
```
def reorderPts(pts):
    # 전달된 도형 정보를 이용해서 4꼭지점 좌표값 구해서 리턴
    idx = np.lexsort((pts[:, 1], pts[:, 0]))
    # 컬럼 0 --> 컬럼 1 순으로 정렬한 인덱스를 리턴 받음

    pts = pts[idx] # x좌표로 정렬

    if pts[0, 1] > pts[1, 1]:
        pts[[0, 1]] = pts[[1, 0]]

    if pts[2, 1] < pts[3, 1]:
        pts[[2, 3]] = pts[[3, 2]]

    return pts
```

```

def perspective():
    filename = './images/test2.jpg'

    if len(sys.argv) > 1:
        filename = sys.argv[1]

    src = cv2.imread(filename)

    if src is None:
        print('image read failed...')
        sys.exit()

    w, h = 720, 400
    srcQuad = np.array([[0, 0], [0, h], [w, h], [w, 0]], np.float32)
    dstQuad = np.array([[0, 0], [0, 0], [0, 0], [0, 0]], np.float32)

    dst = np.zeros((h, w), np.uint8)

    src_gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
    th, src_bin = cv2.threshold(src_gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

    contours, _ = cv2.findContours(src_bin, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

    for pts in contours :
        if (cv2.contourArea(pts) < 10):
            continue

        # 외곽선 균사화 (면적이 10보다 큰 객체만 적용될 것임)
        # 다각형 처리 작업임
        approx = cv2.approxPolyDP(pts, cv2.arcLength(pts, True) * 0.02, True)

        # 컨벡스가 아니거나 사각형이 아니면 제외함
        if not cv2.isContourConvex(approx) or len(approx) != 4:
            continue # 다음 객체로 넘어가게 함

        # 외곽선을 그림
        cv2.polylines(src, [approx], True, (0, 255, 0), 2, cv2.LINE_AA)
        # 원본 이미지의 4꼭지점의 좌표값 획득
        srcQuad = reorderPts(approx.reshape(4, 2).astype(np.float32))

        # 찌그러진 간판을 반드시 사각형으로 변형시키기
        pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)
        dst = cv2.warpPerspective(src, pers, (w, h), flags=cv2.INTER_CUBIC)

        # 간판에서 글자 추출하기
        dst_rgb = cv2.cvtColor(dst, cv2.COLOR_GRAY2RGB)
        str = pytesseract.image_to_string(dst_rgb, lang='Hangul+eng')

        word = " ".join(re.findall("[a-zA-Z]+", str))
        print('문자 추출 : ', word)
        numbers = re.sub(r'^[0-9]', '', str)
        print('번호 추출 : ', numbers)

        dic = {word: numbers}
        print(dic)

    # for end

    cv2.imshow('src', src)
    cv2.imshow('src_gray', src_gray)
    cv2.imshow('src_bin', src_bin)
    cv2.imshow('dst', dst)

    cv2.waitKey()
    cv2.destroyAllWindows()

return dic

```

```
if __name__ == '__main__':
    dic_buf = perspective()
    conn = create_conn()
    outputDB(conn, dic_buf)
```

```
# sign.py

import re
import sys
import numpy as np
import cv2
import pytesseract
import cx_Oracle

# -----
# db 설정

dbURL = 'localhost:1521/xe'
dbUSER = 'c##student'
dbPASSWD = 'student'

def create_conn():
    try:
        conn = cx_Oracle.connect(dbUSER, dbPASSWD, dbURL)
        conn.autocommit = False
        return conn
    except Exception as msg :
        print('오라클 연결 에러 : ', msg)

def close(conn):
    try:
        if conn: # conn 0| None 0| 아니면 (True 이면)
            conn.close()
    except Exception as msg :
        print('오라클 연결 해제 에러 : ', msg)

def commit(conn):
    try:
        if conn: # conn 0| None 0| 아니면 (True 이면)
            conn.commit()
    except Exception as msg :
        print('트랜잭션 커밋 에러 : ', msg) # 커밋 실패시 롤백 처리

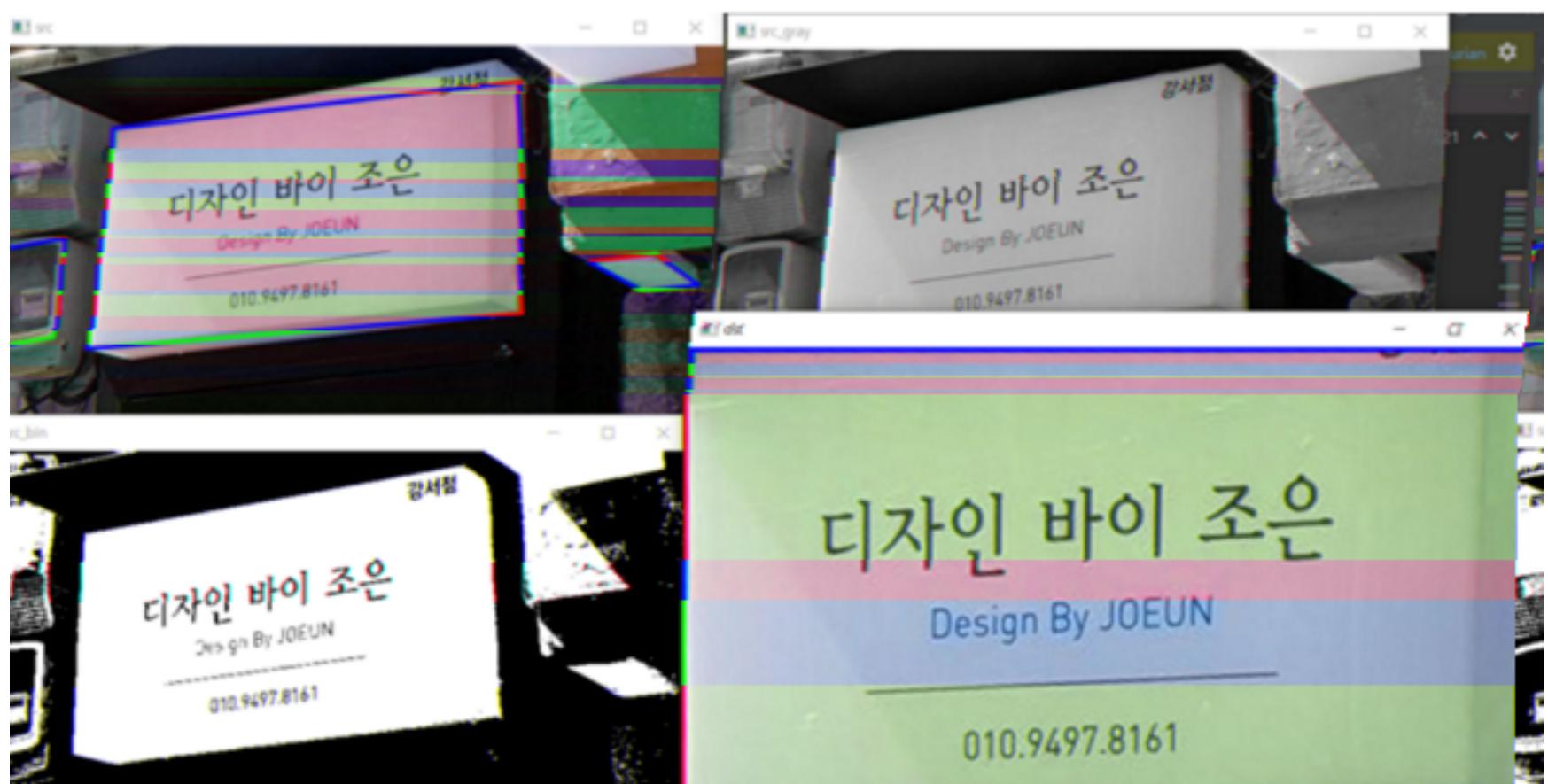
def rollback(conn):
    try:
        if conn: # conn 0| None 0| 아니면 (True 이면)
            conn.rollback()
    except Exception as msg :
        print('트랜잭션 롤백 에러 : ', msg)
```

```
def outputDB(conn, coun):
    tp_value = ()
    cursor = ''
    for i in coun:
        print(coun[i])

    tp_value = (i, coun[i])

    query = "insert into vision values (:1, :2)"

    try:
        cursor = conn.cursor()
        cursor.execute(query, tp_value) # execute할때 튜플을 넣어 쿼리문에 적용시킴
        conn.commit()
    except:
        print("오류 발생")
        conn.rollback()
        cursor.close()
        continue
    finally:
        continue
```



딕셔너리 화면 출력 완료

문자 추출 : Design By JOEUN

번호 추출 : 01094978161

{'Design By JOEUN': '01094978161'}

table 생성 완료

The screenshot shows the VISION database tool interface. The title bar says 'VISION'. The menu bar includes '시작 페이지', '수업용계정', and 'VISION'. Below the menu is a toolbar with icons for edit, insert, delete, and search. A dropdown menu '작업...' is open. The main area is a table with columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS. There are two rows: 1. NAME, VARCHAR2(100 BY...), No, (null), 1 (null); 2. TEL, VARCHAR2(50 BYTE), No, (null), 2 (null).

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 NAME	VARCHAR2(100 BY...	No	(null)	1	(null)
2 TEL	VARCHAR2(50 BYTE)	No	(null)	2	(null)

table에 데이터 insert 완료 화면

The screenshot shows the VISION database tool interface. The title bar says 'VISION'. The menu bar includes '시작 페이지', '수업용계정', and 'VISION'. Below the menu is a toolbar with icons for edit, insert, delete, and search. A dropdown menu '정렬...' is open. The main area is a table with columns: NAME and TEL. There is one row: 1 Design By JOEUN, 01094978161.

	NAME	TEL
1	Design By JOEUN	01094978161