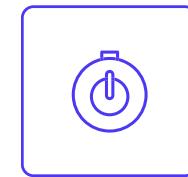


DAO-OCR Flask Web App

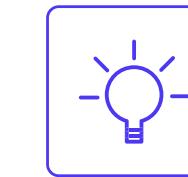
Optical Character Recognition - Using Tesseract-OCR



Table of Contents



1
INTRODUCTION



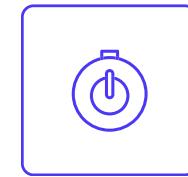
2
PROBLEM
STATEMENT



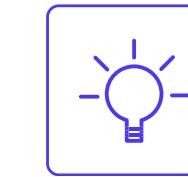
3
MOTIVATION



4
RESEARCH
QUESTIONS



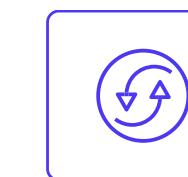
5
TOOLS AND
TECHNOLOGIES



6
PROPOSED
APPROACH



7
CHALLENGES



8
CONCLUSION



Our Team



Danso Daniel

ddanso020@st.ug.edu.gh

Aboagye Seth

skaboagye008@st.ug.edu.gh

Okai Raphael

rookai@st.ug.edu.gh

INTRODUCTION

In today's digital age, we are inundated with data in various forms, including images, PDFs, and scanned documents. Extracting text data from these sources can be a tedious and time-consuming task. This is where Optical Character Recognition (OCR) technology comes into play.



INTRODUCTION

The goal of OCR is to convert these data sources into a digital format that can be easily processed and analyzed.

The DAO-OCR project aims to create a Flask web app that integrates Tesseract OCR to extract text from image files accurately. This article will delve into the problem statement, motivation, research questions, tools and technologies used, proposed approach, and challenges faced by the DAO-OCR project.





Part 3:
Our Key Strategies

PROBLEM STATEMENT



PROBLEM STATEMENT

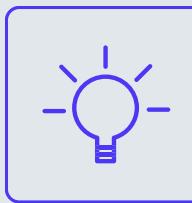
Extracting text data from images, PDFs, and scanned documents is time-consuming and error-prone.



Need for an accurate and reliable Optical Character Recognition (OCR) system to convert data sources into a digital format that can be easily processed and analyzed.



The goal of this project is to create a Flask web app that can integrate Tesseract OCR to extract text from image files accurately.



Part 3:
Our Key Strategies

MOTIVATION

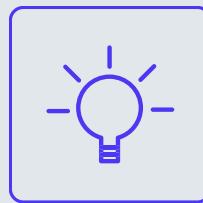


MOTIVATION

Eliminate the need for manual transcription of text from images and improve the efficiency of data extraction.

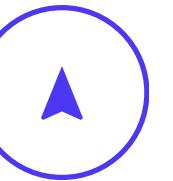


Provide a user-friendly web app that can accurately extract text from images.



Part 3:
Our Key Strategies

RESEARCH QUESTIONS



RESEARCH QUESTIONS

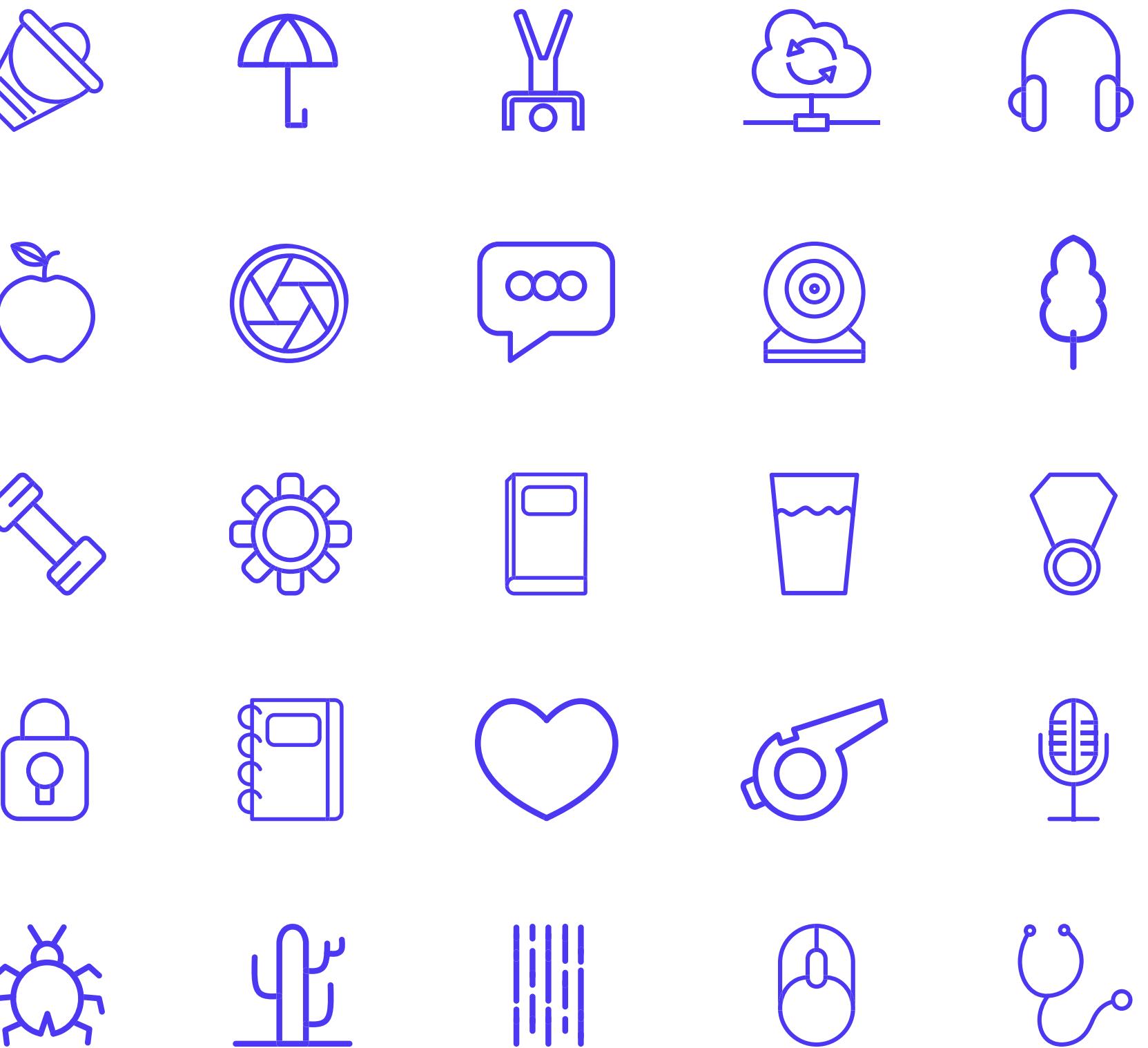
How accurate is the Tesseract OCR system in extracting text from various image file formats?

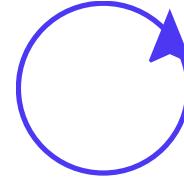
How can we integrate Tesseract OCR into a Flask web app for text extraction from images?

Can we give accurate text extraction?

What is the best way to display the extracted text in a readable format on the web app?

TOOLS AND TECHNOLOGIES



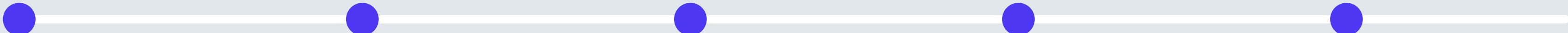


Available on:

<https://dao-ocr-production.up.railway.app/>

TOOLS AND TECHNOLOGIES

For the implementation of DAO-OCR, we will use the following tools and technologies:



01

Python

02

Flask

03

Flask-Scss

04

HTML/CSS/
Javascript

05

Tesseract-
OCR

06

Pytesseract

07

Rails

08

Docker

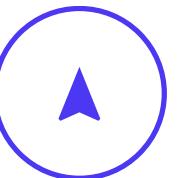
09

Bootstrap

10

Git/GitHub

PROPOSED APPROACH



PROPOSED APPROACH

Test the web app with various image file formats to ensure accuracy.

Develop a Flask web app that can accept image files, extract text using Tesseract-OCR, and display the extracted text in a readable format.



Integrate Pytesseract into the Flask web app to extract text from images.



Use HTML/ CSS/ Javascript to create a user-friendly interface for image upload and text display.



CHALLENGES



CHALLENGES

Ensuring the accuracy of the text extraction from images, especially with low-quality images.

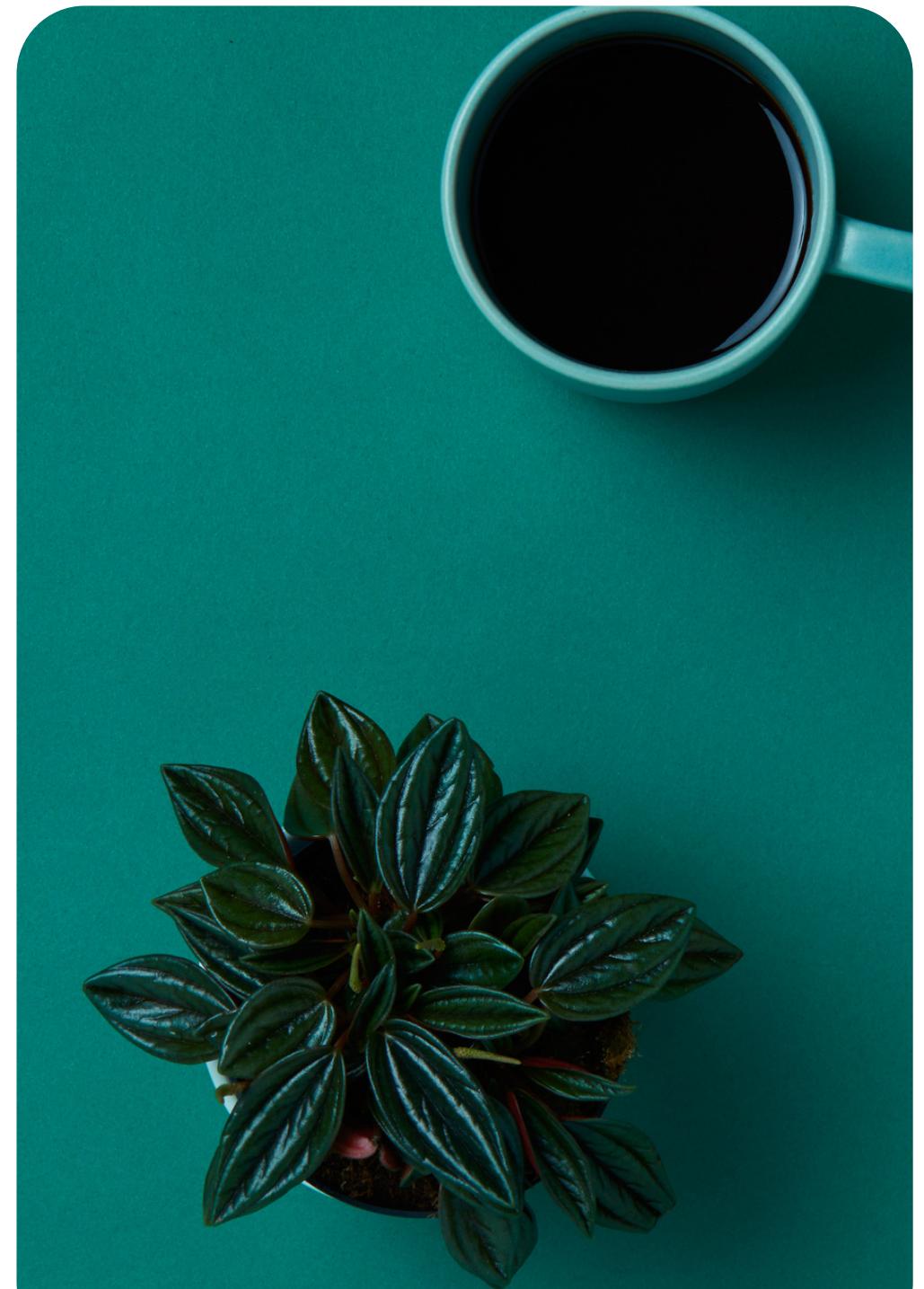


Developing a user-friendly interface that is easy to navigate and use.



Integrating the OCR system into the Flask web app while ensuring efficient and reliable functionality.





CONCLUSION

In conclusion, the DAO-OCR project aims to make text extraction from images more accessible and efficient for users. By leveraging Tesseract OCR and Flask web app development, we hope to provide a reliable and user-friendly tool for digitizing data from image files.



Thank you!