디지털 영상처리

텀 프로젝트 2: Image To Text (문자 인식)

목표

■ 글자 인식

- 테서렉트를 이용하여 영상에서 글자를 추출하는 프로그램 구현


```
Hus

Dohyun Engineering & Construction Co.,Ltd

(쥐도현종합건설 |

a ST

건축사업부                    Tel. 064.759.0449
i                        Fax, 064.712.0449
ay 이호진             E-mail: Ihj5670@huangroup.co.kr

홈페이지: www. huangroup.co.kr
HP. 010.2739.5670          제주특별자치도 제주시 간월동로 72-7
```


```
(OpenCV) D:\Projects\Lecture_source>python ImageToText.py
ImageToText.py:43: DeprecationWarning: an integer is required (got type numpy.float32).
  result = cv2.warpPerspective(img, mtrx, (width, height))
| >

(주도현종합건설
```

배경색이 있는 경우

테서렉스(Tesseract)

■ 테서랙트는 Google의 다양한 운영 체제를 위한 광학 문자 인식 엔진

- 100개가 넘는 다양한 언어들을 인식할 수 있음
- 오픈소스이기 때문에 누구나 무료로 사용가능
- C/C++ 언어로 구성되어 있지만 PyTesseract 패키지를 이용하여 파이썬 환경에서도 사용할 수 있음

Tesseract OCR

테서렉스(Tesseract) 설치

■ 설치 링크 : https://github.com/UB-Mannheim/tesseract/wiki

Tesseract installer for Windows

Normally we run Tesseract on Debian GNU Linux, but there was also the need for a Windows version. That's why we have built a Tesseract installer for Windows.

WARNING: Tesseract should be either installed in the directory which is suggested during the installation or in a new directory. The uninstaller removes the whole installation directory. If you installed Tesseract in an existing directory, that directory will be removed with all its subdirectories and files.

The latest installers can be downloaded here:

- tesseract-ocr-w32-setup-v5.2.0.20220712.exe (32 bit) and
- tesseract-ocr-w64-setup-v5.2.0.20220712.exe (64 bit) resp.

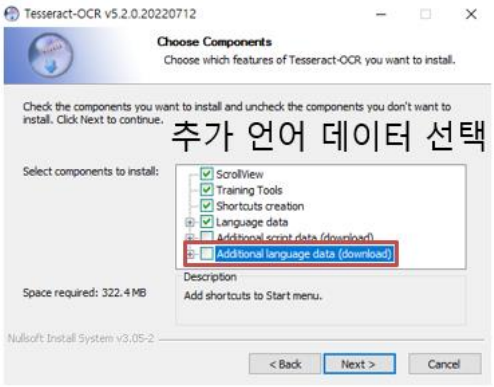
알맞은 윈도우 버전으로 설치

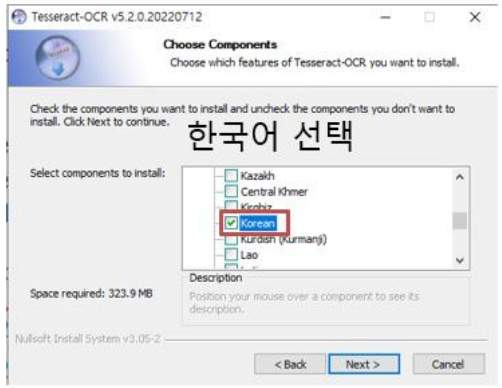
There are also older versions available.

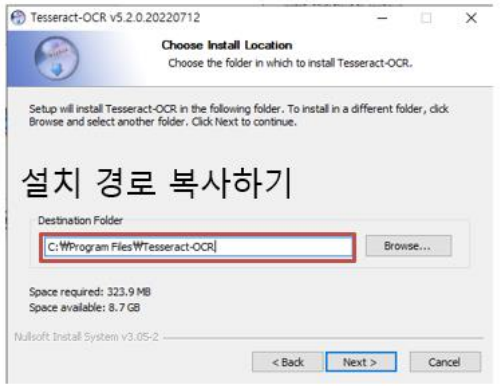
In addition, we also provide documentation which was generated by Doxygen.

테서렉스(Tesseract) 설치

■ 설치 링크 : https://github.com/UB-Mannheim/tesseract/wiki

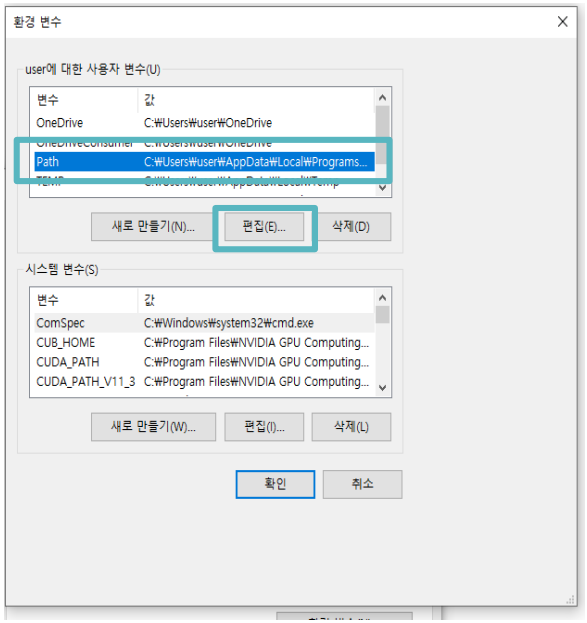


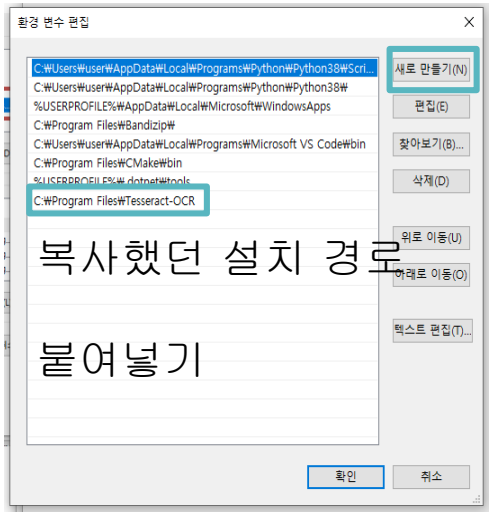




테서렉스(Tesseract) 설치

■ 설치 링크 : https://github.com/UB-Mannheim/tesseract/wiki





코드 실습

```python
import cv2
import numpy as np
import pytesseract

TESSERACT_PATH = "C:/Program Files/Tesseract-OCR/tesseract.exe"
imgpath='./imgs/2.jpg'
win_name = 'Image To Text'
img = cv2.imread(imgpath)


def GetOCR():
    #이미지 처리
    global img

    #OCR모델 불러오기
    pytesseract.pytesseract.tesseract_cmd = TESSERACT_PATH

    #OCR모델로 글자 추출
    text = pytesseract.image_to_string(img, lang='kor+eng')

    return text

cv2.imshow(win_name, img)
cv2.waitKey(0)
text = GetOCR()
print(text)
```

테서렉트 설치경로/tesseract.exe

```
PS D:\Projects\Lecture_source> pip install pytesseract
Collecting pytesseract
  Using cached pytesseract-0.3.10-py3-none-any.whl (14 kB)
Collecting packaging>=21.3
  Downloading packaging-22.0-py3-none-any.whl (42 kB)
     ---------------------------------------- 42.6/42.6 kB 2.2 MB/s eta 0:00:00
Collecting Pillow>=8.0.0
  Downloading Pillow-9.3.0-cp38-cp38-win_amd64.whl (2.5 MB)
     ---------------------------------------- 2.5/2.5 MB 22.5 MB/s eta 0:00:00
Installing collected packages: Pillow, packaging, pytesseract
```

파이썬 pip 명령어로 pytesseract 패키지 설치

```
PS D:\Projects\Lecture_source> pip install numpy
Collecting numpy
  Downloading numpy-1.23.5-cp38-cp38-win_amd64.whl (14.7 MB)
     ---------------------------------------- 14.7/14.7 MB 34.4 MB/s eta 0:00:00
Installing collected packages: numpy
```

파이썬 pip 명령어로 numpy 패키지 설치

코드 실행

■ 임의의 명함 영상을 이용해 인식해 봄

CMD 실행결과

```
(OpenCV) D:\Projects\Lecture_source>python ImageToText_copy.py

(OpenCV) D:\Projects\Lecture_source>
```

실행결과

글자를 제대로 인식하지 못하여

아무것도 출력되지 않거나

맞지 않는 글자들이 출력

구현 결과 소개

프로그램 구현 1

```python
draw = img.copy()
pts_cnt=0
pts=np.zeros((4, 2), dtype=np.float32)
```

전역 변수와 onMouse()함수 추가

```python
def onMouse(event, x, y, flags, param):
    global pts_cnt
    global img
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(draw, (x, y), 10, (0, 255, 0), -1)
        cv2.imshow(win_name, draw)

        pts[pts_cnt] = [x,y]
        pts_cnt += 1
        if pts_cnt == 4:
            sm = pts.sum(axis=1)
            diff= np.diff(pts, axis=1)

            topLeft = pts[np.argmin(sm)]
            bottomRight = pts[np.argmax(sm)]
            topRight = pts[np.argmin(diff)]
            bottomLeft = pts[np.argmax(diff)]

            pts1 = np.float32([topLeft, topRight, bottomRight, bottomLeft])

            w1 = abs(bottomRight[0] - bottomLeft[0])
            w2 = abs(topRight[0] - topLeft[0])
            h1 = abs(topRight[1] - bottomRight[1])
            h2 = abs(topLeft[1]-bottomLeft[1])
            width = max([w1, w2])
            height = max([h1, h2])

            pts2 = np.float32([[0,0], [width-1, 0], [width-1, height-1], [0, height-1]])

            mtrx=cv2.getPerspectiveTransform(pts1, pts2)
            result = cv2.warpPerspective(img, mtrx, (width, height))
            cv2.imshow('scanned', result)
            img=result
```

클릭할 때 마다 녹색 원 생성

녹색 원의 좌표(x, y) 획득

변환 전 좌표

녹색 원들의 좌표간 거리 획득

변환 후 이미지의 가로, 세로 크기

변환 후 좌표

변환 행렬 획득 후 원근 변환 적용

```python
cv2.imshow(win_name, img)
cv2.setMouseCallback(win_name, onMouse)
cv2.waitKey(0)
text = GetOCR()
print(text)
```

Cv2.setMouseCallback() 함수 실행

프로그램 구현 1

- **MouseCallback**
  - 마우스에 관한 이벤트가 발생하면, 콜백함수에 이벤트를 전달해 처리
- **Cv2.setMouseCallback(윈도우, 콜백함수, *사용자정의 데이터)**
  - 마우스 콜백을 설정하는 함수
  - 윈도우는 마우스 이벤트가 발생하는 창
  - 콜백함수는 마우스 이벤트 발생 시, 이벤트를 처리하는 함수
  - 사용자 정의 데이터는 이벤트 전달 시, 함께 전달할 사용자 정의 데이터(옵션)
- **콜백함수(event, x, y, flags, *param)**
  - 마우스 이벤트를 처리하는 함수
  - event는 전달 받은 마우스 이벤트
  - x, y는 마우스 이벤트 발생 시, 마우스의 좌표
  - Flags는 event의 상태를 나타내는 매개변수
  - param은 마우스 이벤트와 함께 전달받은 사용자 정의 데이터(옵션)

프로그램 구현 1

■ Mouse Events

EVENT	
EVENT_MOUSEMOVE	마우스 포인터가 움직일 때
EVENT_LBUTTONDOWN	마우스 왼쪽 버튼 누를 때
EVENT_MBUTTONDOWN	마우스 가운데 버튼 누를 때
EVENT_RBUTTONDOWN	마우스 오른쪽 버튼 누를 때
EVENT_LBUTTONUP	마우스 왼쪽 버튼 뗄 때
EVENT_MBUTTONUP	마우스 가운데 버튼 뗄 때
EVENT_RBUTTONUP	마우스 오른쪽 버튼 뗄 때
EVENT_LBUTTONDCLK	마우스 왼쪽 버튼 더블 클릭할 때
EVENT_MBUTTONDCLK	마우스 가운데 버튼 더블 클릭할 때
EVENT_RBUTTONDCLK	마우스 오른쪽 버튼 더블 클릭할 때
EVENT_MOUSEWHEEL	마우스 상하 스크롤 할 때
EVENT_MOUSEHWHEEL	마우스 좌우 스크롤 할 때

프로그램 구현 1

■ Mouse Event Flags

EVENT FLAGS	
EVENT_FLAG_LBUTTON	마우스 왼쪽 버튼이 눌러져 있음
EVENT_FLAG_MBUTTON	마우스 가운데 버튼이 눌러져 있음
EVENT_FLAG_RBUTTON	마우스 오른쪽 버튼이 눌러져 있음
EVENT_FALG_CTRLKEY	컨트롤 키가 눌러져 있음
EVENT_FLAG_SHIFTKEY	쉬프트 키가 눌러져 있음
EVENT_FLAG_ALTKEY	알트 키가 눌러져 있음

프로그램 구현 2

이미지의 4점을 이용한 Affine 변환

프로그램 구현 결과

■ 테서렉트로 텍스트 추출 결과


```
Hus

Dohyun Engineering & Construction Co.,Ltd

(쥐도현종합건설 |

a ST

건축사업부                    Tel. 064.759
i                          Fax, 064.712
ay 이호진                  E-mail:          @huangroup.co.kr

홈페이지: www. huangroup.co.kr
HP. 010.2739                  제주특별자치도 제주시 간월동로
```

프로그램 구현 결과

■ 배경과 글자가 불분명한 경우

- 이상하게 출력되거나 테서렉트가 글자를 인식 못함

CMD 실행결과

```
(OpenCV) D:\Projects\Lecture_source>python ImageToText_copy.py
ImageToText_copy.py:45: DeprecationWarning: an integer is required (got type numpy.float32).
  result = cv2.warpPerspective(img, mtrx, (width, height))


(OpenCV) D:\Projects\Lecture_source>
```

프로그램 구현 2

이미지 전처리 함수 추가

```python
def ImgProcessing():
    global img

    #이미지파일 그레이스케일로 변환
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #이미지파일 음영 평준화
    norm_img = np.zeros((img.shape[0], img.shape[1]))
    img = cv2.normalize(img, norm_img, 0, 255, cv2.NORM_MINMAX)

    #이미지파일 가우시안블러 처리
    img=cv2.GaussianBlur(img, (3,3), 0)

    #이미지파일 오츠쓰래쉬홀드로 이산화 처리
    _, img=cv2.threshold(img, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

    #처리된 이미지 출력
    cv2.imshow("testing", img)

    #png파일로 저장
    cv2.waitKey(0)
    cv2.imwrite('./imgs/processing.png', img)

    return img
```

```python
cv2.imshow(win_name, img)
cv2.setMouseCallback(win_name, onMouse)
cv2.waitKey(0)
img = ImgProcessing()
cv2.waitKey(0)
text = GetOCR()
print(text)
```

이미지 전처리 함수 실행

프로그램 구현 2

- **cv2.normalize(src, dst, alpha, beta, norm_type, mask) -> dst**
  - 영상의 히스토그램을 정규화하는 함수로, 영상의 명암을 개선
  - src는 입력영상
  - dst는 결과영상
  - alpha는 목표값 혹은 최솟값(범위 정규화 경우)
  - beta는 최댓값(범위 정규화 경우)
  - norm_type은 정규화 방법
  - mask는 마스크 영상

프로그램 구현 2

- **cv2.GaussianBlur(src, ksize, sigmaX, dst, sigmaY, borderType) -> dst**
  - 가우시안 필터를 이용한 블러링으로 고주파 노이즈를 제거
  - src는 입력영상
  - dst는 결과영상
  - ksize는 커널 크기
  - sigmaX는 X방향의 표준편차
  - SigmaY는 Y방향의 표준편차(0이면 sigmaX와 동일하게 설정)
  - borderType은 가장자리 픽셀 처리 방식

프로그램 구현 2

- **cv2.threshold(src, threshold, value, type_flag)**
  - 임계값을 기준으로 이진화 시키는 함수
  - 글자와 배경을 명확하게 나눠주는 역할
  - src는 입력영상
  - threshold는 임계값
  - value는 임계값을 만족했을 때 적용할 값
  - type_flag는 스레시홀딩 적용 방법

결과


```
(OpenCV) D:\Projects\Lecture_source>python ImageToText.py
ImageToText.py:43: DeprecationWarning: an integer is required (got type numpy.float32).
  result = cv2.warpPerspective(img, mtrx, (width, height))
| >

(주도현종합건설
```