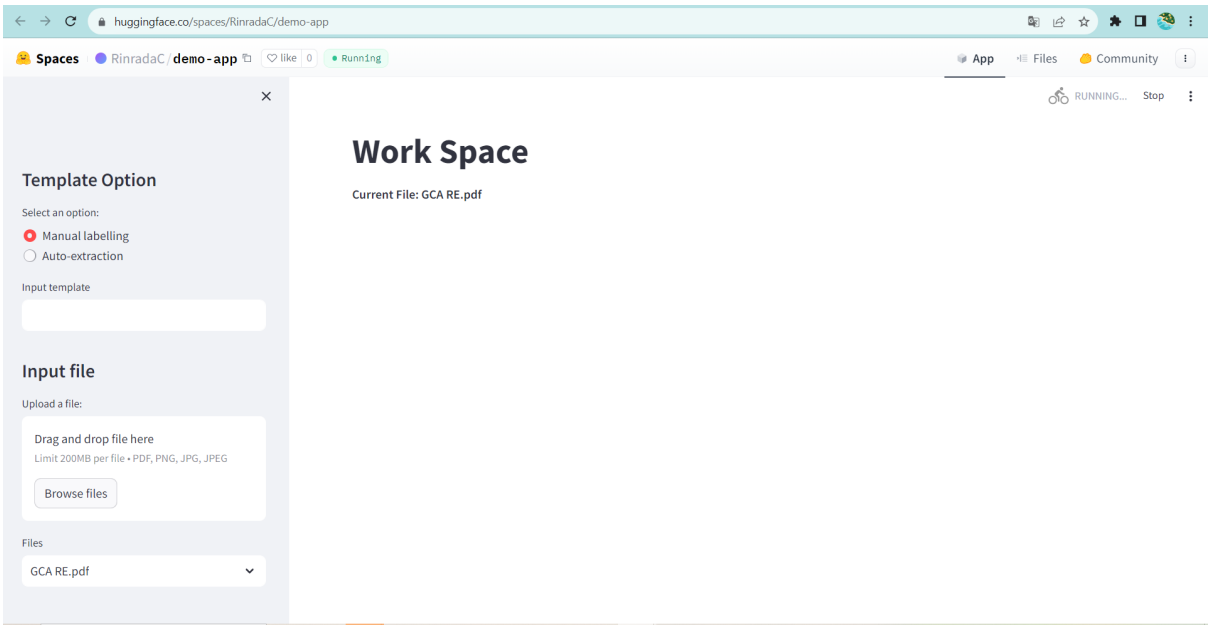
Introduction

Optical Character Recognition (OCR) and image processing technologies together have become a powerful force for revolutionary solutions in today's information technology environment. The goal of this dynamic project, Image Processing OCR, is to extract meaningful textual material from photographs by utilizing the combination of machine learning approaches and sophisticated algorithms. As digital data is increasingly incorporated in visual representations in this day and age, the project tackles the critical need of precisely and quickly transforming unstructured visual data into accessible, structured data. By combining state-of-the-art OCR techniques, multilingual support, and real-time processing capabilities, this project aims to redefine the standards for text extraction accuracy and provide a flexible tool that can be used for a variety of tasks, such as automated data entry, mobile scanning, and document digitization. The overall objective is to further the development of OCR and support its essential role in improving our interaction with visual information in a variety of scenarios as we delve into the nuances of picture preprocessing and investigate pathways for smooth integration with current systems. The overall system of this project utilize the use of Streamlit package to create the user interface and using the paddle OCR library then using the GitHub as a path to integrating this two section together, lastly the operating program was deploy on the on-browser website, Hugging Face to eliminate the requirement of library and package installation in local computer.



User Manual

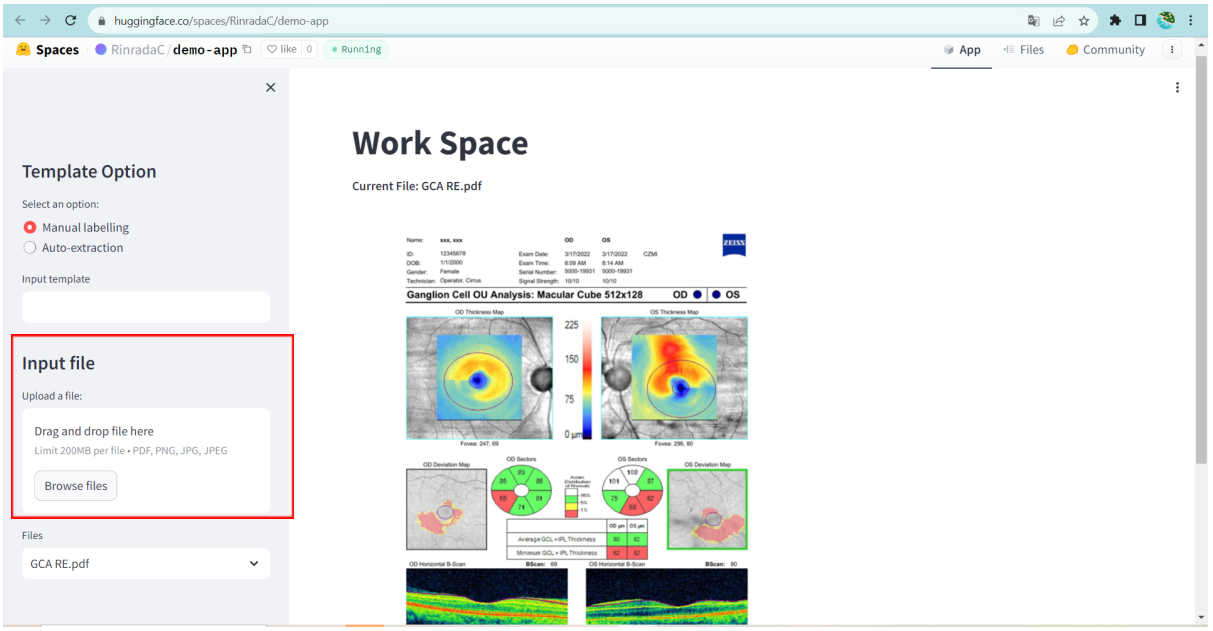
1. Click the link (https://huggingface.co/spaces/RinradaC/demo-app) to access the online application.



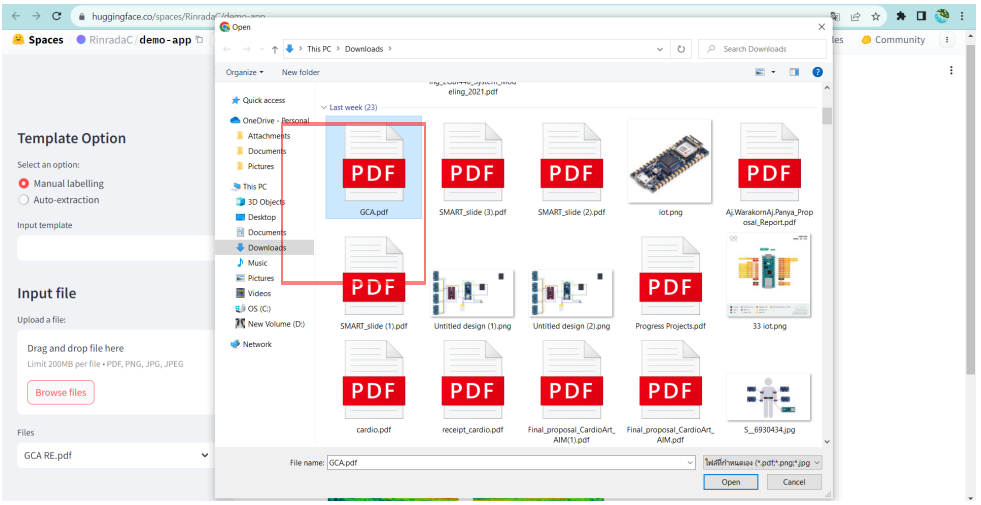
2. Select option:

   2.1 Manual labeling

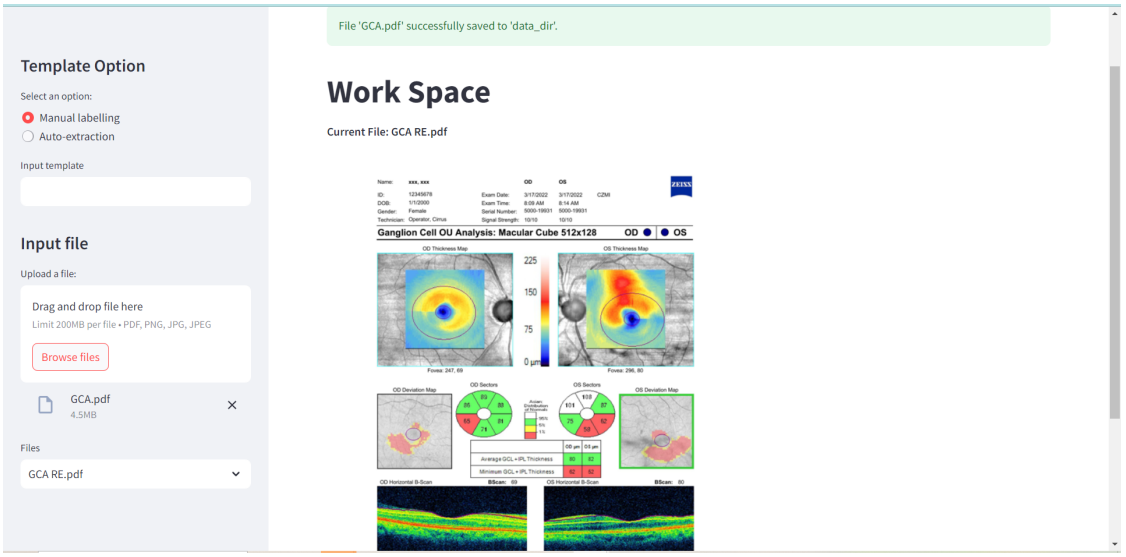
   - Upload file (PDF, PNG, JPG, JPEG)



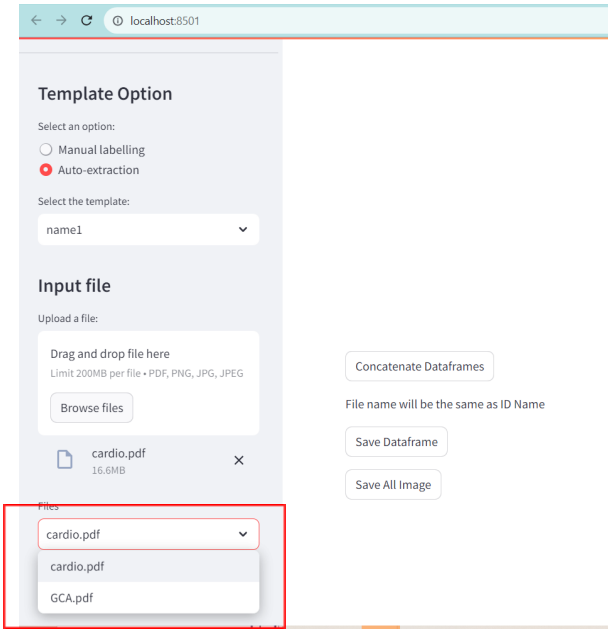
- Select file from the directory lists, the file displayed.



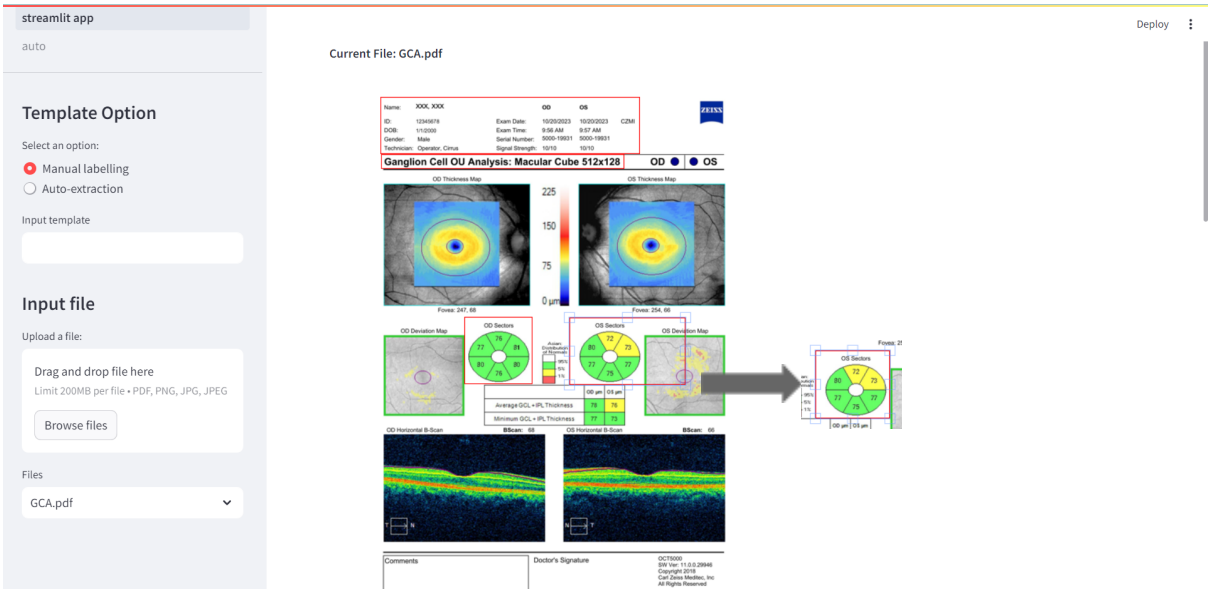
- After the file is uploaded then it will be displayed one by one, go to the next page by clicking the “Next Page” button.



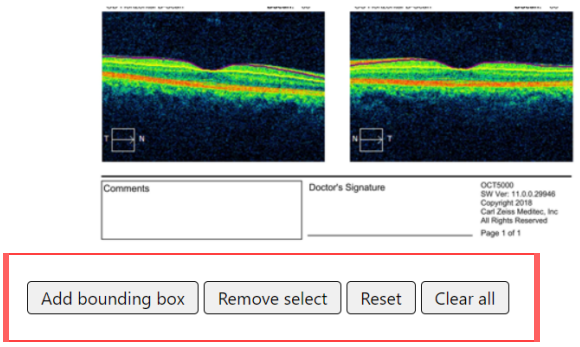
- In cases where the user uploads multiple files, on the sidebar, the user is able to choose the file that decides to perform OCR.



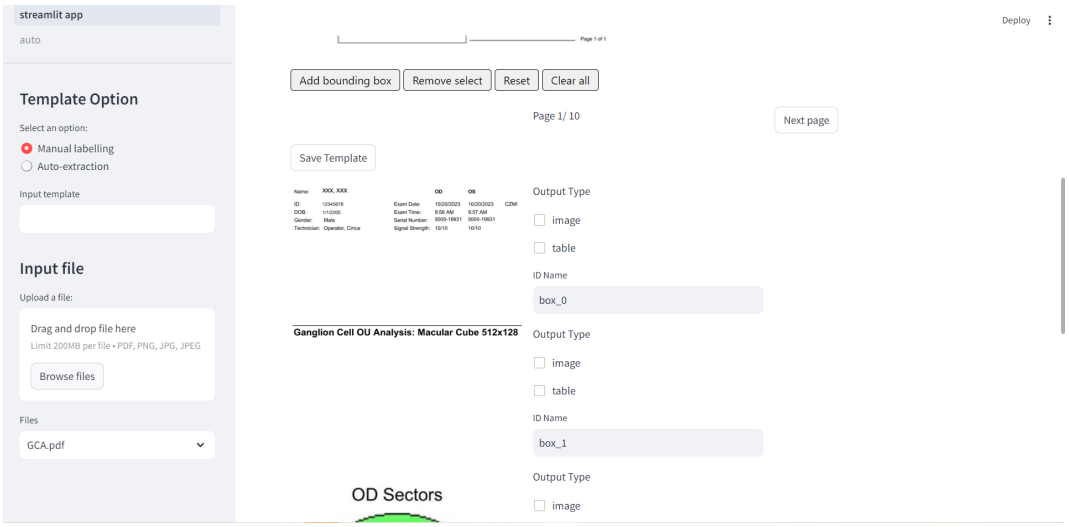
- Crop or select data that is needed by using the “Add bounding box” , which the bounding box can be adjusted manually.



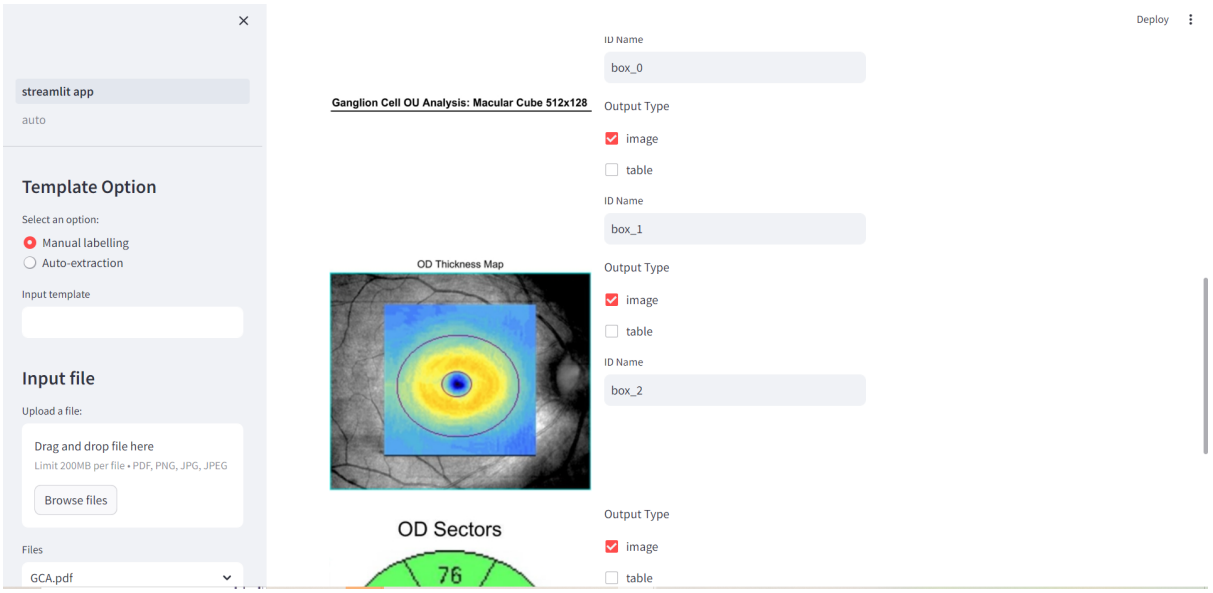
- Others command are provided to conduct; remove select, reset, clear all

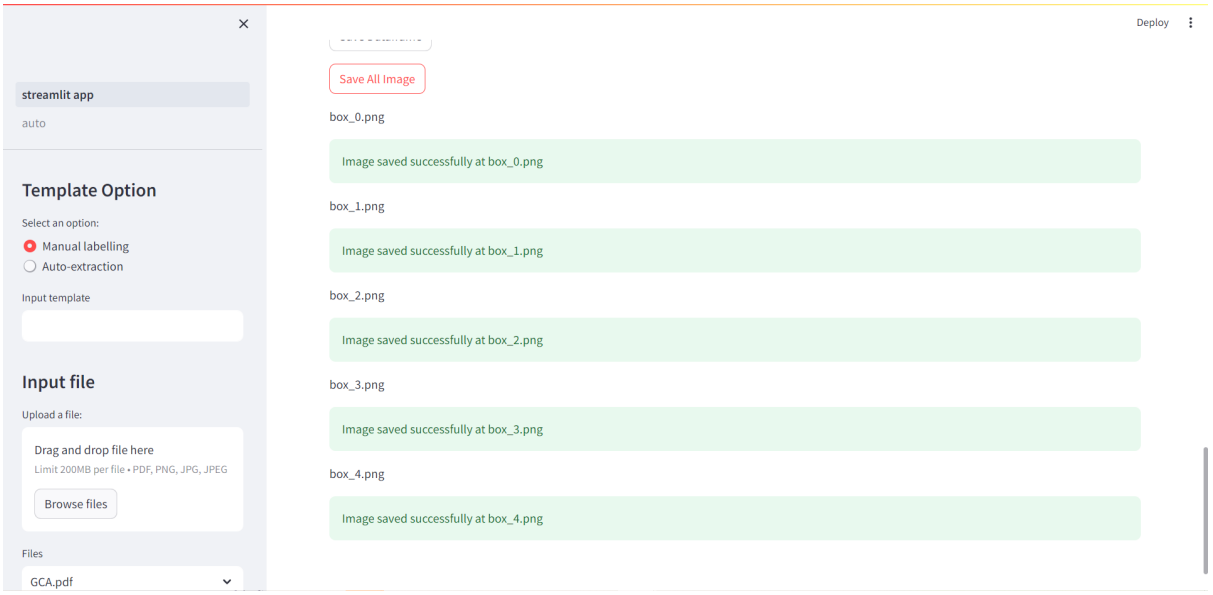


- After positioned the bounding box, the selected data will be displayed below with the dataframe (process of performing OCR)



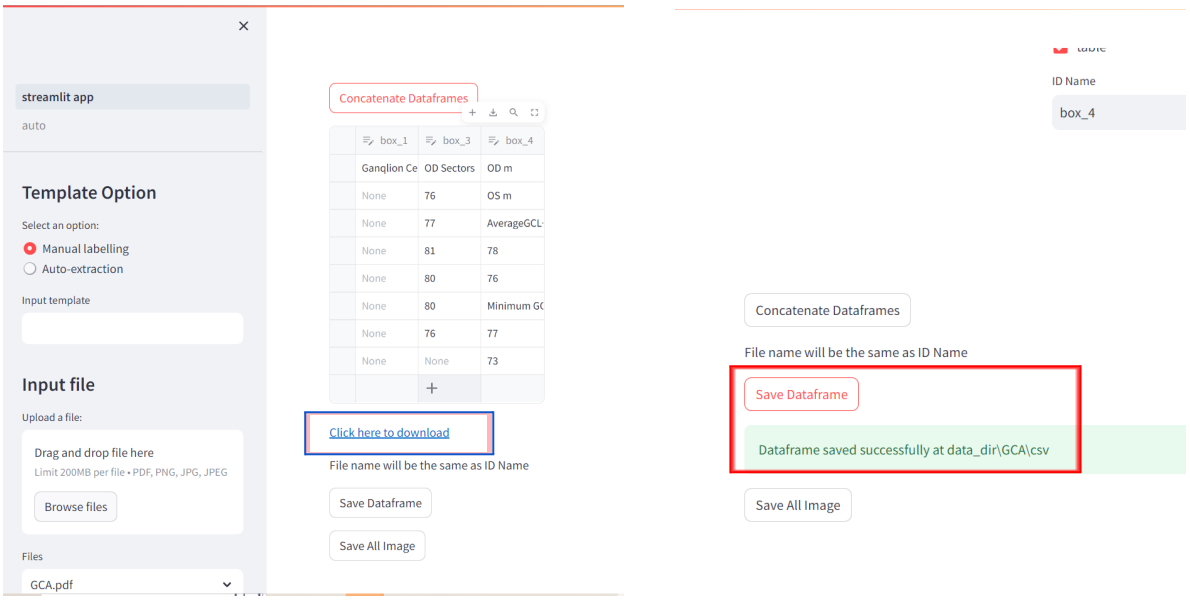
- Users enable to choose either image or table or both
    - If the user wants to save an image, click the image checkbox to be able to access the “Save All Image” button, then the image/images will be saved to the direct directory.



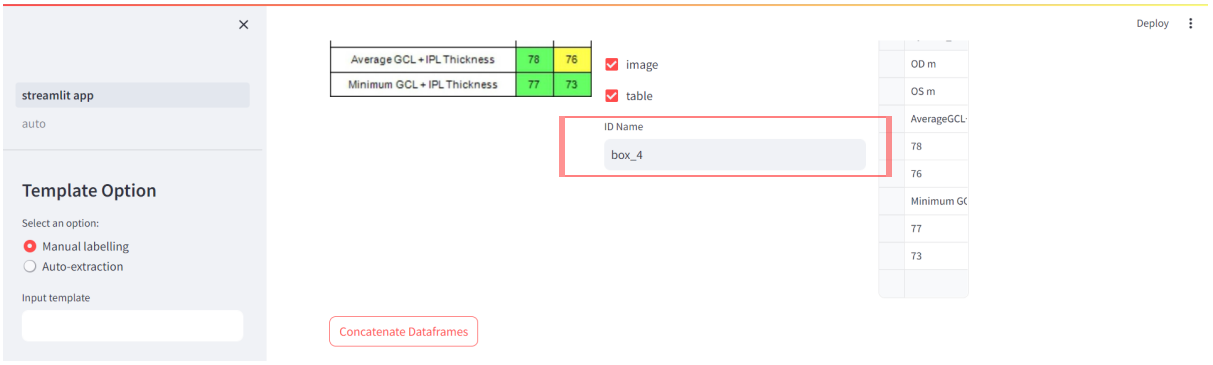


    - For table checkbox, after clicking the data frame will be shown and allow the user to edit the data frame. If there are multiple data frames, at the bottom all data frames will be concatenated and a saving .csv file is available.
        - Dataframe can be save for two methods which are the first one, after click “Concatenate Dataframes” button the concatenated dataframes is the displayed, the user can download the concatenated dataframes will be saved to downloads as the csv.

file type. And if the user clicks on the “Save Dataframe” button, the file will be saved as the default to data_dir\GCA\csv.



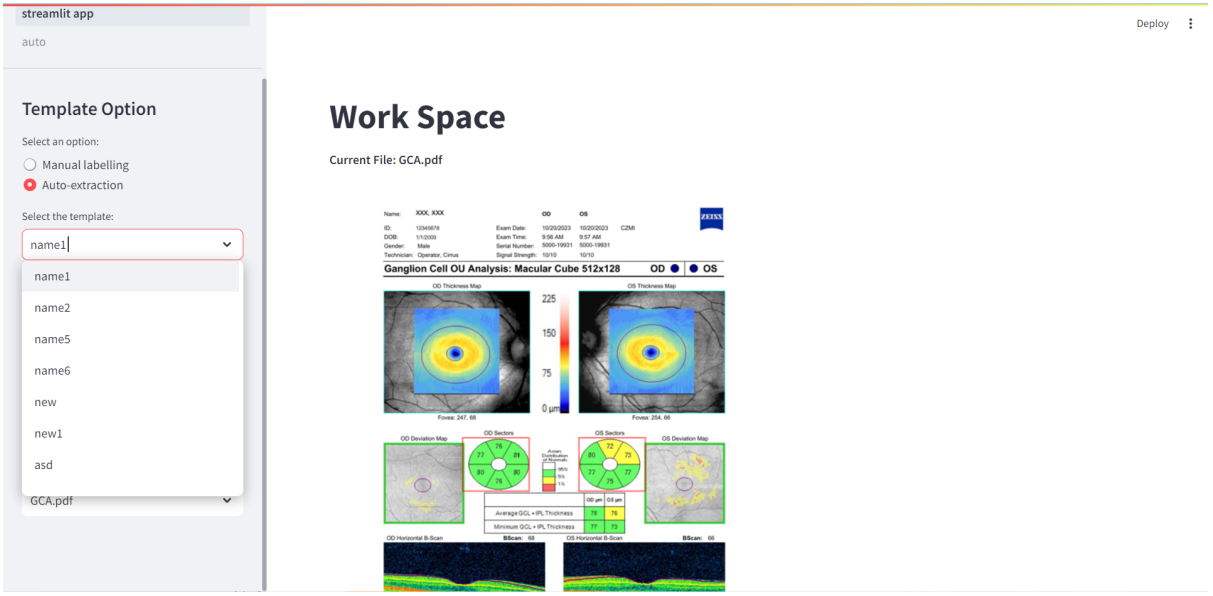
- Name the data frame by input text in the “ID Name” box.



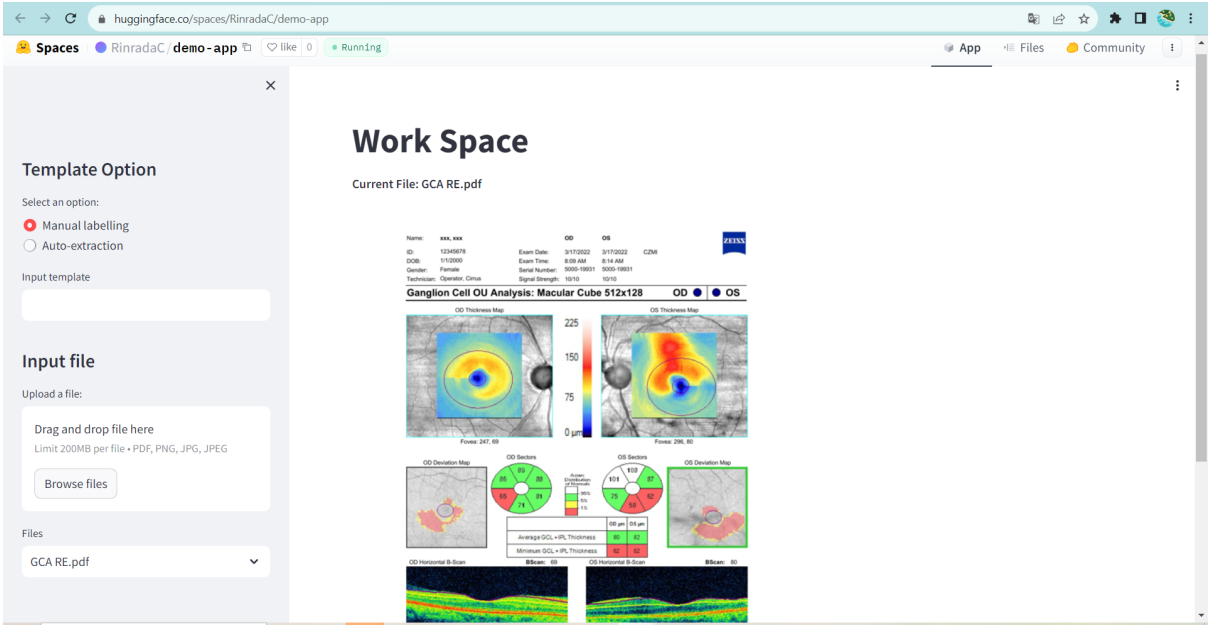
- Users are also able to create their own template by using the “Add bounding box” to crop the selected data and save it as a template by entering name then enter to conduct. By this all the saved templates would be used in the auto-extraction option.

2.2 Auto-extraction

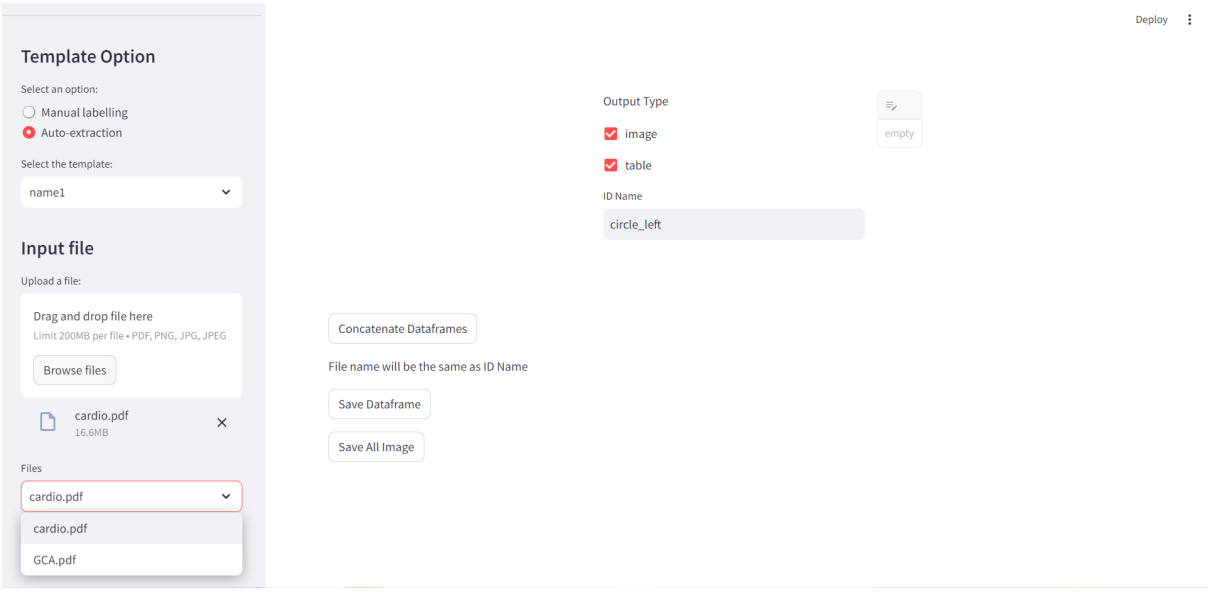
- Select the template for Auto-extraction from the Manual labeling process.



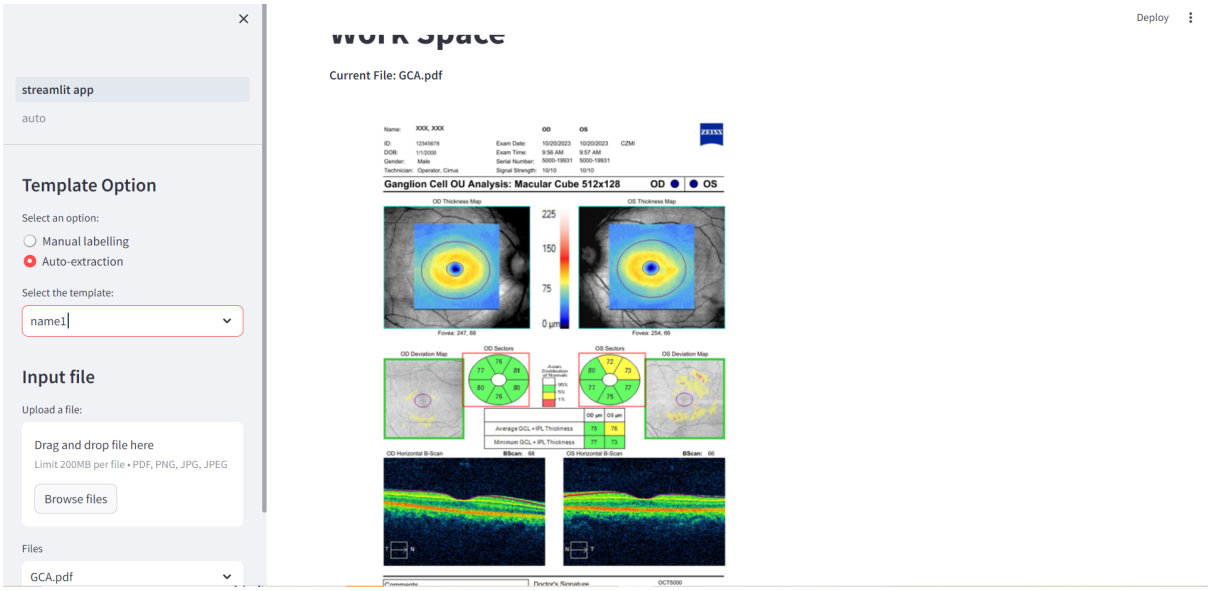
- Upload file (PDF, PNG, JPG, JPEG)



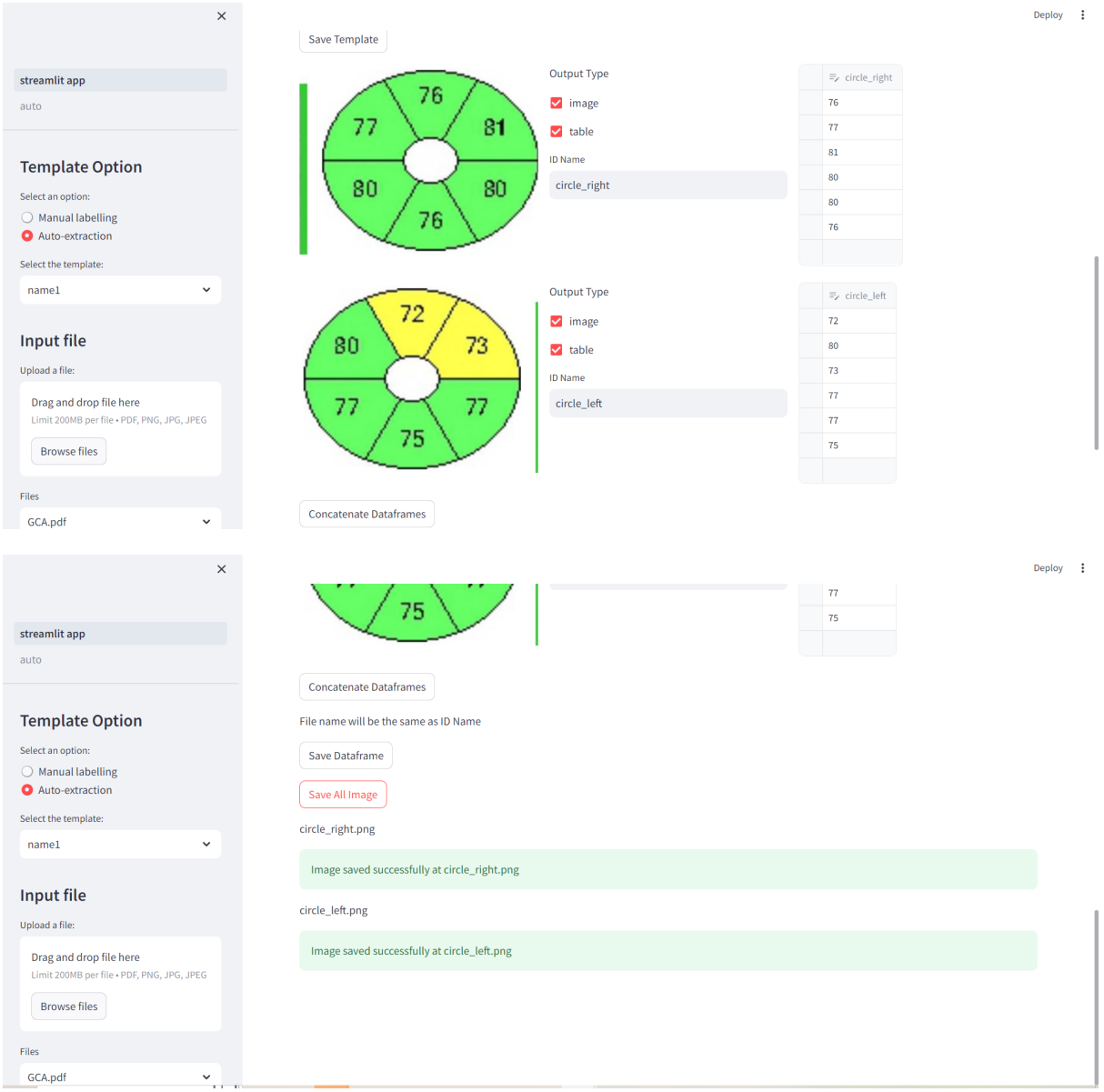
- In cases where the user uploads multiple files, on the sidebar, the user is able to choose the file that decides to perform OCR.



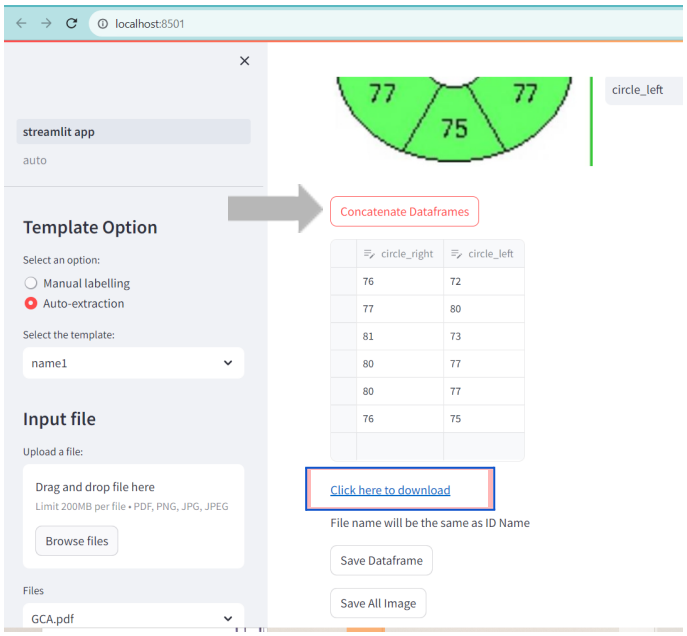
- The bounding boxes are positioned on the specific area as the template, manually adding the bounding box is not needed.

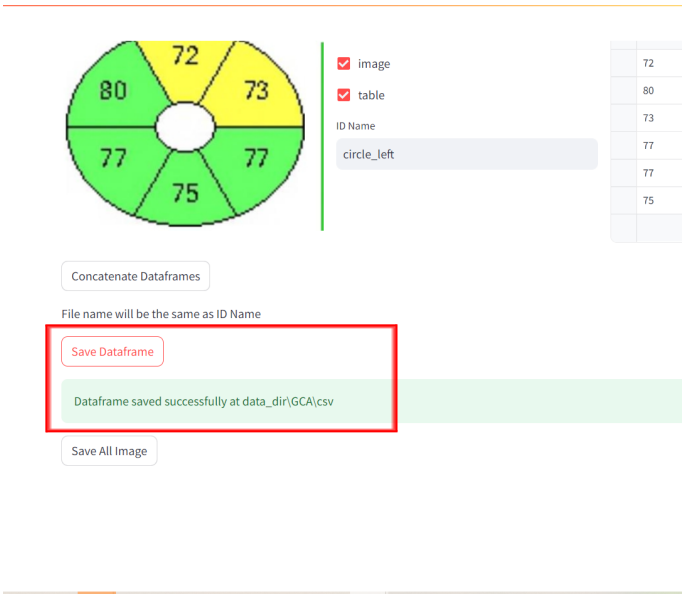


- Users enable to choose either image or table or both
    - If the user wants to save an image, click the image checkbox to be able to access the “Save All Image” button, then the image/images will be saved to the direct directory.



- For table checkbox, after clicking the data frame will be shown and allow the user to edit the data frame. If there are multiple data frames, at the bottom all data frames will be concatenated and a saving .csv file is available.
  - Dataframe can be save for two methods which are the first one, after click “Concatenate Dataframes” button the concatenated dataframes is the displayed, the user can download the concatenated dataframes will be saved to downloads as the csv. file type. And if the user clicks on the “Save Dataframe” button, the file will be saved as the default to data_dir\GCA\csv.


localhost:8501
streamlit app
auto
Template Option
Select an option:
Manual labelling
Auto-extraction
Select the template:
name1
Input file
Upload a file:
Drag and drop file here
Limit 200MB per file • PDF, PNG, JPG, JPEG
Browse files
Files
GCA.pdf
77
77
75
circle_left
Concatenate Dataframes
circle_right
circle_left
76
72
77
80
81
73
80
77
80
77
76
75
Click here to download
File name will be the same as ID Name
Save Dataframe
Save All Image


72
80
73
77
77
75
image
table
ID Name
circle_left
Concatenate Dataframes
File name will be the same as ID Name
Save Dataframe
Dataframe saved successfully at data_dir\GCA\csv
Save All Image

Problems and solutions

Optical Character Recognition (OCR) algorithm

Initially, OCR algorithm wasn't able to recognize the context of the annotated word and image, meaning that the header for each information is not able to be automatically defined and extracted orderly as it should be sequentially arranged to refer to the template of the document, this leads to the difficulties of dataframe pattern design later on. For the solution, the improvement of algorithms and more complex systems were developed to enhance the understanding of OCR algorithms and the ability on header recognition was obtained. For the optimization of the algorithm to resolve the error, most of the major functions that are significant for the software such as uploading and annotation are solved while problems on minor functions such as the annotation box that doesn't have the limit with the image boundaries are still existing.

User interface (UI) design

User interface design requires consideration in the aspect of user experience, deployment, and availability of the platform function to create the interface function for performing the OCR based on various provided templates of ophthalmology documents. As a consequence, the design based on those concept prioritization is raised and comes up with brainstorming sessions to clarify the interface design and separate the task for each other for laters integration. The main problem concerns in the aspect of UI design will be discussed in this section and the solution regarding to the problem will be provided :

- **User expereince**

  As the main user of the OCR software will be the physician, the UI purpose hence to provide the most simple yet not complicated and functional for the doctor to perform the prompt OCR on the document. As a finalize of the concept discussion, the decision on using the Streamlit platform was selected and the main function which is considered major function such as manual labeling, annotation box drawing, template saving, and auto-extraction refer from the template created in the manual labeling along with the dataframe visualization were then divided into different task and merging the function together after the algorithm was finished.

- **Deployment**

  Streamlit, a free-open framework for data and interface visualization based on Python and capability for integration with javascript, was selected for the front-end or user interface platform. It has a limitation to implement and be capable with all computers of the user as users need to install all libraries and packages that are required in order to run the software. As a consequence, Hugging Face platform was used to deploy the software as it is considered as an interactive web-browsing that allows the authorized user to access and use the software without the needs of manual libraries and packages installation.

Integration of front-end and back end session

The complete OCR software for ophthalmology documents requires a combination of UI from front-end and OCR algorithm from back-end session, hence, the iteration of implementing to define the error and bugging of the software is frequently required. As the different units using the different libraries, the installation is then compulsory to be installed and in some situations, there was the problem with library compatibility with used programming language versions like Python. To resolve this problem, our team uses Github as a hub for uploading the updated algorithm with the comments on the update to inform the changes to other contributors which make the workflow become more efficient.

Future Development

1. Integrating the GPU acceleration will be a significant factor to improve the computational performance and lowering the computation time.
2. Deploy the software on the private cloud for data storage which will be beneficial for the hospital usage as the patient's credential data will be kept in a private ecosystem.
3. User interface and Optical Character Recognition (OCR) algorithm could be further improved in the aspect of better user-friendly and higher accuracy after receiving the feedback from the real user or physician.