Optical Character Recognition (OCR) using OpenCV and Tesseract in Python

Sungkyeong (Lucia) Jeon

2020 WAY (Winter At Yonsei)
AI & Design
Prof. Jin Kook Lee

Contents

Topic: Study-planner App Development

General Scheme

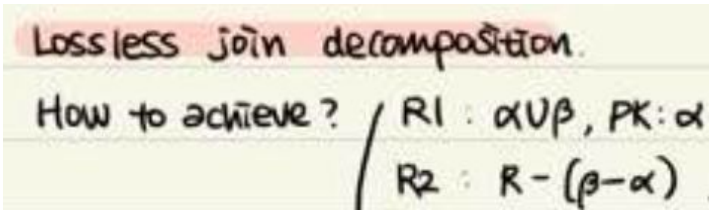
- compares two schedules: before-plan, after-plan
- plan input using hand-written data (specially targeted for iPad users)
- shows the efficiency as a percentage (i.e. You achieved 75% of the plans today!)
- store the personalized data (how long can this person concentrate, how many different subjects worked best, personalized tips for study efficiency improvement)
- modules needed: time management module / app UI module / calculation (AID) module
- further improvement ideas
    - Google calendar-like interface
    - add a timer functionality for each plan so that the user can take advantage of it

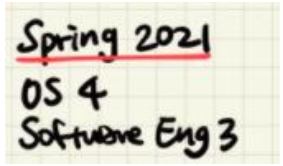
For this project,

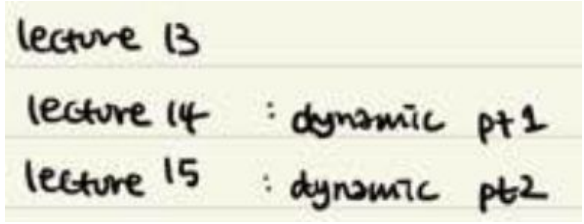
- implementation of AID module
- takes in start_time, end_time as float type and file_name as str type
- use text recognition (OCR) to convert image file to a string
- calculate and evaluate the efficiency of the day

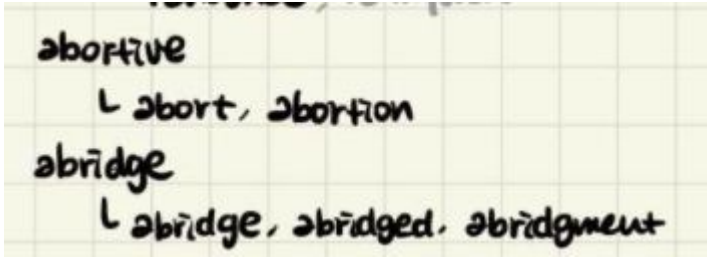
Hand-written Data for Testing

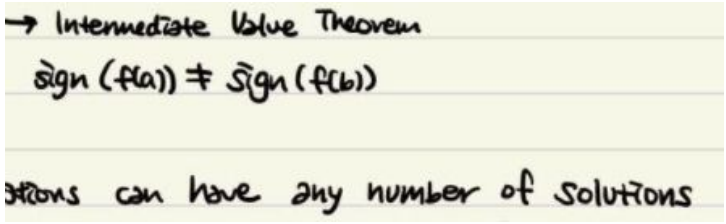
Own handwritten notes from iPad GoodNotes



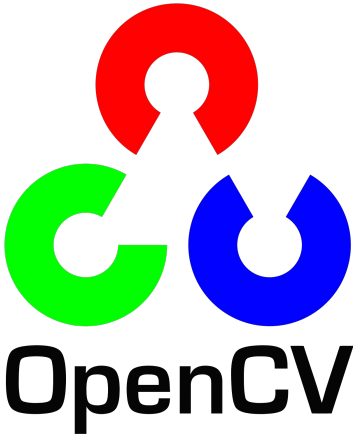








OpenCV and Pytesseract



- Open source computer vision
- supports python library (import cv2)
- imread(filename), cvtColor(), imshow(file_directory) etc

Tesseract OCR

- developed by Google since 2006
- contains a command line program
- supports more than 100 languages
- Pytesseract builds on top of Tesseract

```python
# Lucia Jeon
# AID Final Project
# 01/17/2021

import os
import cv2
import pytesseract
from pytesseract import image_to_string

class Plan:
    def __init__(self, start_time, end_time, name):
        # start_time (float), end_time(float), name(str)
        self.start_time = start_time
        self.end_time = end_time
        self.name = name

plan_count = int(input("How many plans do you have?: "))
plan_list = list()

for i in range(1, plan_count+1):
    start_time = float(input(str(i)+"- start time: "))
    end_time = float(input(str(i)+"- end time: "))
    file_name = input(str(i)+"- image file name: ")
    img=cv2.imread(file_name)
    name = image_to_string(img)
    print(name)

    # create a Schedule class instance
    plan_list.append(Plan(start_time, end_time, name))
```

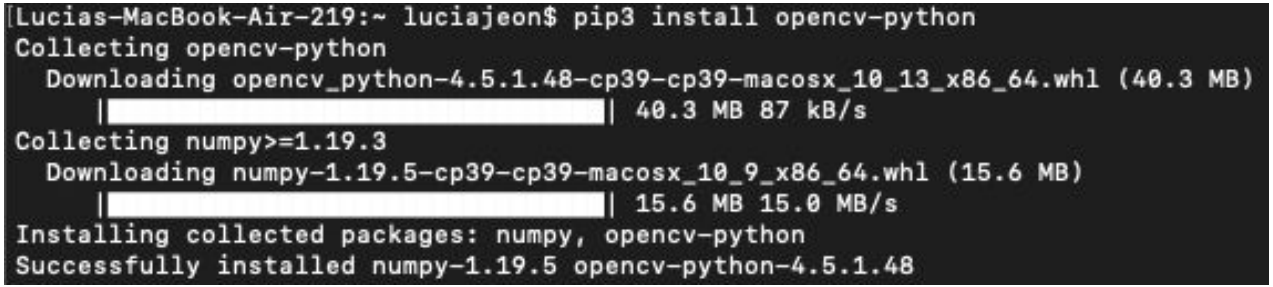
```python
# evaluate each plans
plan_efficiency = list()
for i in range(plan_count):
    actual_start_time = float(input(str(i+1)+"- actual start time: "))
    actual_end_time = float(input(str(i+1)+"- actual end time: "))
    efficiency = (actual_end_time - actual_start_time) / (plan_list[i].e
    print("plan", i+1, "efficiency:", efficiency, "%")
    plan_efficiency.append(efficiency)

print("average efficiency:", sum(plan_efficiency)/len(plan_efficiency))
```

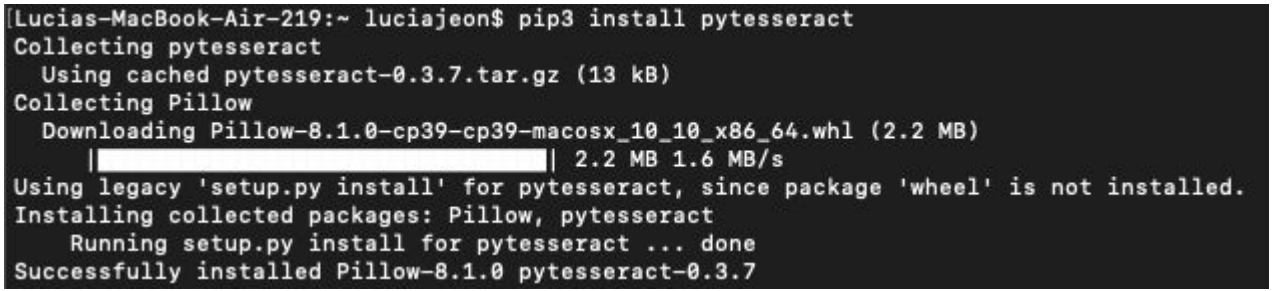
Source code snippets

Results

1) Download opencv-python on terminal using Homebrew



2) Download pytesseract on terminal using pip3 command



cf. both modules can be installed by either Homebrew or pip command

Results

```
==== RESTART: /Users/luciajeon/Desktop/proj
How many plans do you have?: 2
1- start time: 9
1- end time: 12
1- image file name: plan1.png
Stats chap.3
2- start time: 15
2- end time: 18
2- image file name: plan2.png
ML textbook pg.100-140
1- actual start time: 9
1- actual end time: 11.5
plan 1 efficiency: 83.33333333333334 %
2- actual start time: 15.5
2- actual end time: 17.5
plan 2 efficiency: 66.66666666666666 %
average efficiency: 75.0
```

plan1.png

plan2.png

References

- pyimagesearch “Using Tesseract OCR with Python” (https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/)
- StackOverflow: TesseractNotFoundError resolve,
- Homebrew: OpenCV and Tesseract Library Install (https://formulae.brew.sh/formula/tesseract)
- Tessdoc (Tesseract documentation) on Github (https://tesseract-ocr.github.io/tessdoc/Installation.html)
- Illinois Library “Introduction to OCR and Searchable PDFs” (https://guides.library.illinois.edu/c.php?g=347520&p=4121425)