

OCR 라이브러리 및 API 기반 주민등록증 이미지 인식 프로그램

이루지명

백지명, 강진영, 고예성, 박현식, 안영준, 조세연

CONTENTS

- 1 OCR 및 라이브러리/API
- 2 주제 소개
- 3 프로그램 구현
- 4 OCR 라이브러리/API 비교

Part 1.

OCR 및 라이브러리/API

1-1. Easy OCR

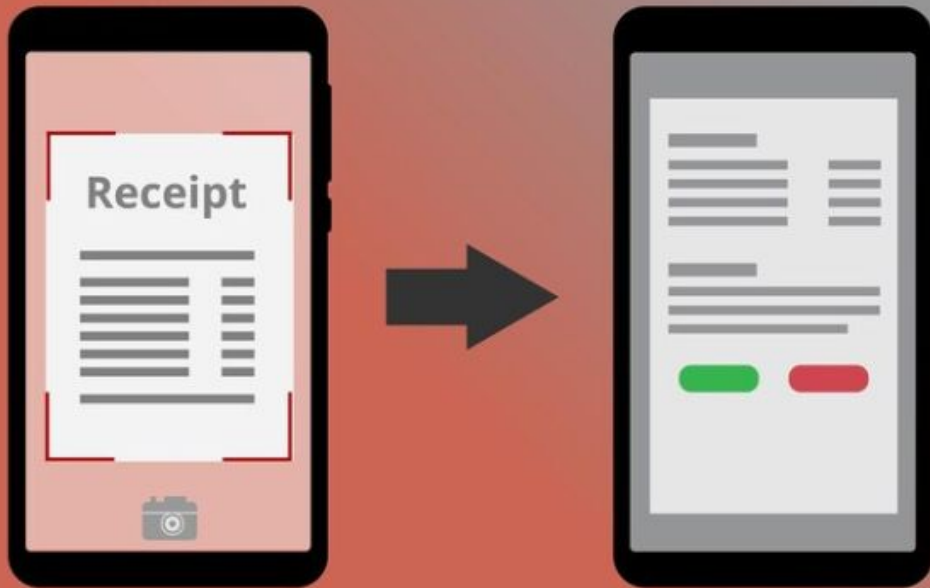
1-2. Paddle OCR

1-3. Google Vision API



OCR(Optical Character Recognition)

Optical Character Recognition (OCR)



OCR(광학 문자 인식)이란?

- 이미지나 문서에서 **텍스트를 자동으로 인식**하고 컴퓨터가 이해할 수 있는 텍스트 형식으로 변환하는 기술

OCR이 중요한 이유!

- 이미지 파일에서는 텍스트를 편집하거나 계산할 수 없지만 **OCR을 사용하면 이미지를 텍스트 문서로 변환하여 편집 및 계산이 가능!**
- 문서 디지털화, 자동 데이터 추출, 검색 가능한 문서 생성, 번역, 자동 번역 등 다양한 응용 분야에 활용할 수 있다
- OCR 시스템은 딥러닝과 컴퓨터 비전 기술의 발전으로 **높은 정확성과 다양한 언어 지원** 등 여러가지 기능을 제공한다

OCR(Optical Character Recognition)

OCR의 작동 방식

① 이미지 획득

- 스캐너를 통해 문서를 읽어들이어 이진 데이터로 변환

② 이미지 전처리

- OCR 알고리즘에 적합한 형태로 전처리
- 노이즈를 제거하고 텍스트를 더 잘 감지할 수 있도록 준비
- 기울기보정 등으로 정렬 문제 해결 + 이미지의 얼룩을 제거하거나 텍스트 이미지의 가장자리를 부드럽게 만들

③ 텍스트 감지 및 분할

- 전처리가 완료된 이미지에서 텍스트의 위치를 감지하여 **Bounding Box**, 텍스트 영역을 개별문자 또는 단어로 분할

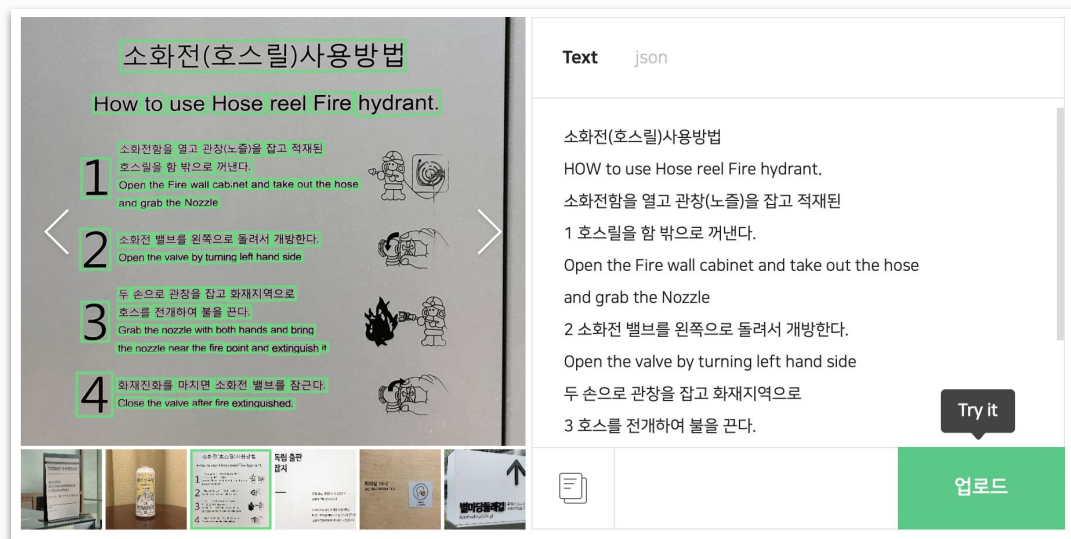
④ 문자 인식

- 딥 러닝 알고리즘을 통해 텍스트를 인식. 미리 학습된 신경망 모델을 사용하여 문자를 인식하고 텍스트로 변환

⑤ 후처리

- 인식된 텍스트의 오류를 보정하거나 포맷을 조정하여 결과물을 더 정확하고 읽기 쉽게 만들

1.1 Easy OCR



특징

- 간단하고 사용하기 쉬운 오픈소스 OCR 라이브러리
- Python기반으로 개발되었으며, 40개 이상의 언어를 지원한다

장점

- 오픈소스이기 때문에 사용과 활용에 무료설치와 설정이 간단하며, 몇 줄의 코드로 OCR 기능을 적용할 수 있음
- 한글을 비롯한 다양한 언어의 인식이 가능
- Tesseract OCR과 CRAFT(Character Region Awareness for Text Detection)를 기반으로 하여, 높은 인식률을 제공

단점

- 정교한 튜닝 없이 사용하기에는 성능이 떨어질 수 있다
- 일부 언어에서 다른 상용 OCR 솔루션에 비해 정확성이 낮을 수 있다

1.2 Paddle OCR

学生个人信息登记表					
姓 名	王致和	性别	女	民族	汉
曾用名	无	出生日期		2000年2月27日	
籍 贯	甘肃省兰州市	出生地		甘肃省兰州市	
政治面貌	党员	QQ		589658215	
身份证号	620100200002270054				
E-mail	589658215@qq.com	在校期间电话		13008710546	
学生干部 任职情况	2010年 6 月——2012年9 月 在 班级担任 班长				
家庭详细 地址	甘肃省兰州市紫金花园13号楼1单元201				
固定电话	13008710546	邮政编码		730000	

특징

- PaddlePaddle 딥러닝 프레임워크를 기반으로 한 오픈소스 OCR 라이브러리
- 다양한 딥러닝 모델을 지원

장점

- 한글을 비롯한 다양한 언어의 텍스트 인식을 지원하여 정확성과 다양성이 높음
- 오픈소스이며, 상업적인 사용에도 자유로움
- 고성능과 확장성을 갖추고 있어 대량의 데이터를 처리하는데 용이

단점

- 최근에 개발되었기 때문에 문서화가 적은편
- 모델에 대한 이해도가 높아야 최대 성능을 이끌어낼 수 있음
- 특정 언어나 작은 데이터셋에 대한 성능은 다른 OCR 라이브러리보다 떨어짐

1.3 Google Vision API



특징

- 구글에서 제공하는 클라우드 기반 **OCR**서비스로, 높은 인식 정확도와 다양한 기능을 제공합니다

장점

- Google의 강력한 기계 학습 알고리즘과 딥러닝 모델을 기반으로 하여 **높은 정확도와 성능을 제공**
- 이미지 안의 객체, 얼굴, 텍스트 등 다양한 객체들을 정확하게 인식하고 분석
- 다국어를 지원하며, 다양한 언어로 된 텍스트를 추출하고 인식
- 클라우드 기반 서비스이므로, 확장성이 높고 다양한 플랫폼과 애플리케이션에서 쉽게 사용할 수 있습니다.

단점

- 유료 서비스이기 때문에 대량의 이미지 처리나 높은 사용량의 경우 추가 **비용이 발생**
- 민감한 정보가 포함된 이미지의 사용은 주의해야 함
- 일부 사용자 정의 기능을 제공하지 않음

Part 2.

주제 소개



2. 주제

문자 탐지



내부 문자 영역 찾기

영역 탐색

주민번호 960709 - 1

필요한 영역 추출

문자 인식

960709 - 1 → 960709

“나이는 만 27살 입니다”

이미지에서 문자열로 변환 후
성인 여부 판별

Part 3.

프로그램 구현



3.1 Paddle OCR practice

코드1. 반듯한 사진인 경우

```
ocr = PaddleOCR(use_angle_cls=True, lang='korean')

img = cv2.imread('주민등록증.jpg')

x, y, w, h = cv2.selectROI('img', img, False)
if w and h:
    roi = img[y: y + h, x: x + w]

    result = ocr.ocr(roi)

    for line in result:
        for word_info in line:
            print(f"텍스트: {word_info[1][0]}, 좌표: {word_info[0]}")

cv2.imshow('roi', roi)
cv2.waitKey()
```

- ✓ PaddleOCR 모델 초기화 + 각도 감지 모델 사용 + 한국어 모델 사용
 - ✓ 확인하고싶은 주민등록증 이미지 파일을 불러와서 사용
 - ✓ cv2.selectROI()함수를 사용해서 이미지 영역을 선택할 수 있는 창을 띄운다
 - ✓ if문을 사용해 선택한 영역이 존재하는 경우에만 아래의 코드를 실행
-
- ✓ for문을 사용해 PaddleOCR의 결과를 순회하면서 각 줄에 대한 정보에 접근
 - ✓ for문을 사용해 각 줄에서 단어 정보에 접근
 - ✓ 각 단어의 텍스트와 해당 좌표를 출력

3.1 Paddle OCR practice

실행결과

12.0, 362.0], [1289.0, 362.0], [1289.0, 500.0], [412.0, 500.0]], ('주민등록
83.0, 615.0], [1211.0, 620.0], [1210.0, 732.0], [383.0, 726.0]], ('조세연趙
27.0, 825.0], [1278.0, 833.0], [1277.0, 905.0], [326.0, 897.0]], ('961107-2
95.0, 1006.0], [712.0, 1012.0], [711.0, 1086.0], [293.0, 1080.0]], ('인천광
58.0, 1011.0], [1504.0, 1017.0], [1504.0, 1089.0], [75

주민번호: 961107-2
성인 여성입니다.



! 팀원의 주민등록증 사진을 적용했을 때 결과

- ✓ 위에서부터 순서대로 주민등록증, 이름, 한자, 주민번호, 주소 등 텍스트를 도출해냈습니다.
- ✓ 그 결과 뽑아낸 주민번호로 유효성 검사와 성인 여부도 판단이 가능했습니다.

이미지에서 텍스트를 정확하게 도출한것을 확인할 수 있습니다

3.1 Paddle OCR practice

코드2. 기울어진 사진

+ 원근 변환 전처리를 한 경우

```
ocr = PaddleOCR(use_angle_cls=True, lang='korean')
img = cv2.imread('./img_2.png')

1 usage
def perspective_selection(event, x, y, flags, param):
    global pts, idx
    if event == cv2.EVENT_LBUTTONDOWN:
        pts[idx] = [x, y]
        idx += 1
        cv2.circle(img, (x, y), 5, (0, 0, 255), -1) # 선택한 점을 빨간색 원으로 표시

pts = np.zeros(shape=(4, 2), dtype=np.int32)
idx = 0
```

- ✓ PaddleOCR 모델을 초기화 하고
- ✓ perspective_selection(): 마우스로 영역을 선택하는 콜백함수를 정의
- ✓ pts : 점 4개의 좌표를 저장할 배열을 생성

```
while True:
    cv2.imshow('img', img)
    cv2.setMouseCallback('img', perspective_selection)
    if idx == 4:
        break
    cv2.waitKey(1)

w, h = 600, 400
dst_pts = np.array(p_object: [[0, 0], [w, 0], [w, h], [0, h]], np.float32)
M = cv2.getPerspectiveTransform(pts.astype(np.float32), dst_pts)
warped_img = cv2.warpPerspective(img, M, (w, h))
```

- ✓ while문으로 무한루프 시작
- ✓ if문으로 사용자가 마우스로 점 4개를 선택한 경우에 아래 코드를 실행
- ✓ w, h : 선택한 영역의 가로와 세로 크기를 설정
- ✓ dst_pts: 선택한 영역을 정방향으로 기울기 보정

3.1 Paddle OCR practice

실행결과2



✓ 주민등록증에서 추출하고자 하는 영역의 꼭짓점을 찍어주면 해당 영역이 선택됩니다.

✓ 선택된 영역의 기울기가 보정되고, 기울기가 보정된 사진에 PaddleOCR 을 이용해서 텍스트를 읽어옵니다.

기울어진 사진의 경우, 원근 변환 처리를 입혀준 후에도 텍스트를 읽어오는 정확도가 떨어졌습니다

3.2 Easy OCR practice

이미지 전처리 및 EasyOCR 활용

```
def extract_text_in_selected_area(image, points):
    mask = np.zeros_like(image)
    cv2.fillPoly(mask, pts=[points], color=(255, 255, 255))
    selected_area_image = cv2.bitwise_and(image, mask)

    width, height = 800, 600
    srcQuad = np.array(points, np.float32)
    dstQuad = np.array([0, 0], [width, 0], [width, height], [0, height], np.float32)
    pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)
    selected_area_image = cv2.warpPerspective(selected_area_image, pers, dsize=(width, height))

    langs = ['ko', 'en']
    reader = Reader(lang_list=langs, gpu=True)
    results = reader.readtext(selected_area_image)

    result_text_list = []
    for (bbox, text, prob) in results:
        result_text_list.append(text)

    return result_text_list
```

- ✓ cv2.getPerspectiveTransform 함수를 사용하여 srcQuad와 dstQuad로부터 기울기 보정에 필요한 변환 행렬(pers)을 계산합니다.
- ✓ cv2.warpPerspective 함수를 사용하여 selected_area_image에 pers를 적용하여 선택한 영역을 기울기를 보정한 직사각형 형태로 변환합니다.

3.2 Easy OCR practice

결과



성인입니다 주민등록번호: 960709-1 , 나이: 27

EasyOCR은 기울기를 전처리 하지 않으면 올바르게 인식하지 못하였고
기울기 보정을 적용 후 올바른 데이터를 추출하였습니다.

3.3 Google Cloud Vision API

Google Vision API를 사용하여 이미지에서 텍스트를 인식하는 부분입니다.

API 불러오기

```
service_account_path = "/content/drive/MyDrive/데분팀/0731/service-account-file.json"
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = service_account_path
client = vision.ImageAnnotatorClient()
```

```
img_path = '/content/drive/MyDrive/데분팀/0731/1.jpg'
with io.open(img_path, 'rb') as image_file:
    content = image_file.read()
image = vision.Image(content=content)

response = client.text_detection(image=image)
texts = response.text_annotations
```

```
for text in texts:
    ocr_text = text.description
    print('\n"{}"'.format(ocr_text))
```

✓ 이 코드에서는 서비스 계정 키 파일의 경로를 설정하고, 이를 환경 변수 `GOOGLE_APPLICATION_CREDENTIALS`에 지정하여 구글 클라우드 비전 API와 인증합니다.

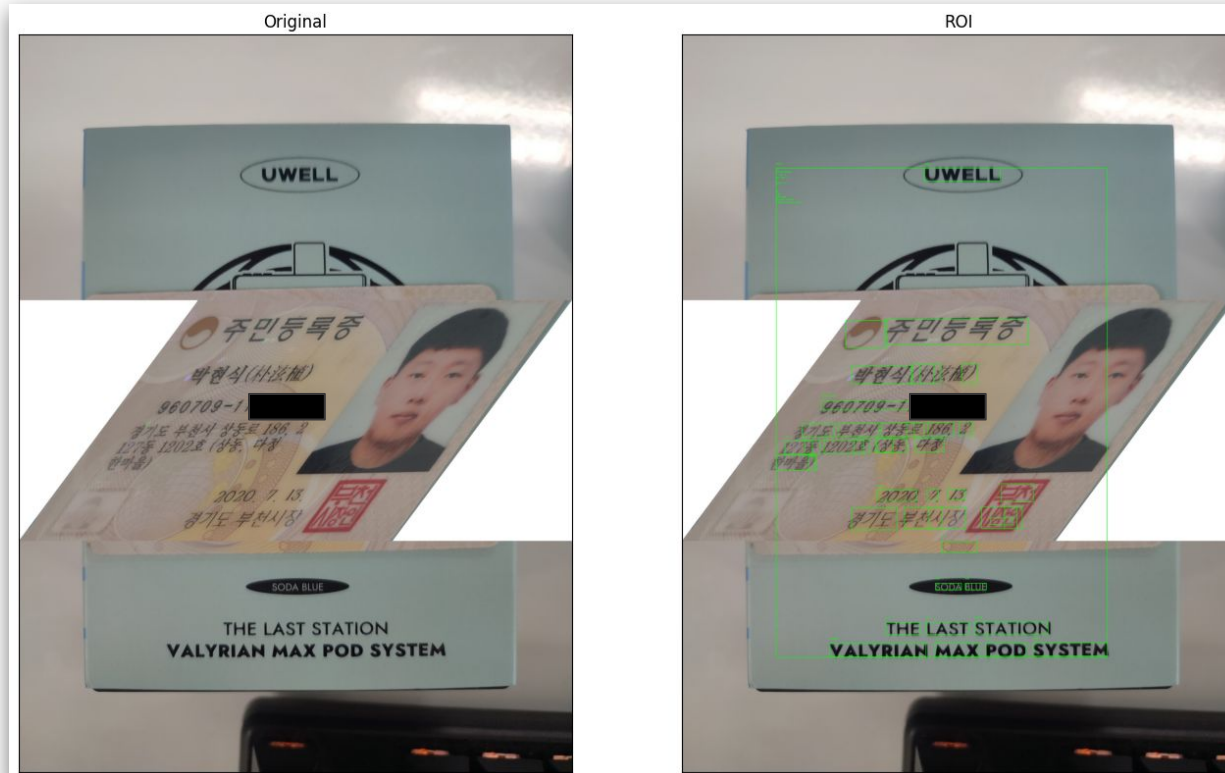
✓ 지정된 경로의 이미지 파일을 읽습니다.

Vision API의 `text_detection` 메서드를 호출하여 준비된 이미지를 전달합니다. API는 이미지를 처리하고 이미지 내의 텍스트를 인식합니다. 인식된 결과는 `texts` 변수에 저장됩니다.

✓ 이 코드에서는 인식된 텍스트 어노테이션을 반복하면서 `text.description` 속성에서 인식된 텍스트를 추출합니다. 추출된 텍스트는 `ocr_text` 변수에 저장됩니다. 이후 성인 판별 함수를 이용하여 결과를 도출해 냅니다.

3.3 Google Cloud Vision API

결과



"960709-1 [redacted]
주민등록번호: 96070911 [redacted]
성인 주민등록번호: 960709-1 [redacted] 나이: 27

Google Cloud Vision API 의 경우 전처리를 하지 않아도 우수하게 텍스트를 추출하였습니다.

Part 4.

OCR 라이브러리/API 비교



4. OCR 라이브러리/API 비교

Paddle OCR



PaddleOCR의 경우 전처리가 필요하지 않은 이미지를 이용하는 경우 우수한 성능을 보여줬지만 품질이 떨어지는 이미지를 이용하는 경우에는 다른 라이브러리들에 비해 인식 성능이 떨어졌습니다.

Easy OCR



EasyOCR은 원본 이미지의 품질이 우수한 경우 인식 오류가 적어집니다.
하지만 전처리를 하지 않으면 성능이 다른 라이브러리/API 보다 성능이 떨어진다고 생각합니다.

Google
Vision API



Google Cloud Vision API의 경우 초기 설정이 번거로운 느낌이 들지만
타 라이브러리/API와 다르게 전처리를 하지 않아도 우수한 성능이 보여진다고 생각합니다.

감사합니다 🥰