Text Extraction and Detection from Images
using Machine Learning Techniques
: A Research Review

Text extraction # Text detection # OCR(Optical Character Recognition)

박용주

2023-06-01

목 차

Abstract

Img1

img2

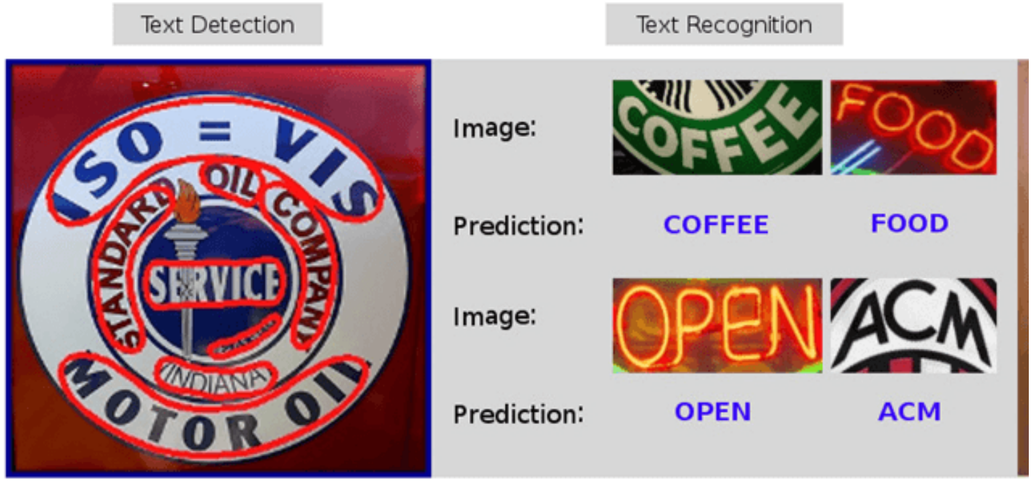
KEEP OUT
RESTRICTED
ACCESS

IMG3

- Text Detection / Text Extraction
  - 중요한 ML 응용 분야 중 하나
  - 이미지 속 텍스트의 사이즈, 방향, 정렬, 글씨체, 대비 정도, 배경 등이 상이하다는 이슈
- OCR (Optical Character Recognition)
  - 대표적인 ML 알고리즘
  - 이미지 데이터에서 텍스트를 추출하고 검색 가능, 편집 가능한 데이터로 변환하는 알고리즘

1. Introduction

- Text Detection
  - Computer vision 분야의 주요한 문제
  - 이미지에서 Text에 bounding box 짓는 Task
  - 자율 주행, 산업 자동화, 네비게이션 등 응용 분야 넓음
  - 색깔 객체 탐지 → canny edge detection algorithm
  - 텍스트 detection → OCR(Optical Character Recognition)
  - 주요 모델: CRAFT (Character Region Awareness for Text Detection)
- Text Recognition
  - 이미지로부터 텍스트 추출하는 Task
  - 주요 모델: RedAI



[Canny Edge Detection Step by Step in Python — Computer Vision]

윤곽선 탐지 알고리즘

2. RTBPN (Randon Transform and Back Propagation Network)

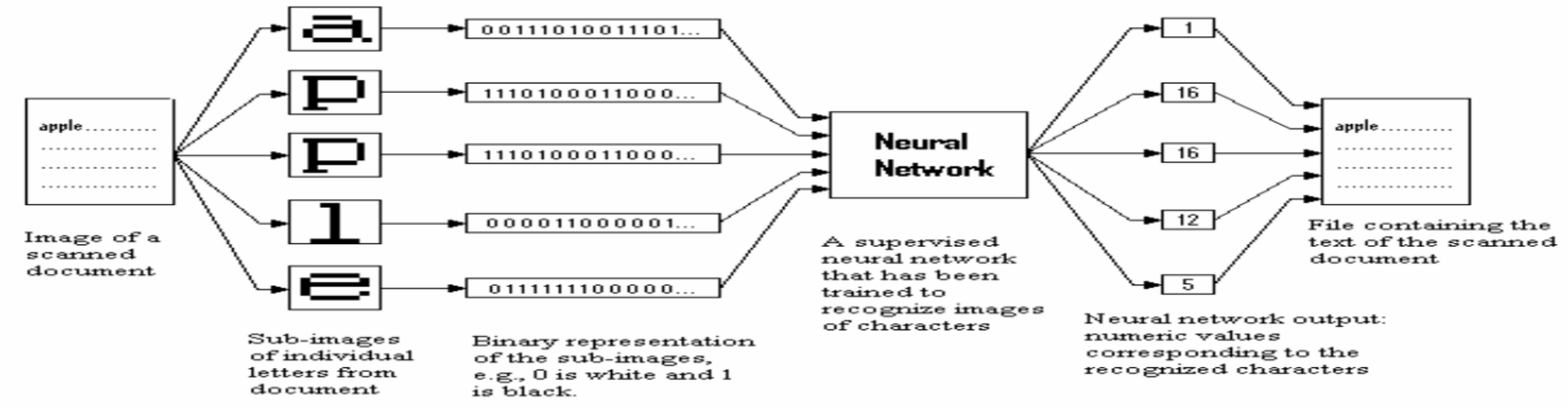


Fig. 7. Flow diagram of proposed algorithm

RTBPN (Randon Transform and Back Propagation Network) machine learning algorithm

Step 1) original image를 OCR software에 의해 sub-image로 분리함

Step 2) back propagation method 으로 image 에서 binary 형식으로 변환함

3. OCR (Optical Character Recognition)

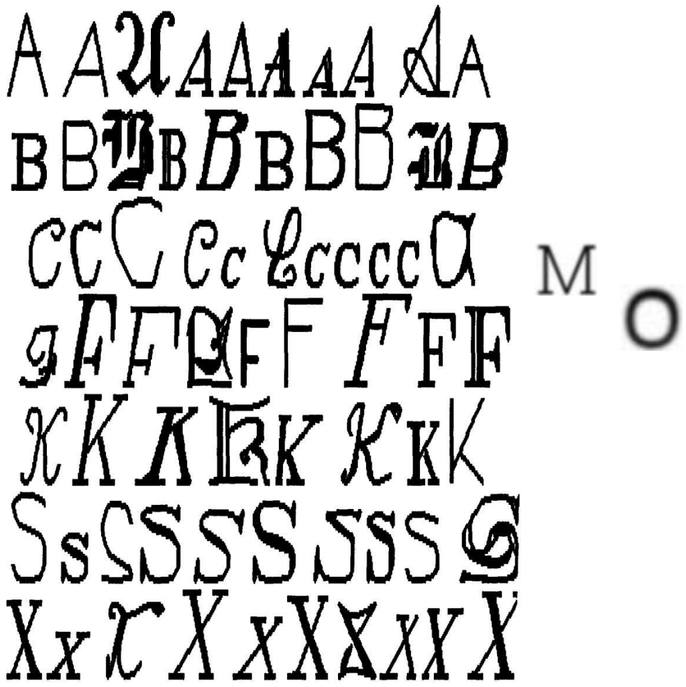
- OCR(Optical Character Recognition)이란
  - Text recognition task라고도 불린다
  - [input] 스캔한 문서, 카메라 이미지, 이미지로만 구성된 pdf
  - [output] machine encoded format의 문서
  - Cf) EasyOCR 라이브러리
    - Python package. OCR 쉽게 구현 가능함
  - 작동 단계
    - Image Acquisition > Pre-processing > Segmentation > Extraction w/ OCR
      > Training & Testing > Text detected

3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → Segmentation → Extraction w/ OCR → Training & Testing → Text detected

3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → Segmentation → Extraction w/ OCR → Training & Testing → Text detected



step 1. Image Acquisition

- Raw input dataset(non-editable text data)
- Images, scanned documents
- standard OCR dataset | Kaggle
- Letter Dataset | Papers With Code

3. OCR (Optical Character Recognition)

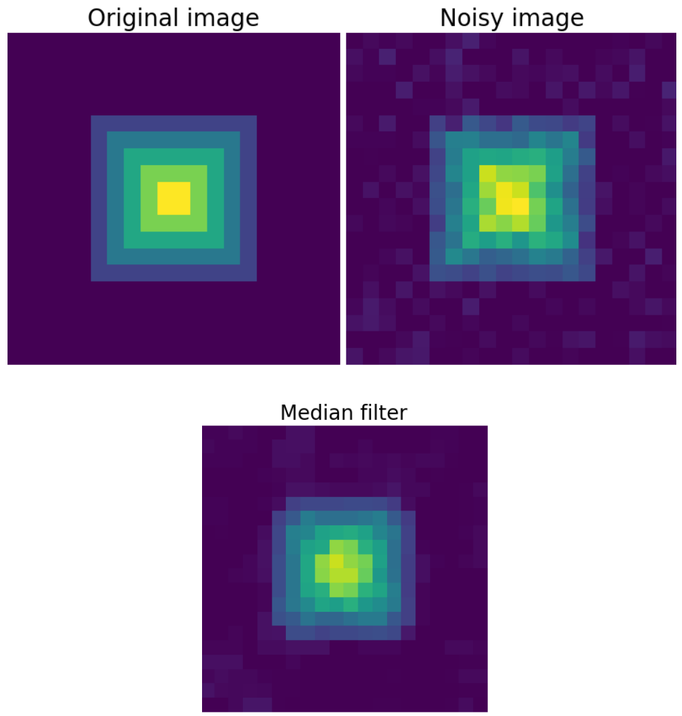
Image Acquisition → Pre-processing → Segmentation → Extraction w/ OCR → Training & Testing → Text detected

step 2. Preprocessing

- Remove noise, null value, duplicate data, irregular data
- Median filter, gaussian filter 적용

3. OCR (Optical Character Recognition)

Image Acquisition → **Pre-processing** → Segmentation → Extraction w/ OCR → Training & Testing → Text detected



step 2. Preprocessing

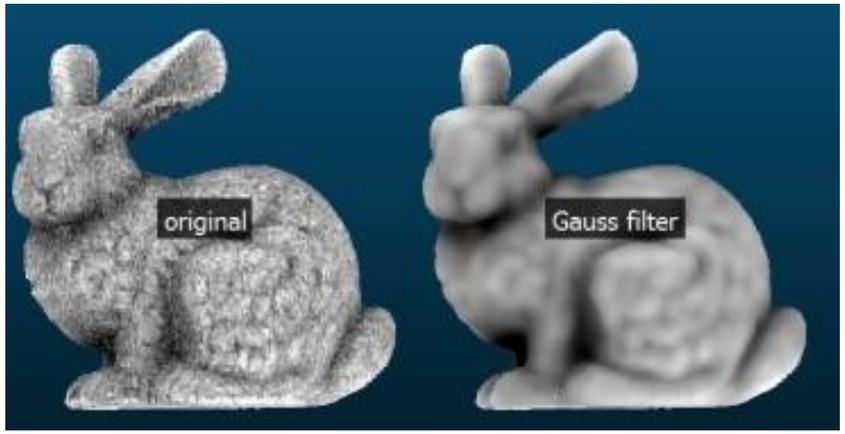
- Remove noise, null value, duplicate data, irregular data
- Median filter, gaussian filter 적용

Median filter

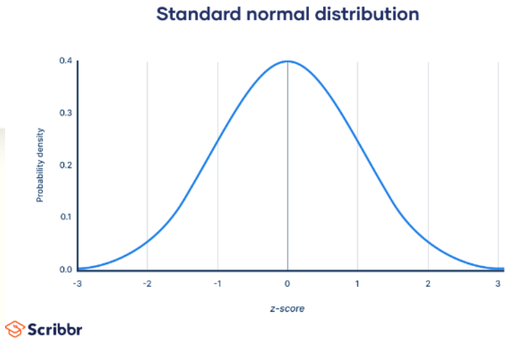
- Non-linear digital filtering
- 이미지나 신호에서 노이즈를 제거하기 위해 사용됨
- Mean filter와 달리 평균 말고 중앙값으로 대체함
- Spatial Filters - Median Filter

3. OCR (Optical Character Recognition)

Image Acquisition → **Pre-processing** → Segmentation → Extraction w/ OCR → Training & Testing → Text detected



$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$$



step 2. Preprocessing

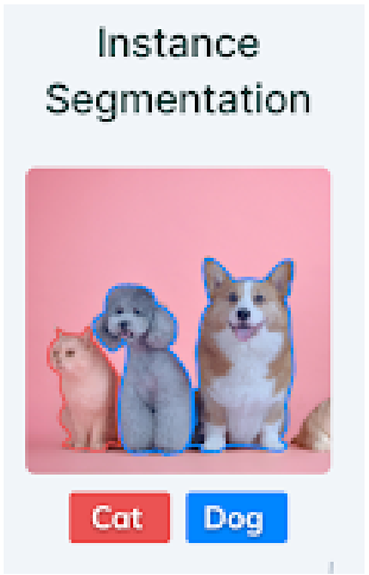
- Remove noise, null value, duplicate data, irregular data
- Median filter, gaussian filter 적용

Gaussian filter

- Gaussian distribution을 근사하여 생성한 필터 마스크를 사용하는 필터링 기법
- 필터링 대상 픽셀 근처에서 가중치를 크게 주고
- 필터링 대상 픽셀과 먼 주변부에서는 가중치는 조금만 주어서
- 가중 평균을 구한다 [OpenCV] 블러링 기법과 가우시안 필터

3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → **Segmentation** → Extraction w/ OCR → Training & Testing → Text detected



전경과 배경을
나누는 작업

step 3. Segmentation

- 전처리된 이미지를 character 를 기준으로 분할한다
- 각 영역으로 분할된다
- 문자를 포함하고 있는 픽셀 그룹에 대한 이미지 스캔이 포함된다
- 각 문자가 자체 클래스에 할당됨
- Minimum / maximum thresholding value 를 고려해야 함
  - Ostu’s binary thresholding
  - Adaptive thresholding 적용 필요

3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → **Segmentation** → Extraction w/ OCR → Training & Testing → Text detected

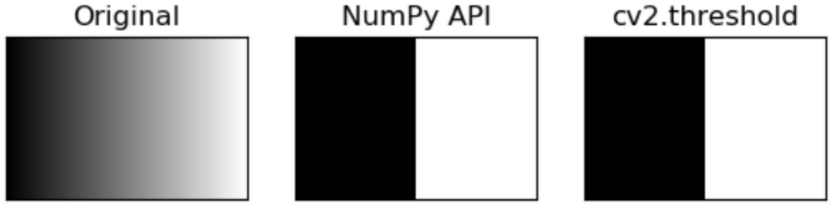
- Threshold-based segmentation method
  - Image binarization process (binary image 만드는 대표적 방법)
  - 어떤 임곗값을 기준으로 두 부류로 나눈다(말하자면 전경과 배경)
  - 예를 들면, [0, 120] → 0, (120, 255] → 255
- 3가지 방식
  - Global threshold segmentation
  - Otsu’s threshold segmentation
  - Adaptive local threshold segmentation

3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → **Segmentation** → Extraction w/ OCR → Training & Testing → Text detected

1. Global threshold segmentation

임곗값보다 크면 255, 작으면 0



2. Otsu's threshold segmentation

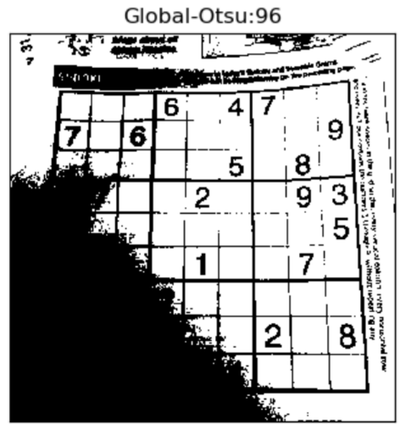
임곗값을 정하는 방법

명암 분포가 가장 균일할 때의 임곗값을 선택한다

3. Adaptive local threshold segmentation

균일하지 않은 경우

지역을 나누어 계산



3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → Segmentation → **Extraction w/ OCR** → Training & Testing → Text detected

각 영역의 높이가 x, 너비가 y

step 4. Extraction with OCR

- OCR template 으로 픽셀 그룹의 feature 를 추출
- OCR text extraction 의 방식은 다음과 같음
  - Region-based
  - Hybrid-based
  - Texture-based

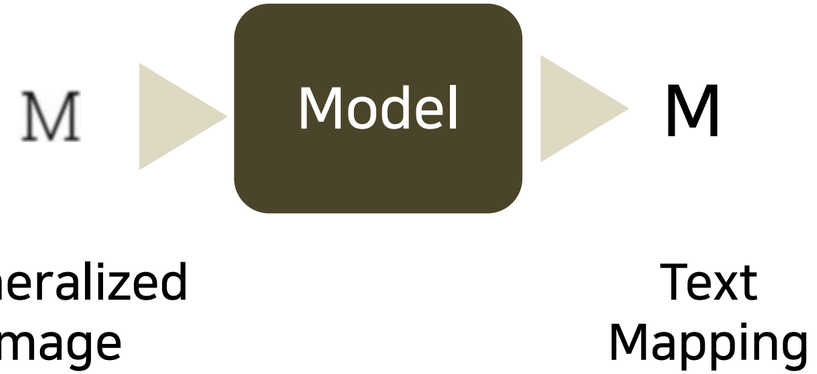
3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → Segmentation → Extraction w/ OCR → **Training & Testing** → Text detected

Figure 3: Pre-processing of the sample image

step 5. Training & Testing

- 다양한 하이퍼 파라미터가 포함됨



3. OCR (Optical Character Recognition)

Image Acquisition → Pre-processing → Segmentation → Extraction w/ OCR → Training & Testing → **Text detected**

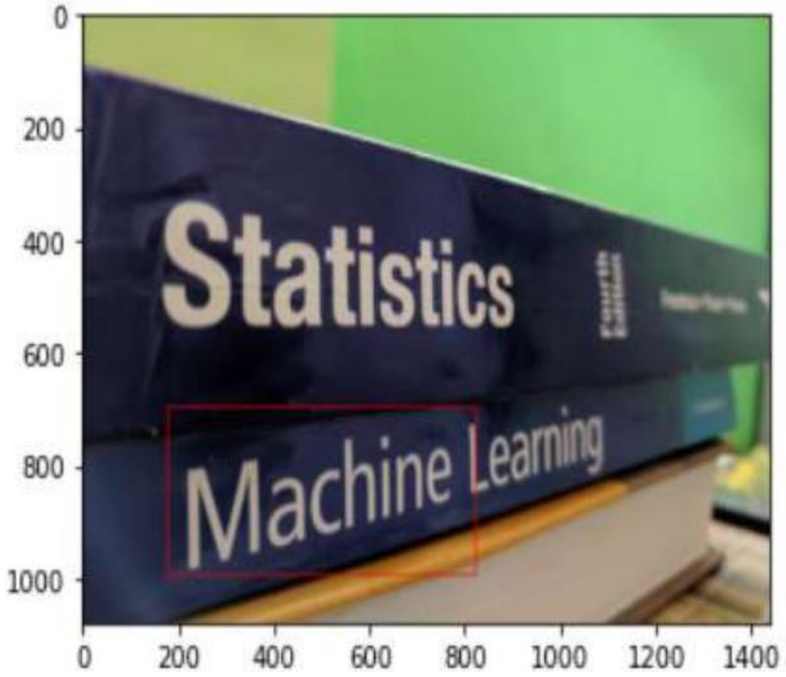


Figure 5(a): Text detection

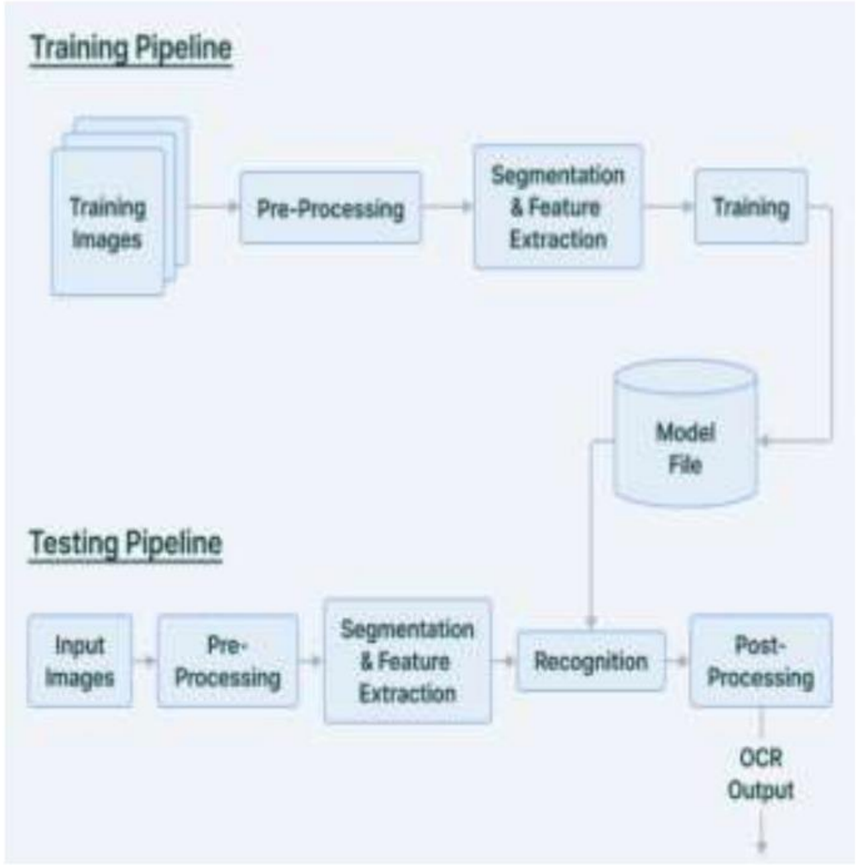
```
output

[([[141, 304], [946, 304], [946, 608], [141, 608]],
  'Statistics',
  0.6300734877586365),
 ([[774.7574158649826, 715.6888679100582],
   [1108.24643686469 47, 671.1484372203447],
   [1127.2425841350175, 818.3111320899418],
   [793.7535631353053, 861.8515627796553]],
  'Learning',
  0.9920356869697571),
 ([[179.21138355776188, 803.7558522105795],
   [791.2259041294647, 696.5540755887588],
   [824.7886164422381, 888.2441477894205],
   [212.77409587053535, 994.4459244112412]],
  'Machine',
  0.9928504824638367)]
```

Figure 5(b): Sample Output

step 6. Text detected

- 사람의 피드백이 필요함

3. OCR (Optical Character Recognition)



최종 OCR 파이프라인

4. Text-based technologies and various applications

Various OCR technologies: 이미지, 손글씨, 스캔된 문서에서 데이터 추출하는 데 사용됨

Google (2006)

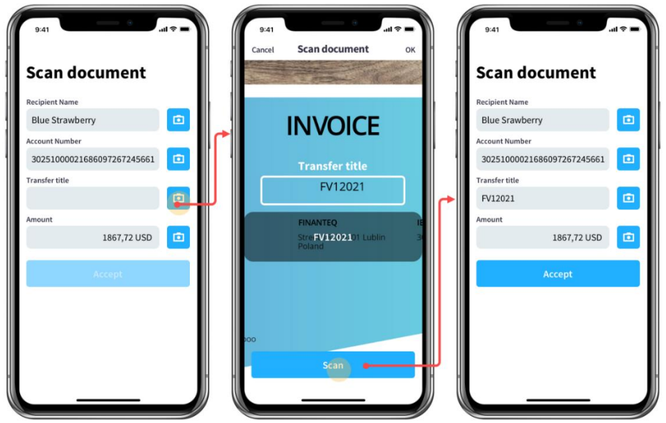
오픈 소스 OCR 엔진

유니코드 지원, 100개 이상의 언어

Python 모듈 pytesseract

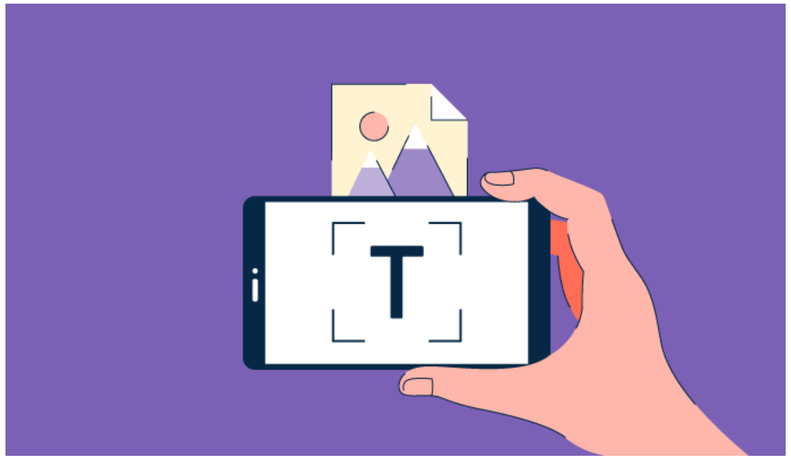
OmniPage, Abby Cloud, UiPath Document, Python library EasyOCR

4. Text-based technologies and various applications



은행, 정부

- 수표 처리 자동화
- 신분증 전산화



의료, 법조계 등 수많은 산업

- 모든 문서를 데이터베이스화

Optical character recognition helps students with dyslexia

By Antonio Williamson

Optical character recognition (OCR) plays a crucial role in converting printed materials into digital text files. These digital files can be helpful to children and adults for those who have trouble reading. That is because digital text can be utilized with software programs that support reading differently.

교육 분야

- 난독증 학생을 지원
- Text → Audio

5. Conclusion

Various technologies가 text recognition에 사용될 수 있음을 알게 됨

Banking, Education, Government, Healthcare sectors에서 사용됨

실습

Python 광학 문자 인식(OCR): 튜토리얼 | 내장 (builtin.com)
JaidedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. (github.com)
Optical Character Recognition (OCR) in Python - Python Code (thepythoncode.com)