College of Applied Business

(Tribhuvan University)

A REPORT ON PYTHON'S PROJECT

Optical Character Reader

(OCR)

Submitted by:
Kashyap Raj Karki
Mekhraj Gurung
Amrit Kumar Thapa
Suman Gurung
BSc.CSIT (2nd Semester)
Faculty: Science and Technology

Submitted to:
Kamal Acharya
Date of Submission: 2021/04/20

ACKNOWLEDGEMENT

"It's not possible to prepare a project report without the assistance and encouragement of other people. This one is certainly no exception."

On the very outset of this report, we would like to extend our sincere and heartfelt obligation towards all the personages who have helped us in this endeavor. Without their active guidance, help, cooperation and encouragement, we would not have made headway in this project.

We are ineffably indebted to Mr. Kamal Acharya, our python instructor, for conscientious guidance and encouragement to accomplish this project.

We extend our gratitude towards College of Applied Business for giving us this opportunity.

We also acknowledge a deep sense of reverence, our gratitude towards our parents and family members of our family, who have helped us morally.

At last but not the least, gratitude goes to all of our friends who directly or indirectly helped us to complete this project report.

Any omission in this brief acknowledgement does not mean lack of gratitude.

ABSTRACT

The project proposal was given to create an application using python. So we have created an “OCR”.

OCR stands for optical character reader that converts text in an image to text. We thought of making it for the purpose of digitally storing notes and submitting it as we were in need of it at the time of lockdown.

We used Google’s tessatact engine for making this OCR. This engine was developed for the purpose of making an OCR. If the image has a certain level of clarity and is written in universally accepted English fonts than the program can convert an image to string. This is the core concept of an image OCR.

Table of Contents

Chapter One

Project Introduction

Optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example: from a television broadcast).

Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Among the various application of OCR, we are doing text reorganization for now. Our code is able to convert and image with text to string format and also has capability of extracting other languages as well.

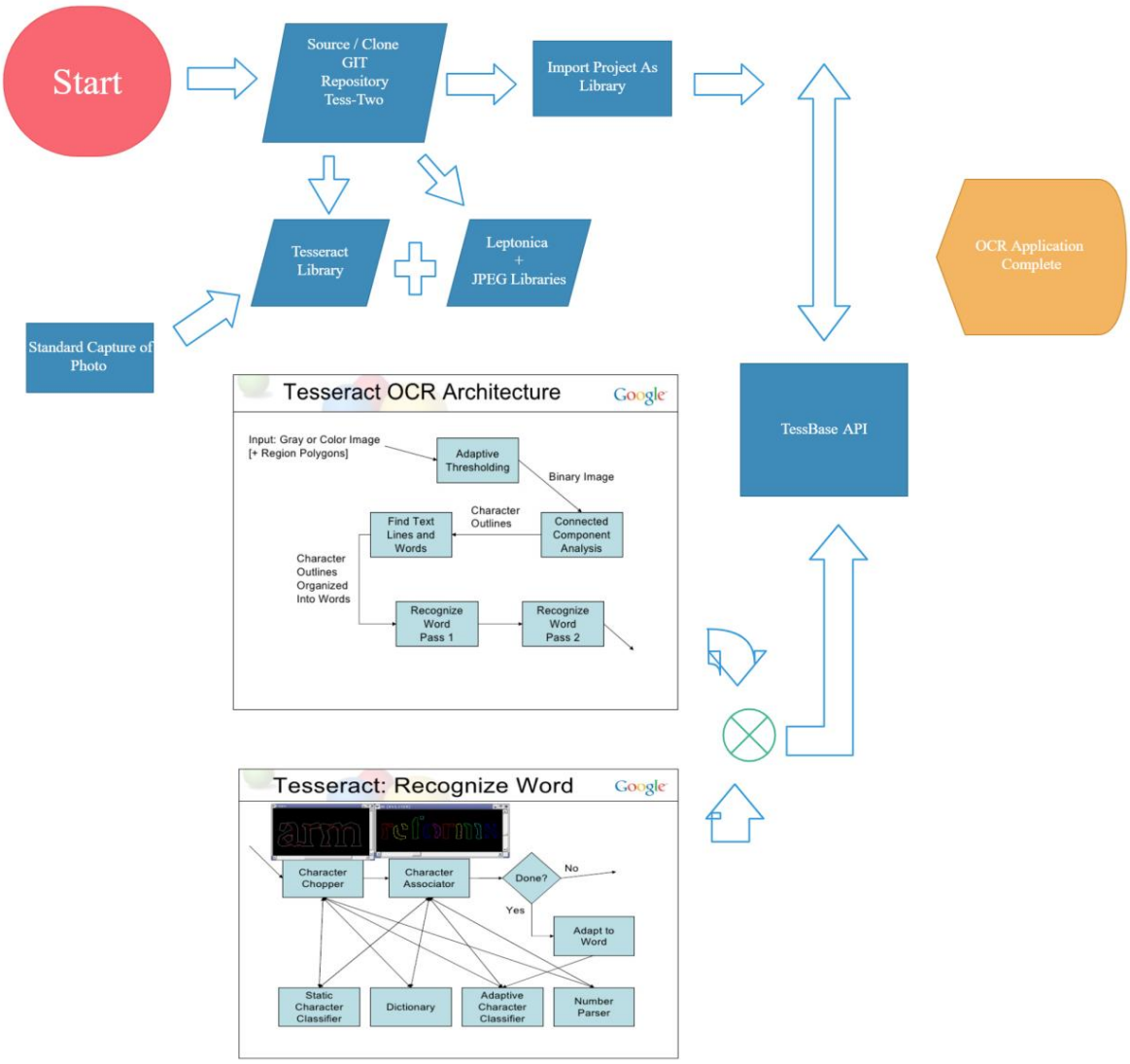
Tools used

We selected **python** as our programming language for this project as we were studying python and this is a part of our python's project. Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. We selected python because of its simplified syntax and with an emphasis on natural language. Also, for every task it has a build in module which made our task a lot easier.

We used tessatact engine, pytesseract module to bind python and tessetact engine, openCV module for image processing as well as numpy. module.

Chapter Two

Flow chart



Chapter Three

Our program tries its best to give an output with maximum clarity. The result may vary upon clarity of an image, the fonts and language used in an image, the background of the image etc.

Code:

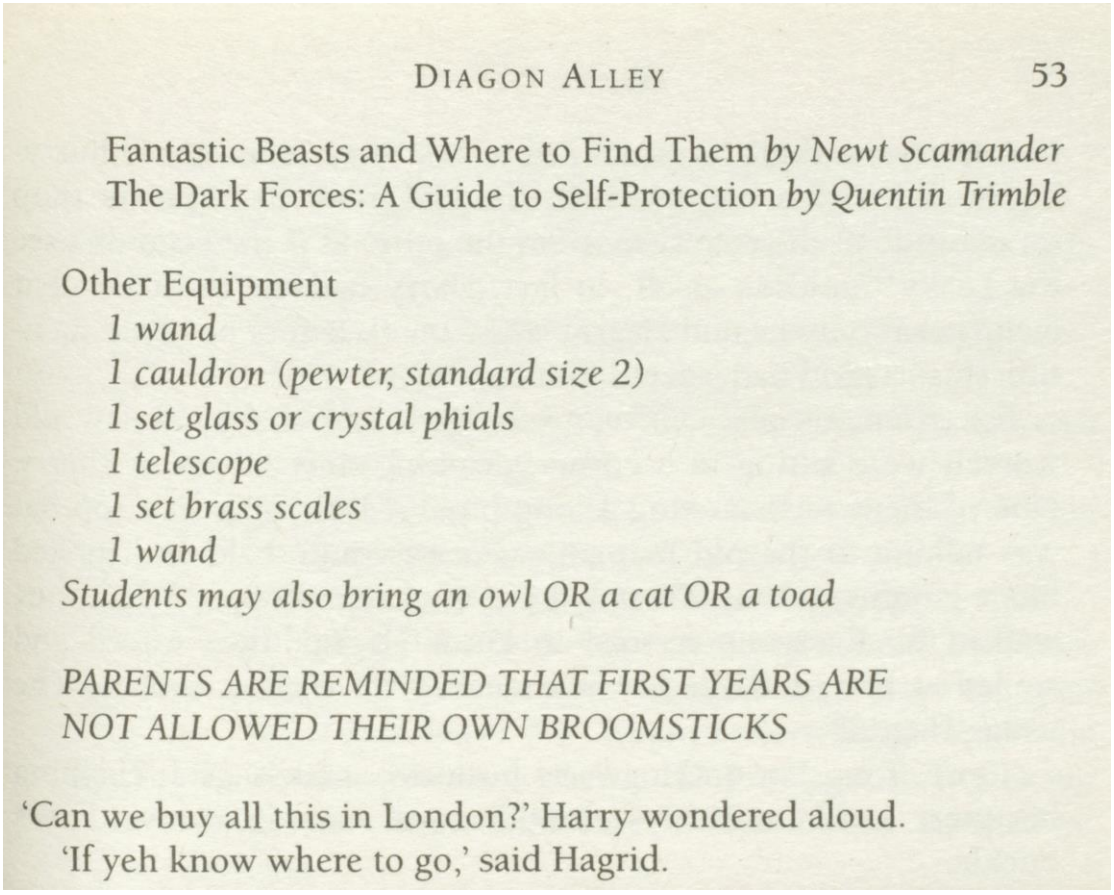
```
import cv2
import numpy as np
import pytesseract

pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-
OCR\tesseract.exe"

img = cv2.imread("jk.jpg")
img = cv2.resize(img, None, fx=0.2,fy=0.2)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

adaptive_threshold = cv2.adaptiveThreshold(gray, 225,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,85,11)

config = "--psm 4"
text = pytesseract.image_to_string(adaptive_threshold,config=config,
lang='eng')
print(text)

cv2.imshow("img",img)
cv2.imshow("gray",gray)
cv2.imshow('adaptive th',adaptive_threshold)
cv2.waitKey(0)
```

Output:

```
main
C:\Users\you00\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/you00/PycharmProjects/pythonProject/main.py
DiaGon ALLEY 53

Fantastic Beasts and Where to Find Them by Newt Scamander
The Dark Forces: A Guide to Self-Protection by Quentin Trimble

Other Equipment
1 wand
1 cauldron (pewter, standard size 2)
I set glass or crystal phials
I telescope
I set brass scales
I wand
Students may also bring an owl OR acat OR a toad

PARENTS ARE REMINDED THAT FIRST YEARS ARE
NOT ALLOWED THEIR OWN BROOMSTICKS

'Can we buy all this in London?' Harry wondered aloud.
'If yeh know where to go,' said Hagrid.
```

Steps:

Real-image:



DIAGON ALLEY 53

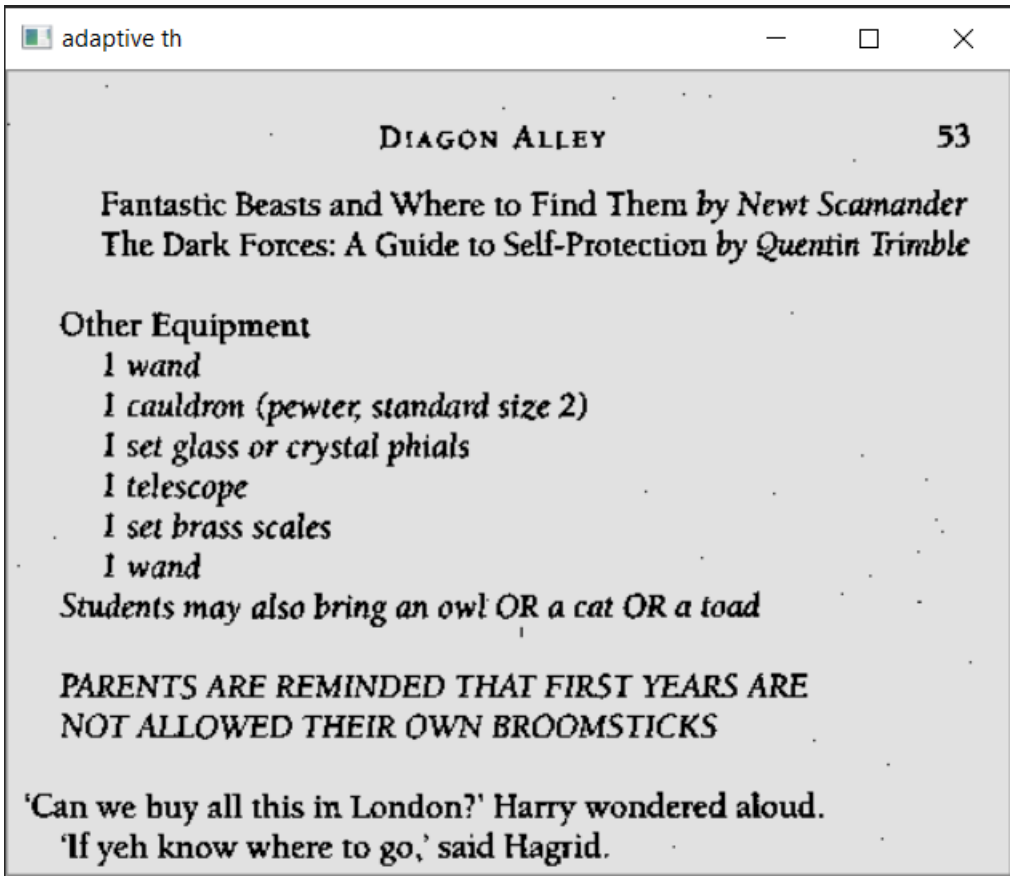
Fantastic Beasts and Where to Find Them *by Newt Scamander*
The Dark Forces: A Guide to Self-Protection *by Quentin Trimble*

Other Equipment
1 wand
1 cauldron (pewter, standard size 2)
1 set glass or crystal phials
1 telescope
1 set brass scales
1 wand
Students may also bring an owl OR a cat OR a toad

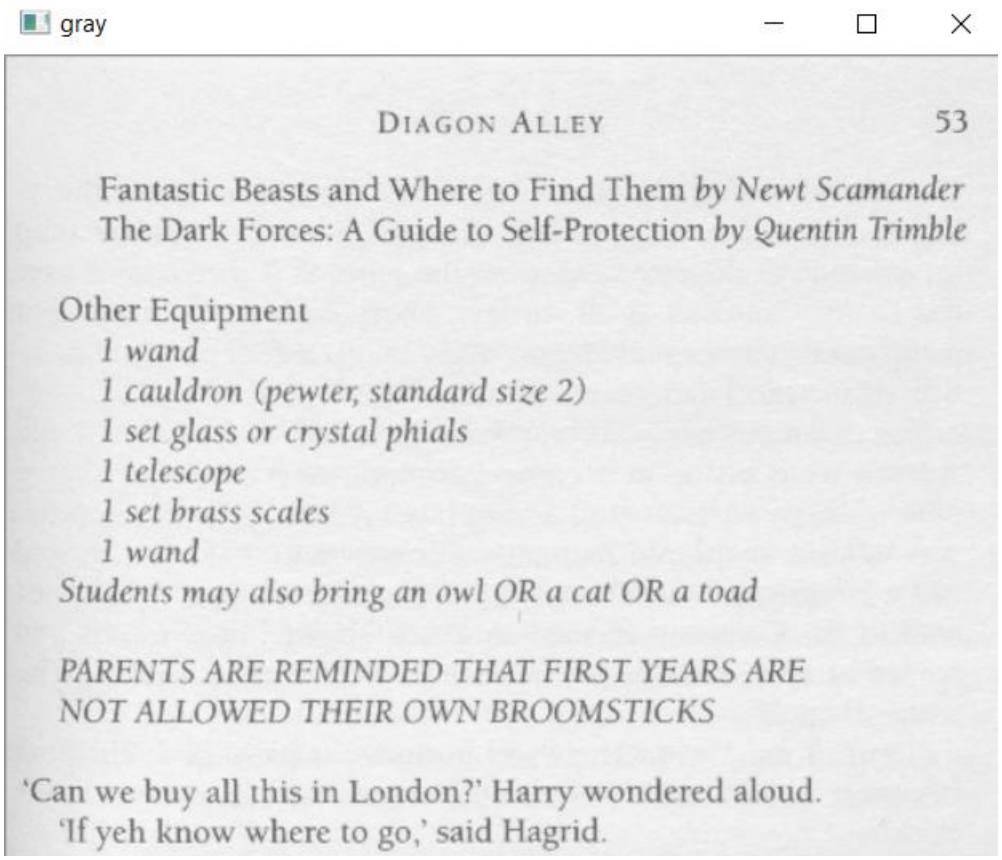
PARENTS ARE REMINDED THAT FIRST YEARS ARE
NOT ALLOWED THEIR OWN BROOMSTICKS

'Can we buy all this in London?' Harry wondered aloud.
'If yeh know where to go,' said Hagrid.

Adaptive-threshold image:-



Grey scale image:-



Summary of the Output:-

Here we see the code try to convert an image of a Harry Potter novel to string. We see it try its best to obtain correct string. We displayed the output with the adaptive threshold image which failed to distinguish between an ‘I’ and a ‘1’ as it looked similar. However, the grey scale image clearly distinguishes between a ‘1’ and an ‘I’. Optimization of these simple yet crucial mistakes is what we look forward to correct in the future.

CONCLUSION

We learned about python and some python module while we were working on this project. We learned to manipulate data by importing then form external source.

The ocr's source code is fairly simple but constructing a GUI for it was not so much of an easy task. The GUI is still a work in progress as the KIWI mode is a bit on the difficult side and tkinter model didn't support android phones. Another problem that arose was to find the correct was to edit an image so that it becomes easy for the program to recognize the text easily. We tried converting image into gray scale and resizing it but if the lighting was in sufficient the text would not be recognized. Ultimately we went with erasing the background as it gave the clearest text reorganization and maximum percentage of output with minimum error.

This program is not a perfect and definitely has room for improvement. This prototype program lacks functioning GUI that will be completed at the final submission. There are many places where improvements can be made. For instance, we can also translate the language of the text as well. For example, if the extracted text is in Japanese than we can convert it into English aswell.

This project taught us a lot about application development using python and Google's tessatact engine. The database is new and in improving in daily basis. These projects also made us better understand python and improved our coding skills. It was an awesome experience creating a working python application with daily use capability.