**연구 내용**

부서명 : 통로이미지㈜ 디지털 연구소

작성자 : 김근수  
참가자: 이재봉, 천길성, 홍승호, 김근수

작성일 : 2020.12.28

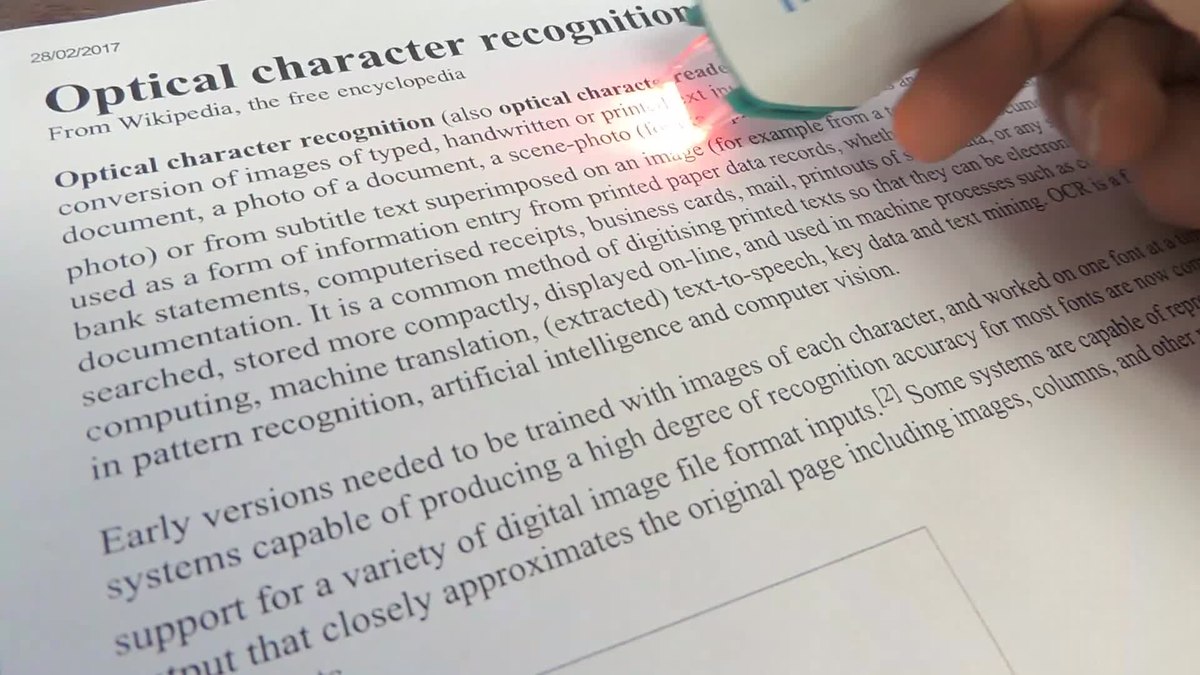
|  |  |
| --- | --- |
| 연구명 | 머신러닝\_이미지\_텍스트\_추출\_연구개발 |

1. 검토사항

* 이미지에 있는 텍스트를 추출할 수 있는 도구가 있는가?
* 이미지 텍스트를 추출 기술은 어떤것이 있는가?
* 이미지 텍스트 추출 기술을 통해 이미지내의 텍스트 여부를 판별할 수 있는가?
* 처리 할수 있는 이미지 확장자(jpg, png, gif 등등)는 어디까지 인가?
* 영어를 제외한 다국어도 인식이 가능한가?
* 텍스트 인식률은 어느정도 인가?
* 머신러닝을 통한 이미지내의 텍스트 인식율을 높일수 있는가?

1. 기술 정의

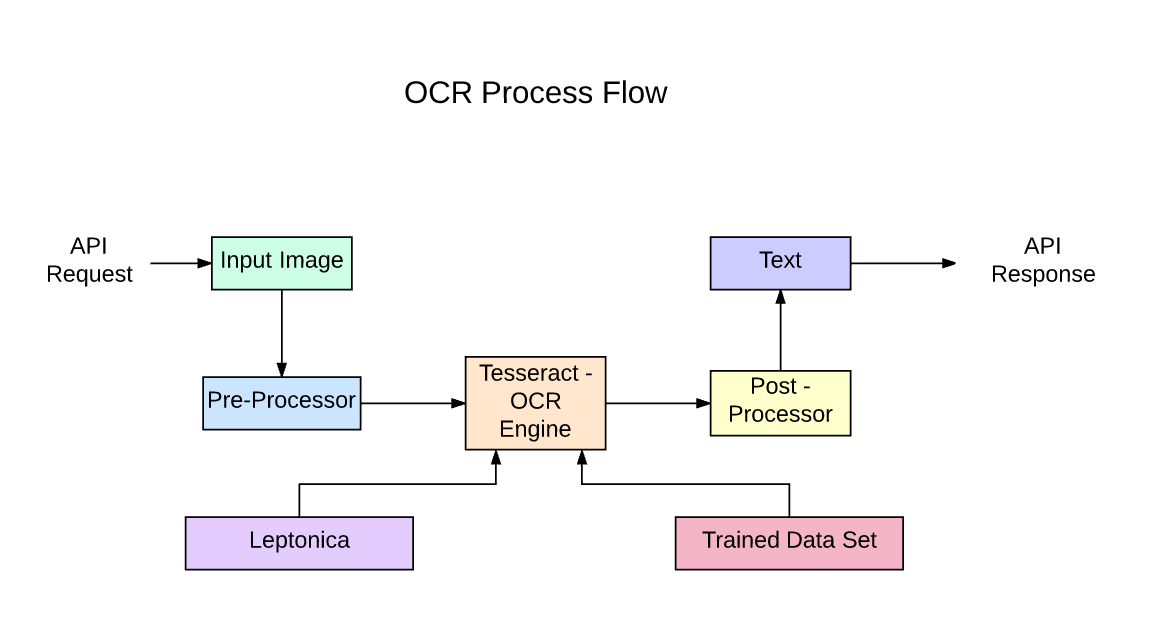
* 광학 문자 인식(Optical character recognition: OCR)
  + 사람이 쓰거나 기계로 인쇄한 문자의 영상을 이미지 스캐너로 획득하여 기계가 읽을 수 있는 문자로 변환

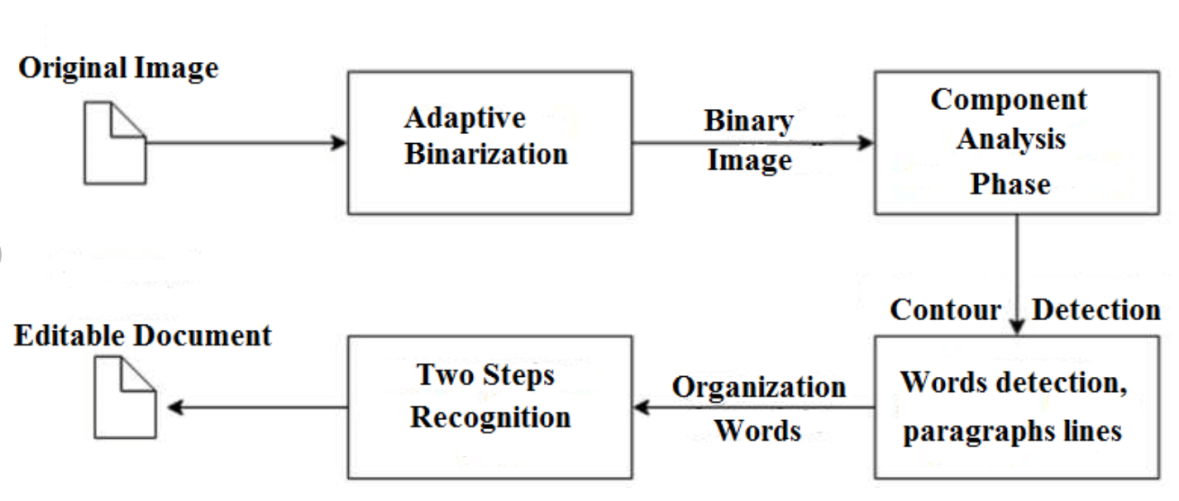


* + 이미지 스캔으로 얻을 수 있는 문서의 활자 영상을 컴퓨터가 편집 가능한 문자코드 등의 형식으로 변환하는 소프트웨어로써 일반적으로 OCR이라고 하며,   
    OCRDMS 인공지능이나 기계 시각의 연구분야로 시작되었다.

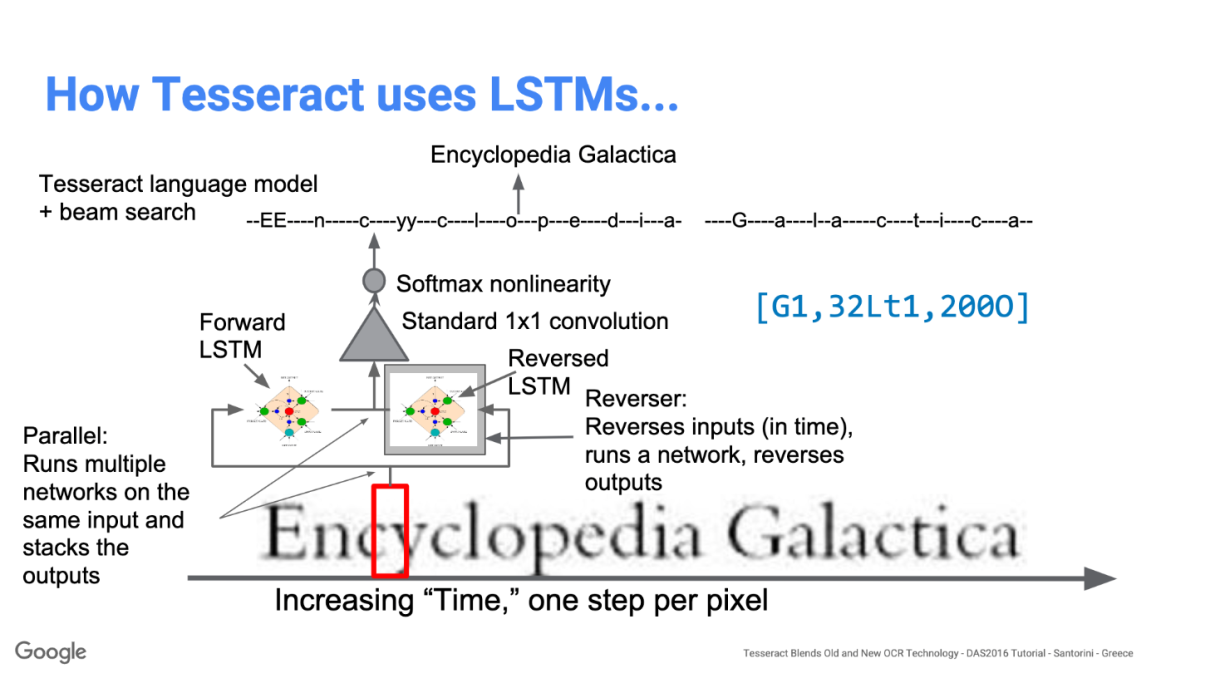


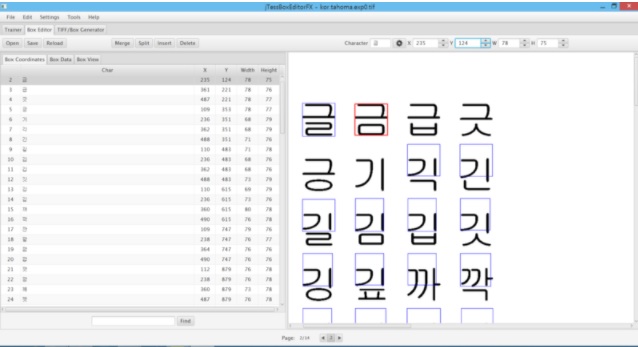
* Tesseract(ver 4.1.1)
  + 이미지로부터 텍스트를 인식하고 추출하는 소프트웨어로 1984 ~ 1994년 HP 연구소에서 개발된 오픈소스로 다양한 운영 체제를 위한 광학 문자 인식 엔진이다.





* + Apache License 버전 2.0에 따라 배포되는 무료 소프트웨어이며 2006년부터 Google에서 개발을 후원했다.
  + 현재까지도 LSTM과 같은 딥러닝 방식을 통해 텍스트 인식률을 지속적으로 개선하고 있음



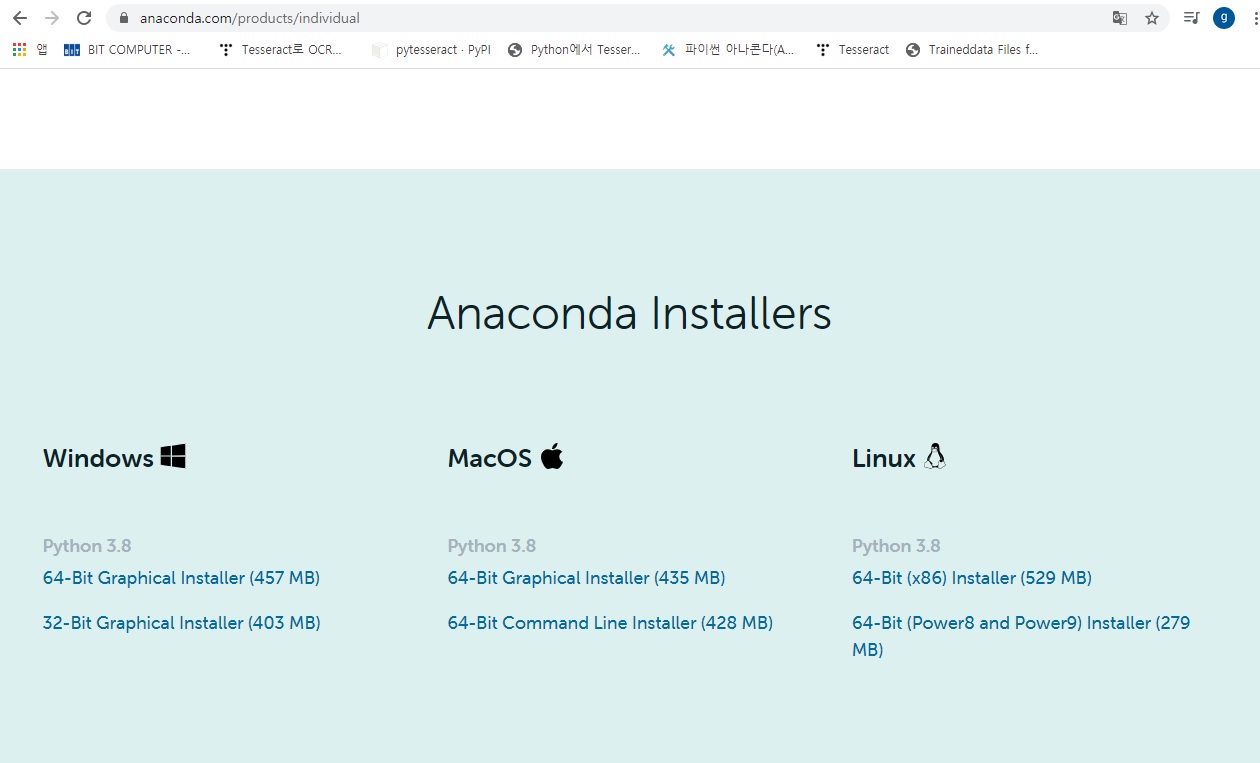


* PyTesseract(ver 0.3.7)
  + Python 용 광학 문자 인식(OCR) 도구로 이미지에 포함된 텍스트를 읽는다.
  + Google’s Tesseract-OCR Engine용 래퍼
  + Jpeg, png, gif, bmp, tiff 등을 포함하여 Pillow 및 Leptonica 이미지 라이브러리에서 지원하는 모든 이미지 유형을 읽을수 있다.
  + 지원하는 Python 버전: 3.6 ~ 3.9

1. 개발환경

* OS: Window
* Langage: Python(ver 3.6)
* Anaconda(가상환경)

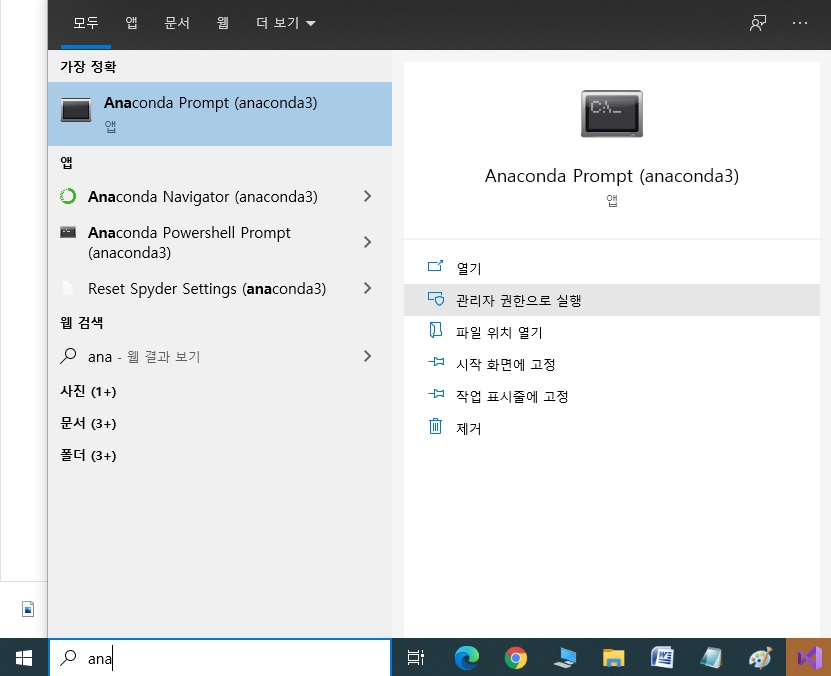
1. 환경설정
   1. Anaconda Install

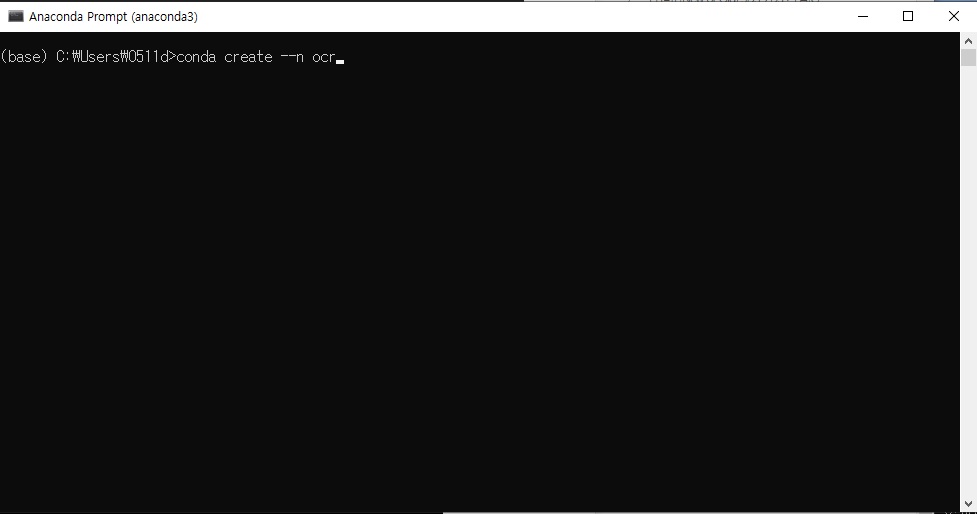


* 가상환경 생성

Anaconda Prompt를 실행한 후 하단의 명령어로 가상환경을 생성한다.

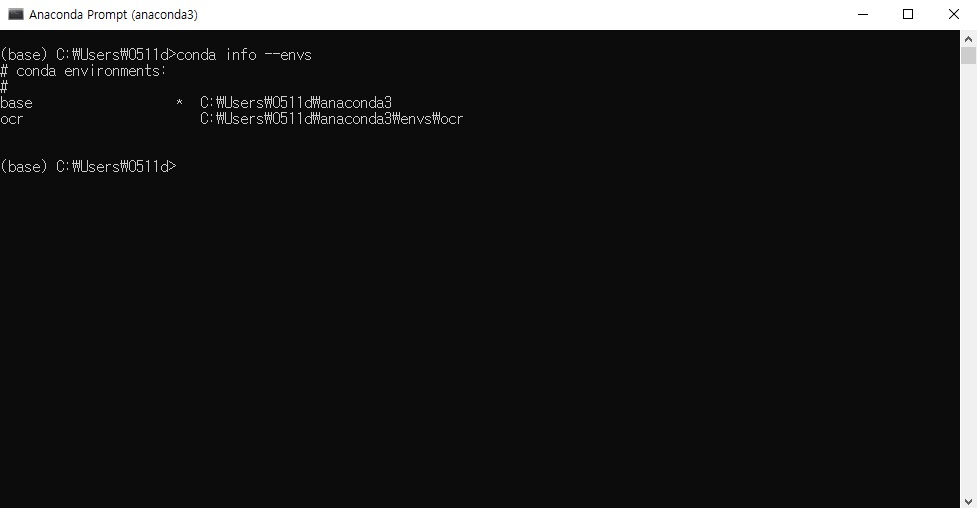
Conda create –n 가상환경명



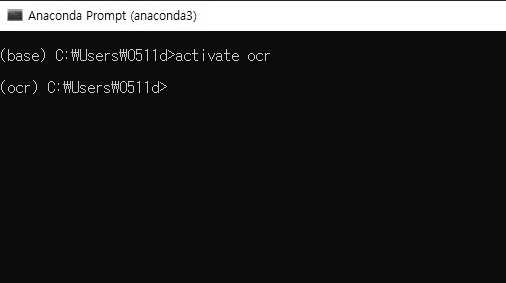


* 생성된 가상환경 리스트 확인 명령어

Conda info --envs



* 가상환경 활성화



* 1. Visual Studio 2019 설치



1. PyTesseract 패키지

[정의]

* Python 용 광학 문자 인식 (OCR) 도구로 이미지에 포함 된 텍스트를 인식하고 읽는다.
* Python-tesseract는 Google의 Tesseract-OCR 엔진 용 래퍼다. 또한 jpeg, png, gif, bmp, tiff 등을 포함하여 Pillow 및 Leptonica 이미징 라이브러리에서 지원하는 모든 이미지 유형을 읽을 수 있으므로 tesseract에 대한 독립 실행 형 호출 스크립트로도 유용하다.

[기능]

* get\_languages: Tesseract OCR에서 현재 지원하는 텍스트를 인식하는 모든 언어(영어, 한글, 일본어 등등)를 반환한다.
* get\_tesseract\_version: 시스템에 설치된 Tesseract 버전을 반환힌다.
* image\_to\_string: Tesseract OCR 처리에서 수정(grayscale or thresholding 등등)되지 않은 출력을 문자열로 반환한다.
* image\_to\_boxes: 인식 된 문자와 해당 상자 경계를 포함하는 결과를 반환한다.
* image\_to\_data: 상자 경계, 신뢰도 및 기타 정보가 포함 된 결과를 반환한다.
* image\_to\_osd: 방향 및 스크립트 감지에 대한 정보가 포함 된 결과를 반환한다.

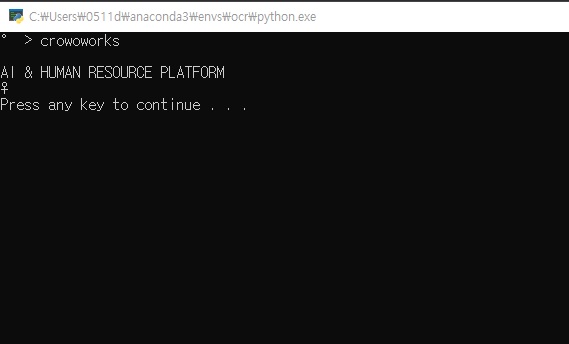
1. PyTesseract을 활용한 이미지에서 텍스트 추출
   1. 영어 텍스트가 있는 이미지



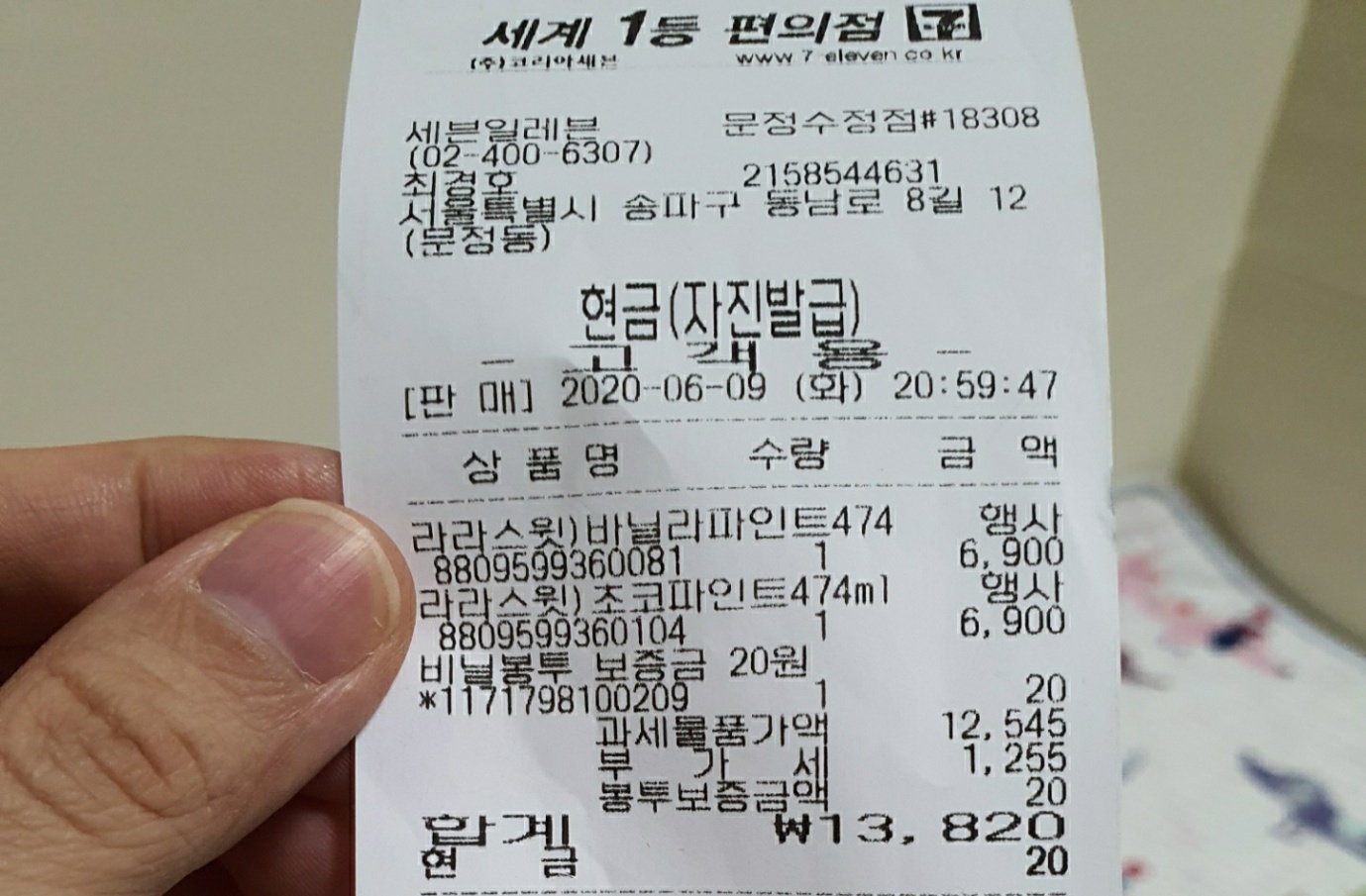
* 영어 텍스트가 있는 이미지에서 텍스트를 추출하는 코드



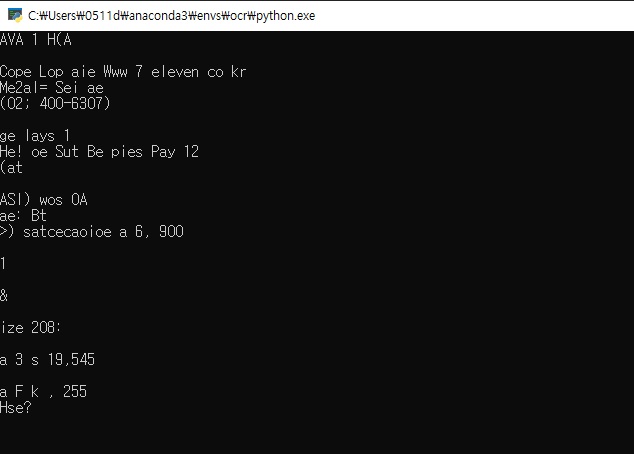
* 영어 텍스트가 출력된 화면



* 1. 한글 텍스트가 있는 이미지



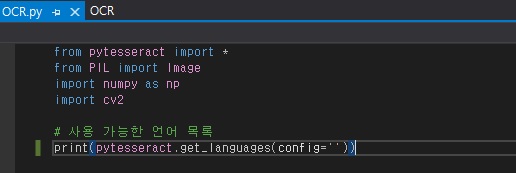
* 한글 텍스트가 출력된 화면

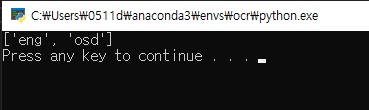


* 한글이 추출되지 않는 문제 발생함

[원인]

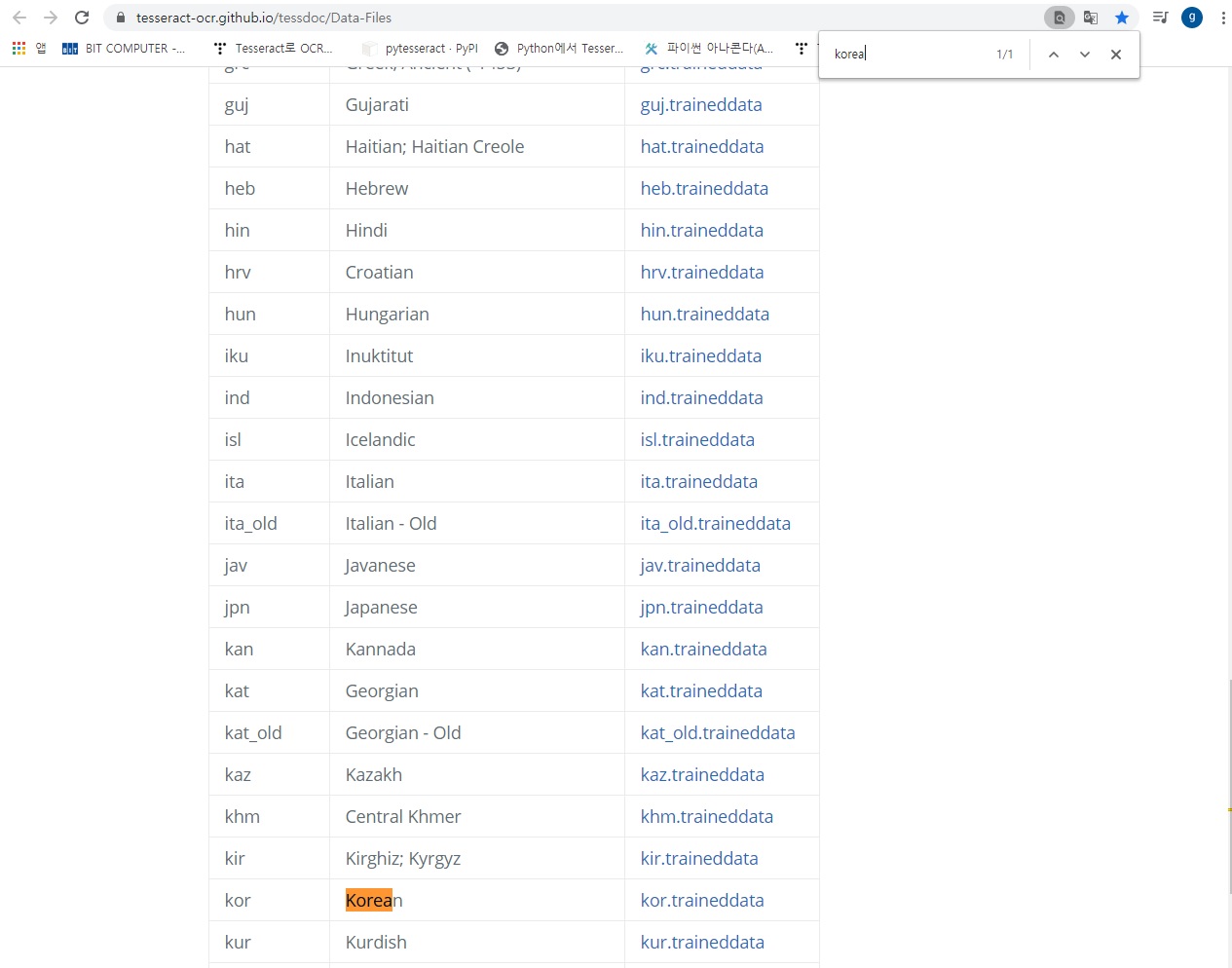
PyTesseract의 get\_languages를 출력해 보면 하단의 이미지와 같이 한글은 미 포함된 것을 알수 있다.

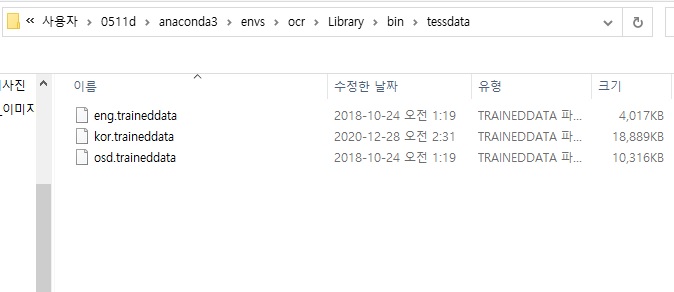




[해결책]

<https://tesseract-ocr.github.io/tessdoc/Data-Files> 사이트에서 원하는 언어의 LSTM 모델 데이터를 다운받은 후 “선택언어.traineddata” 파일을 가상환경의 tessdata 디렉토리에 추가해 준다.

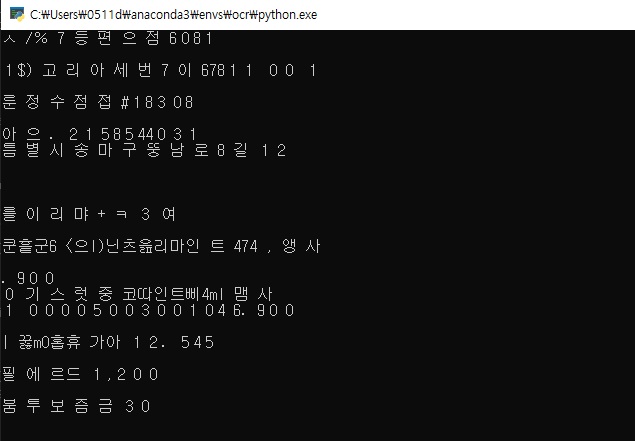




이미지에서 한글을 추출하는 코드 및 실행한 결과

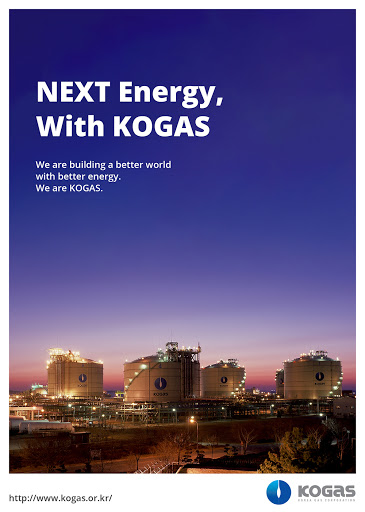
(image\_to\_string 함수의 parameter lang을 추가해야만 정상동작 함)

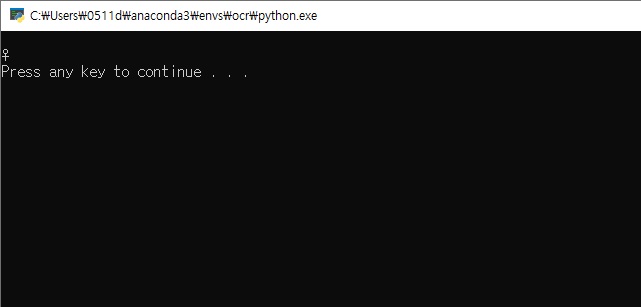




* 1. 텍스트 추출 정확도 향상-1

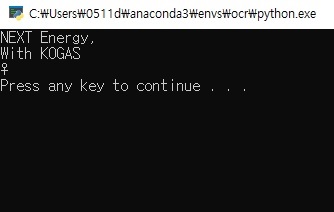
하단의 이미지를 실행시키면 아래 그림과 같이 텍스트를 인식하지 못한다.





아래 그림과 같이 특정부분만이 있을 경우 텍스트가 인지가 된다.

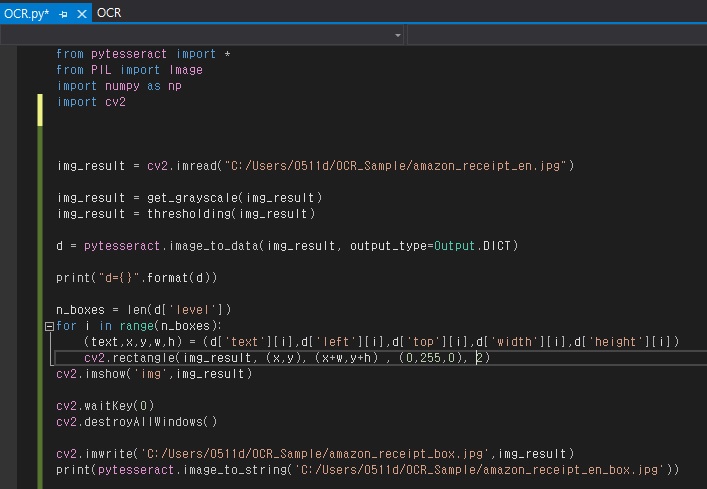




이미지의 전체 영역에서 텍스트를 인지하는것보단, 텍스트를 인지하는 영역을 한정지어서 인지시키는게 인식률이 더 높은것을 알 수 있다

하단의 코드를 실행하여 아래와 같이 텍스트 인지하는 영역을 한정지어서 텍스트를 추출 할 수 있다.

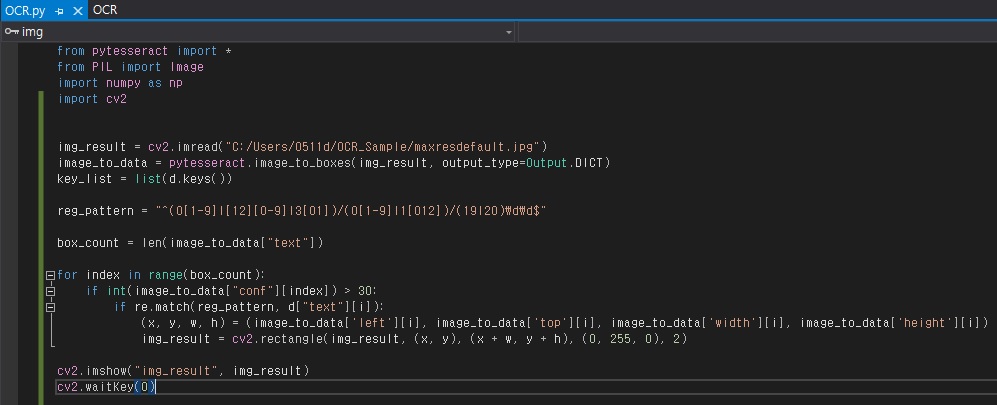
.







추출하고자 하는 텍스트의 조건이 있을 경우 아래 코드와 같이 정규표현식과 일치성의 정확성을 설정하여 추출할 수도 있다.

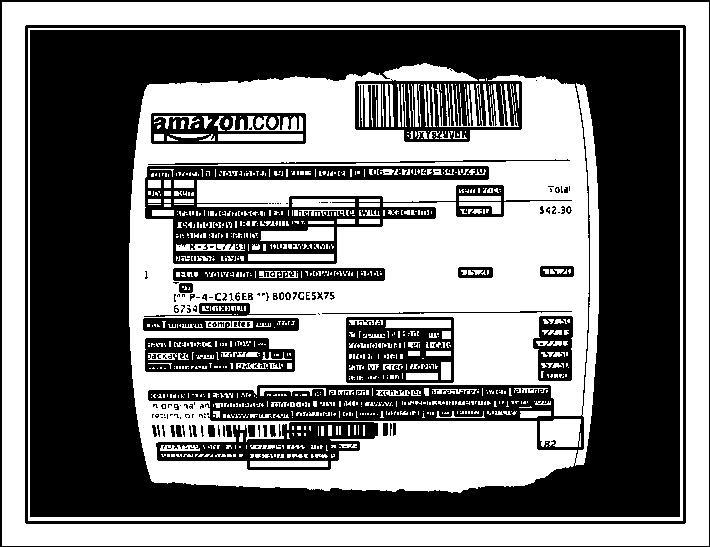


아래 코드는 이미지내의 텍스트가 배치된 각도 및 언어 타입을 추출하는 코드다.

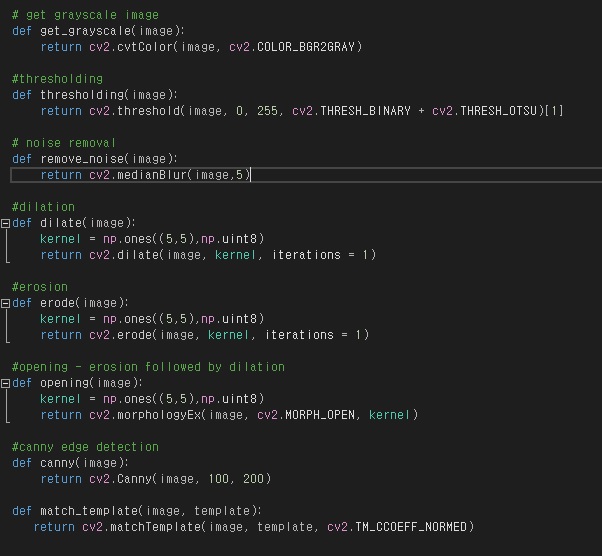


* 1. 텍스트 추출 정확도 향상-2

아래 코드와 같이 grayscale & thresholding를 추가로 처리하면 인식되는 텍스트가 더 많아진다.

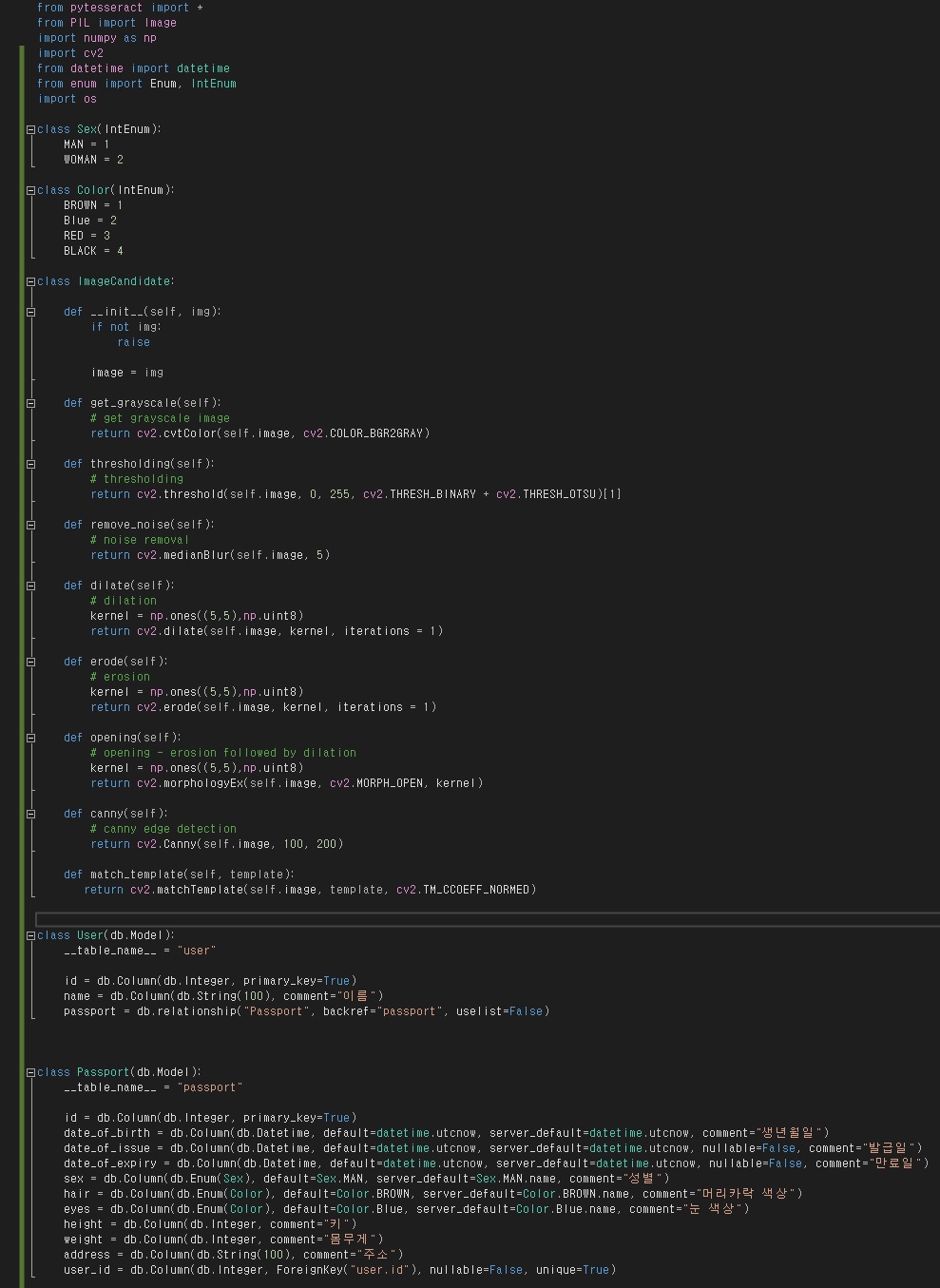


인식률 증가를 위한 대표적인 선처리 방법은 아래와 같다.

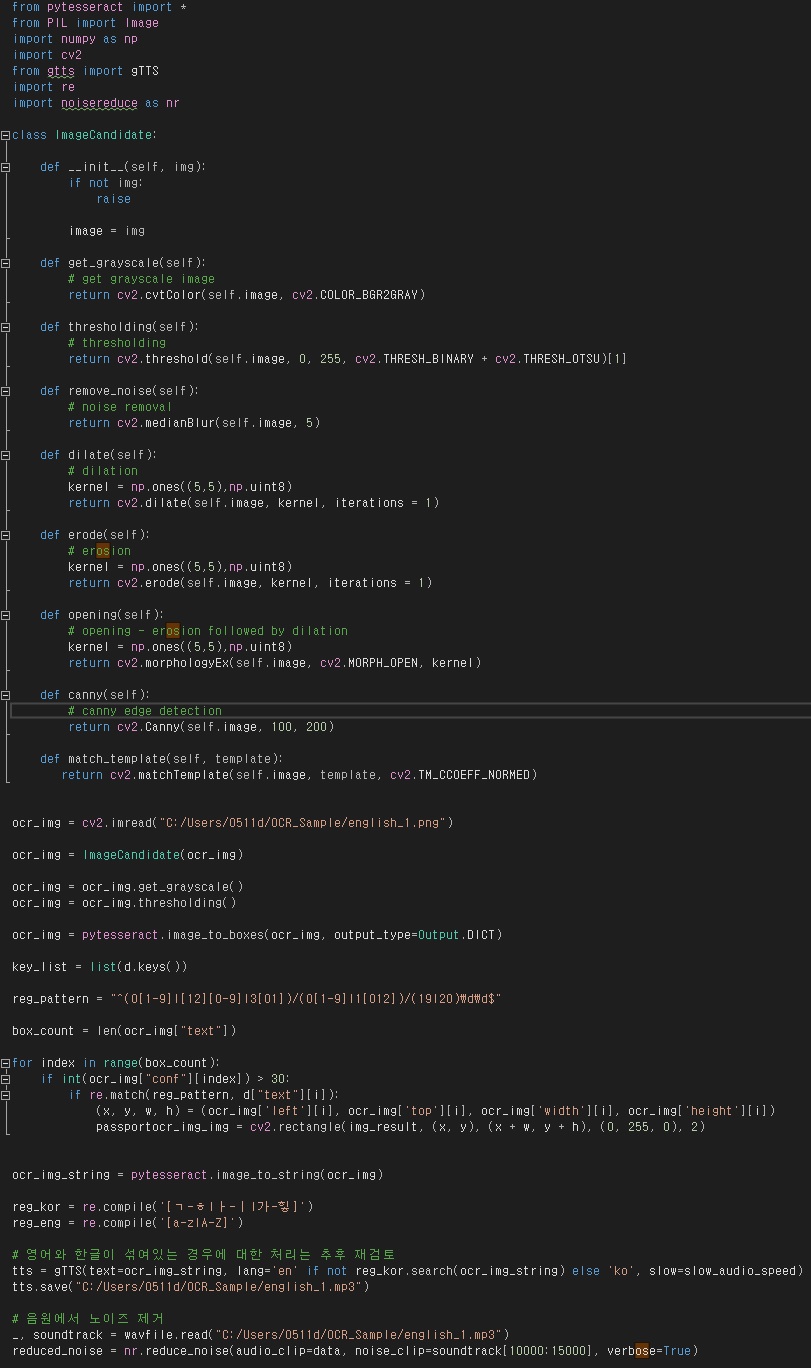


* 1. 상단의 기술검토를 바탕으로 아래와 같은 이미지의 정보를 읽어 DB에 저장 및 화면노출에 사용된 소스





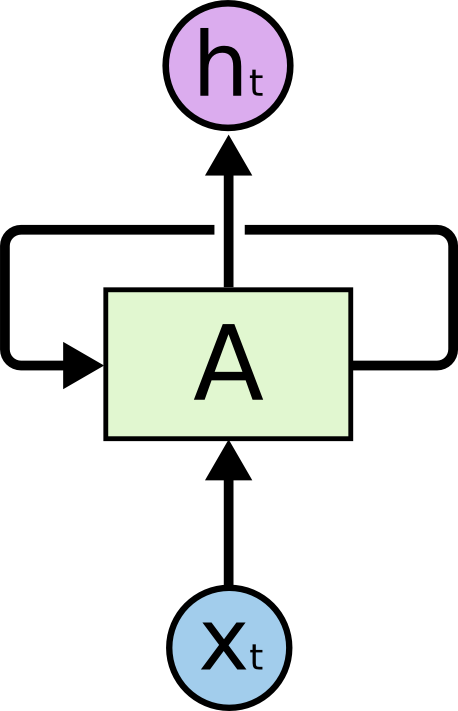


* 1. 이미지의 텍스트를 분석 한 후 음원으로 저장 및 음원 노이즈 제거
  2. 결론
  + 이미지에서 영어 및 한글의 텍스트 추출은 가능
  + 이미지내에서 텍스트 추출은 아래의 조건에 부합 할수록 인식률을 높아지는 것을 알수 있다.
    - 높은 이미지 선명도
    - 이미지 보정으로 불필요한 부분은 제거
    - 이미지내의 글자영역들을 그룹화하여 각 그룹별의 영역만 인지시킬수 있도록 이미지내 글자영역 분리
  + 언어별 LSTM 모델 데이터를 다운받을 수 있어 다국어 추출 가능

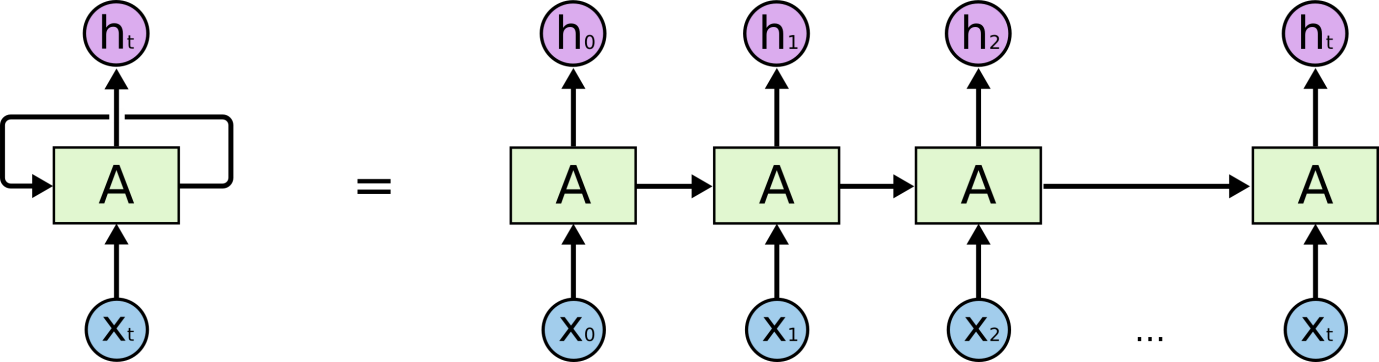
1. 언어 인식률 향상을 위한 머신러닝 학습

8-0. LSTM Network에 대한 이해

* Recurrent Neural Networks

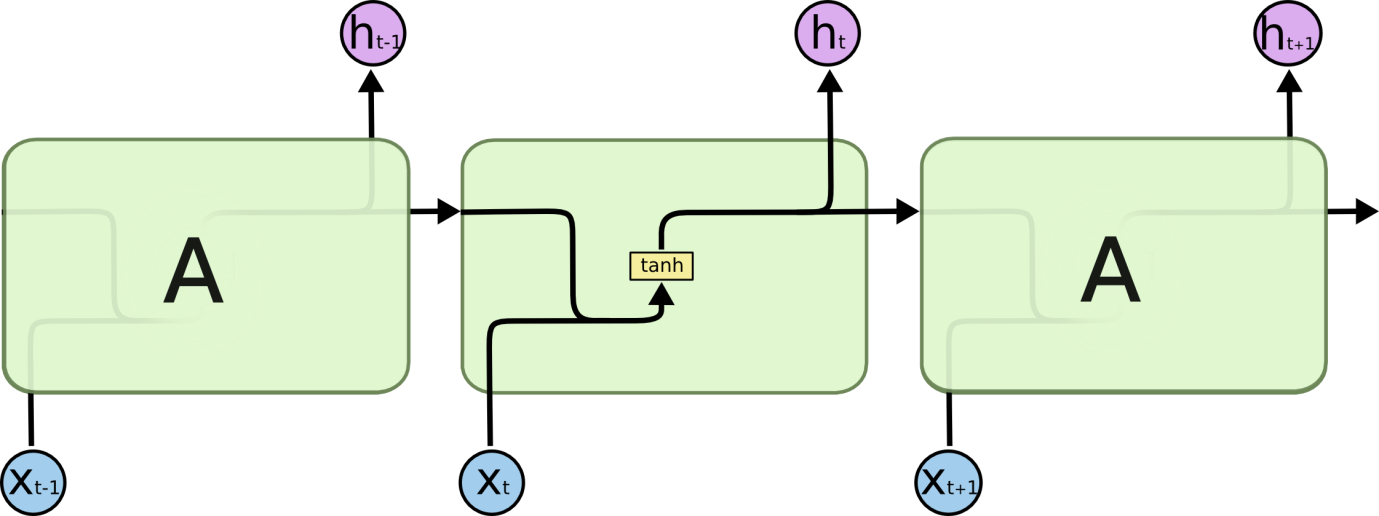


Recurrent Neural Networks have loops

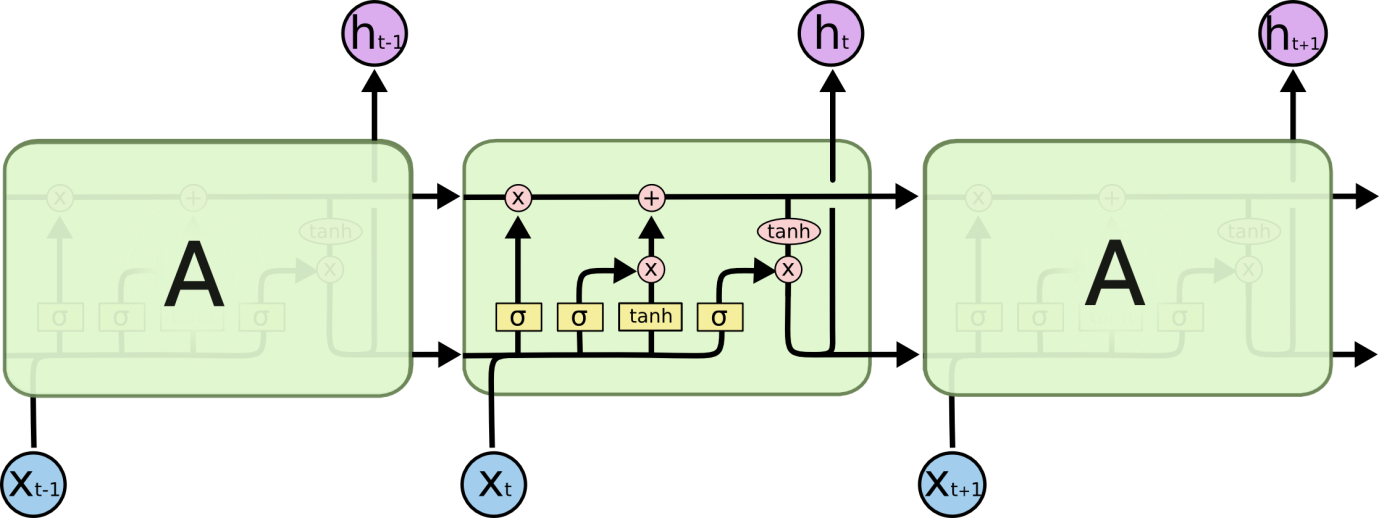


An unrolled recurrent neural network.

* LSTM Networks



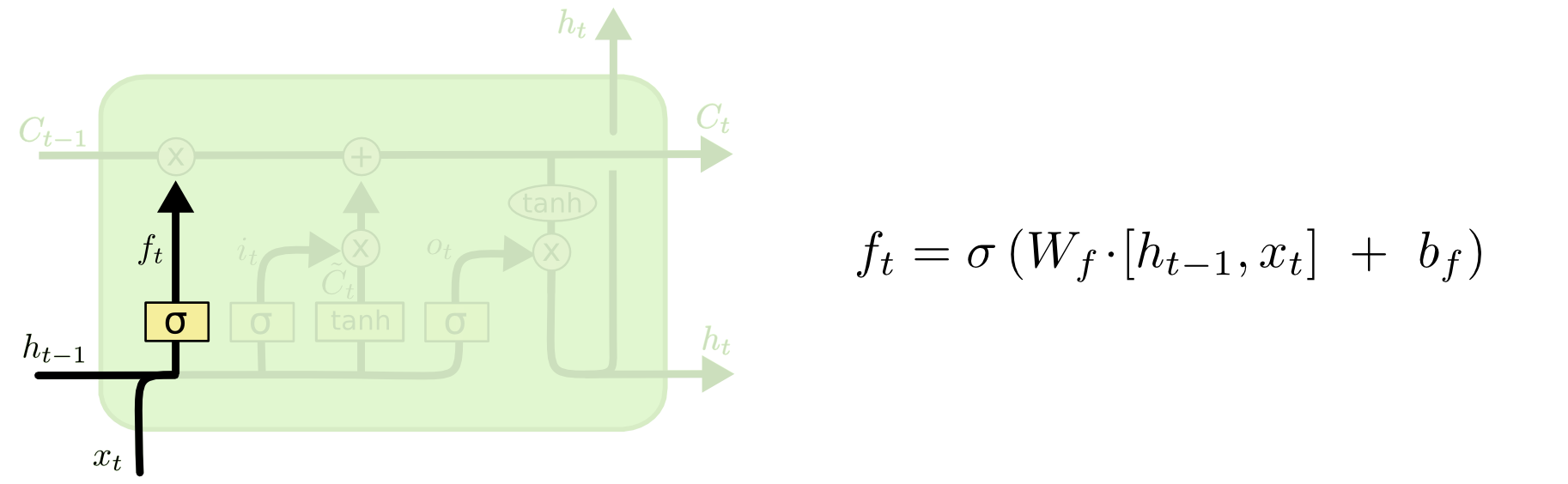
The repeating module in a standard RNN contains a single layer.

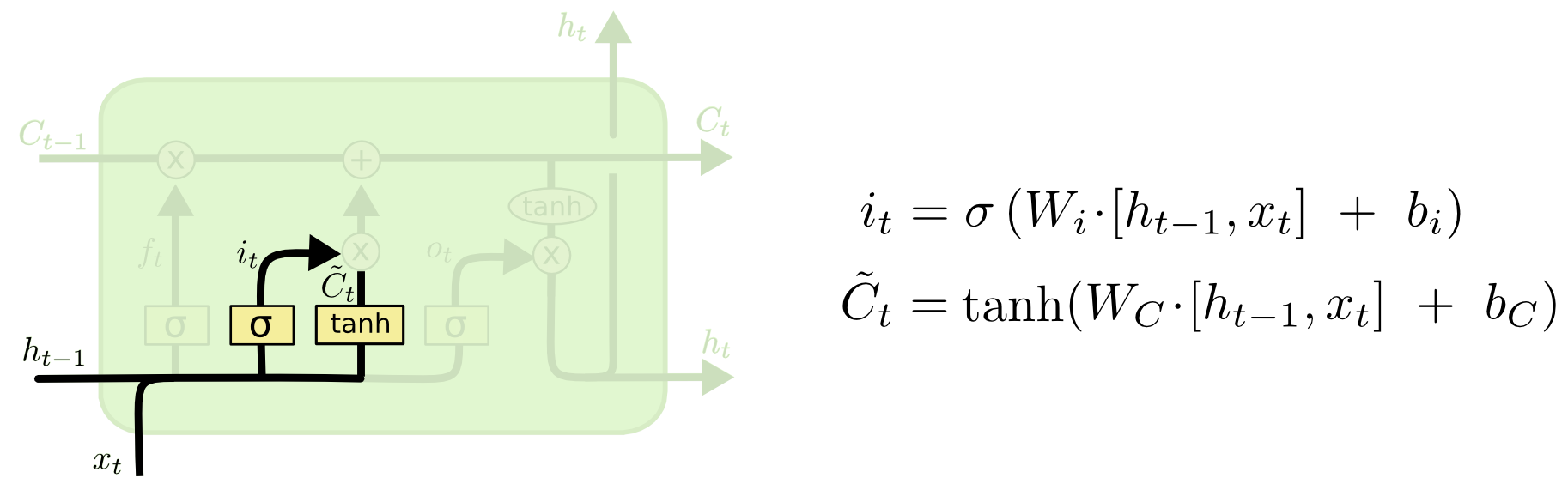


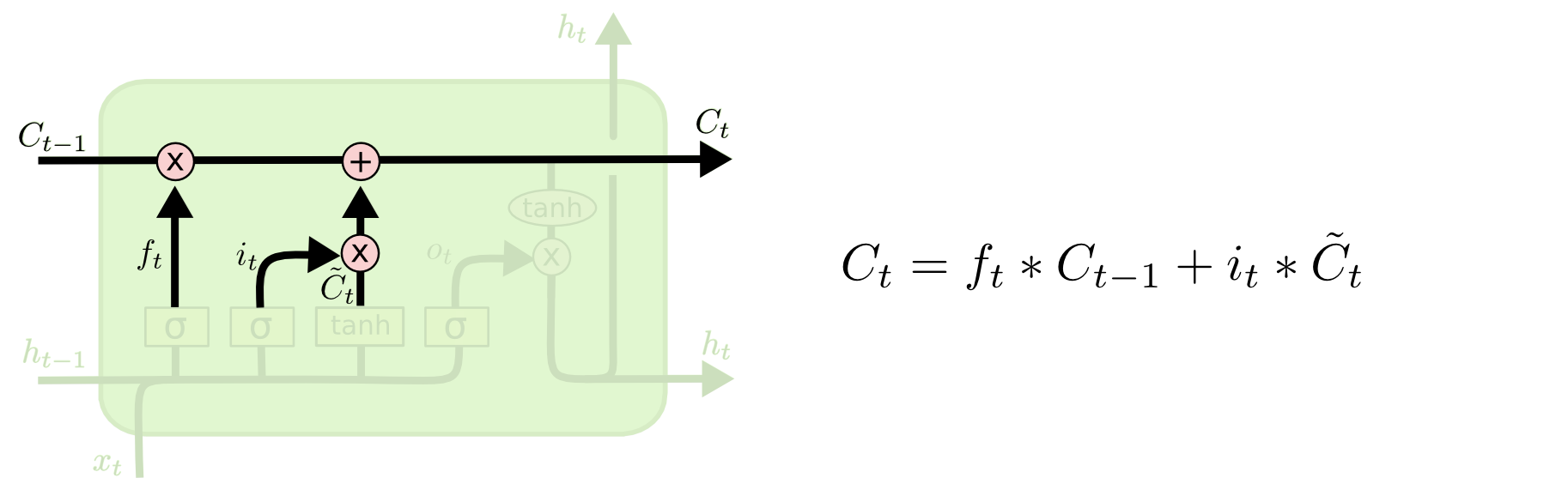


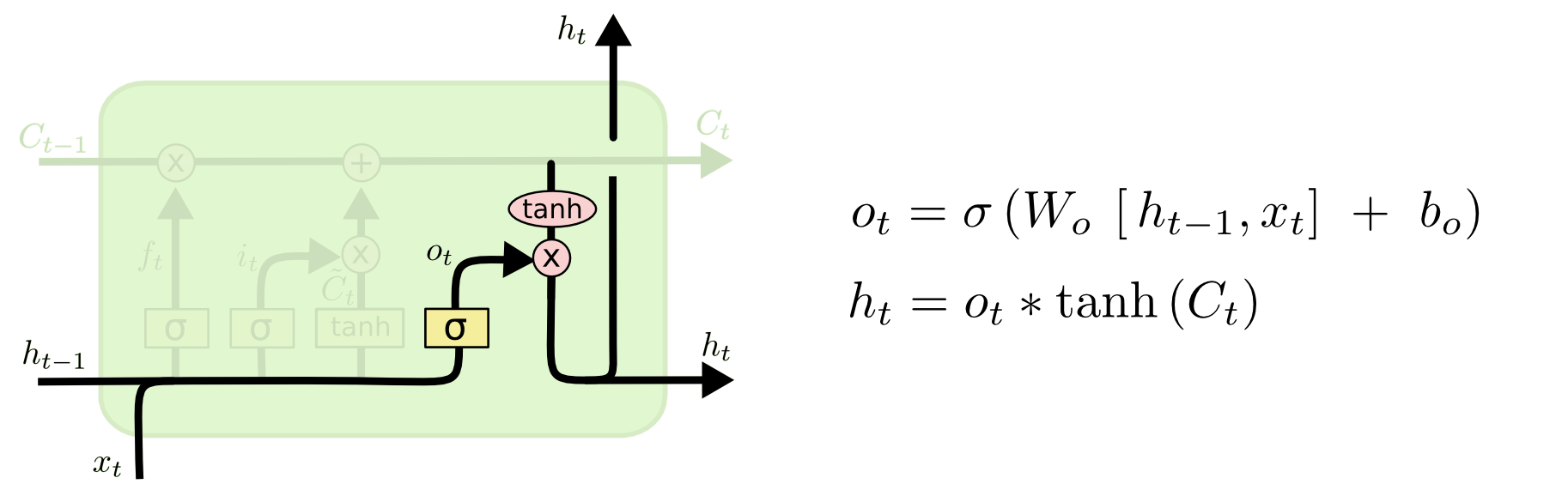
The repeating module in an LSTM contains four interacting layers.

* Step-by-Step LSTM Walk Through

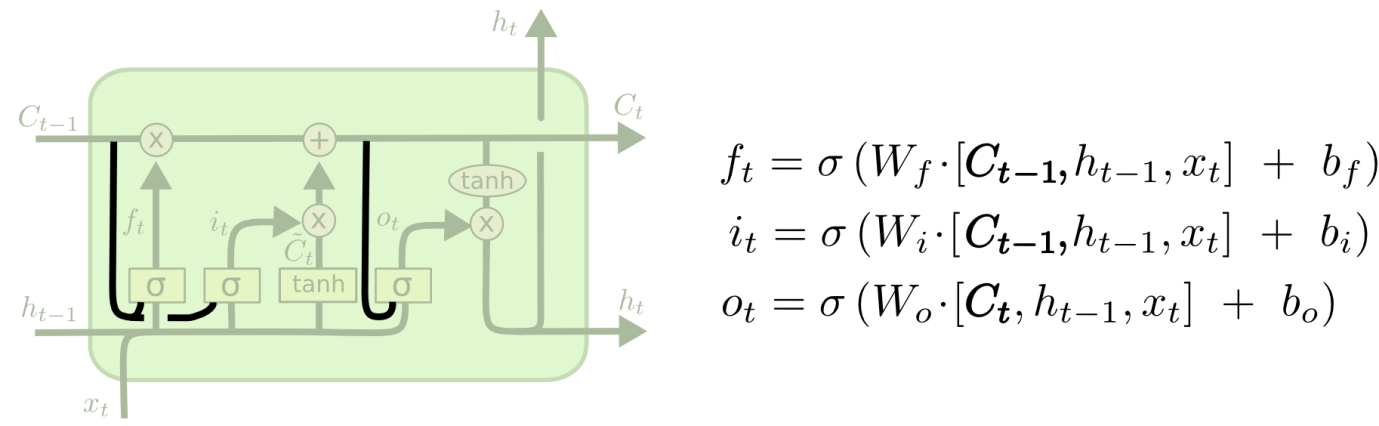


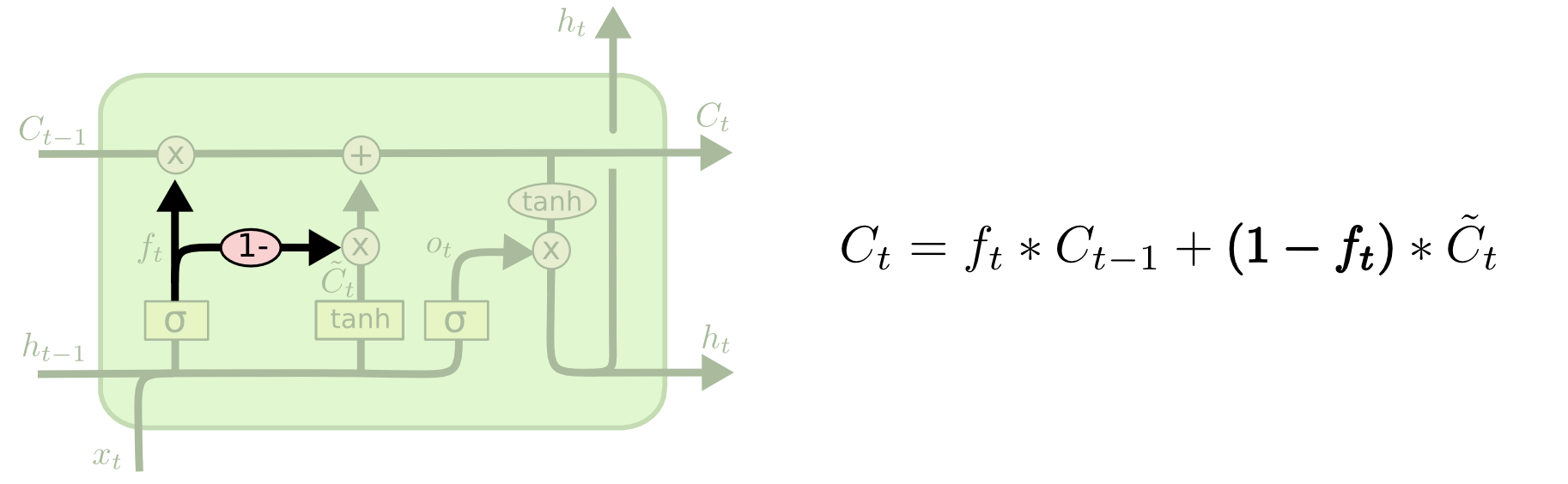


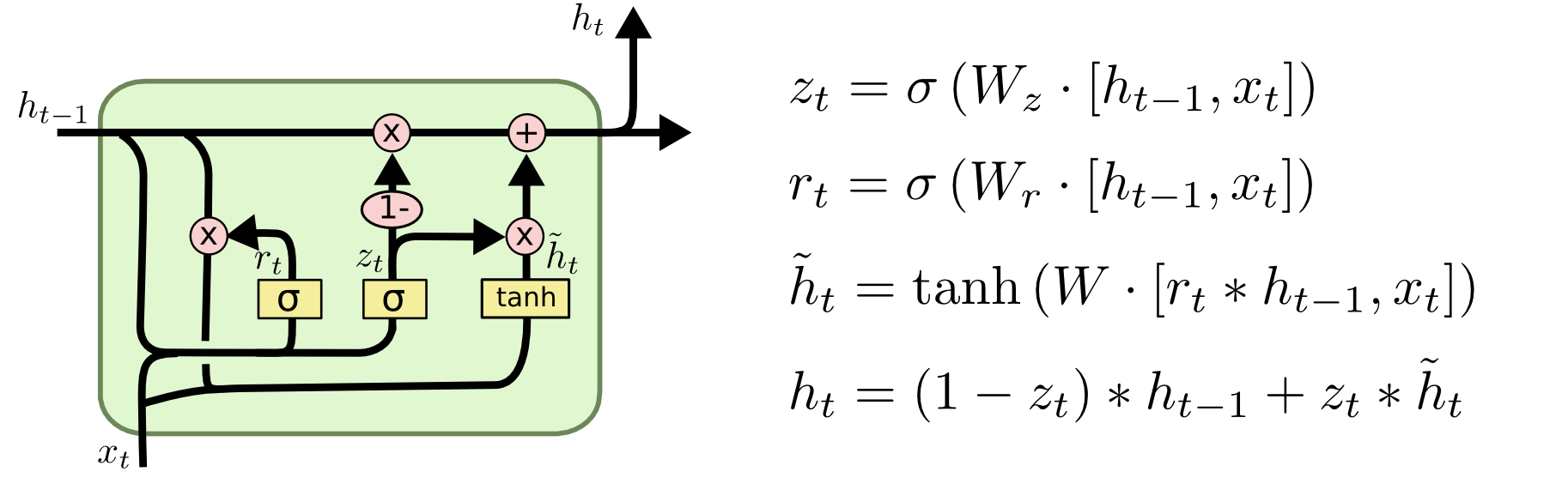




* Variants on Long Short Term Memory





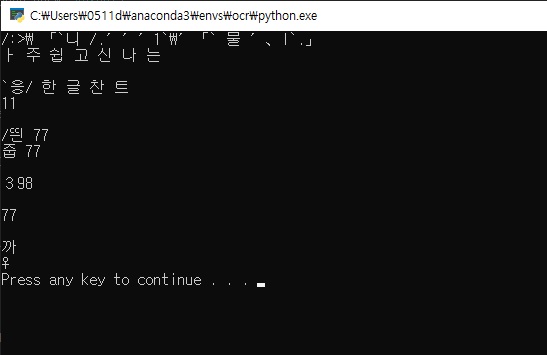


8-1. 한글 인식률 저하 문제

한글 샘플 이미지

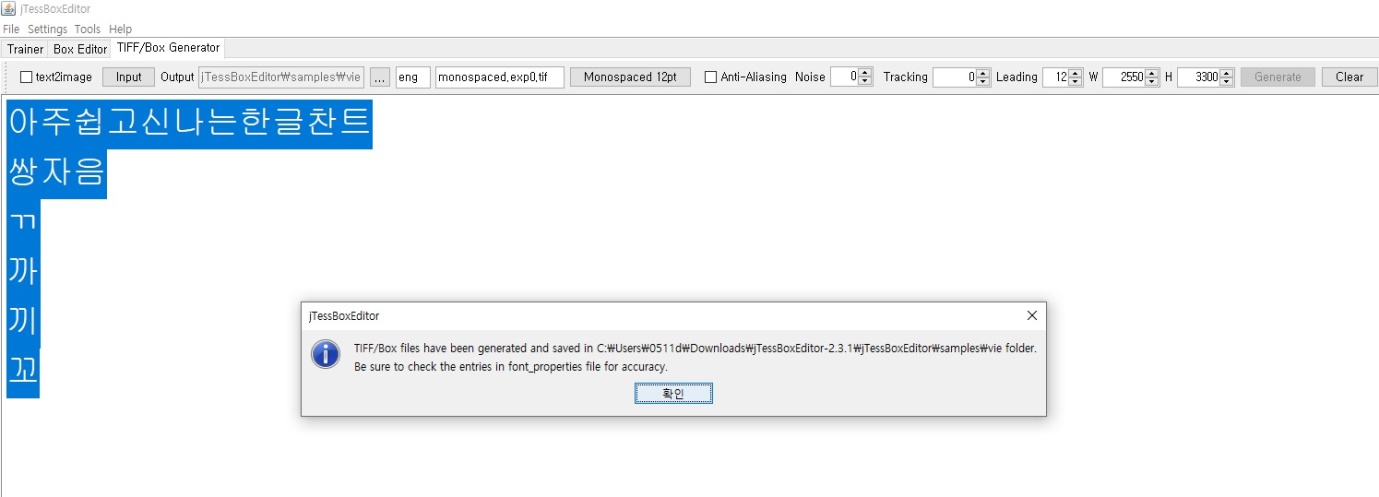


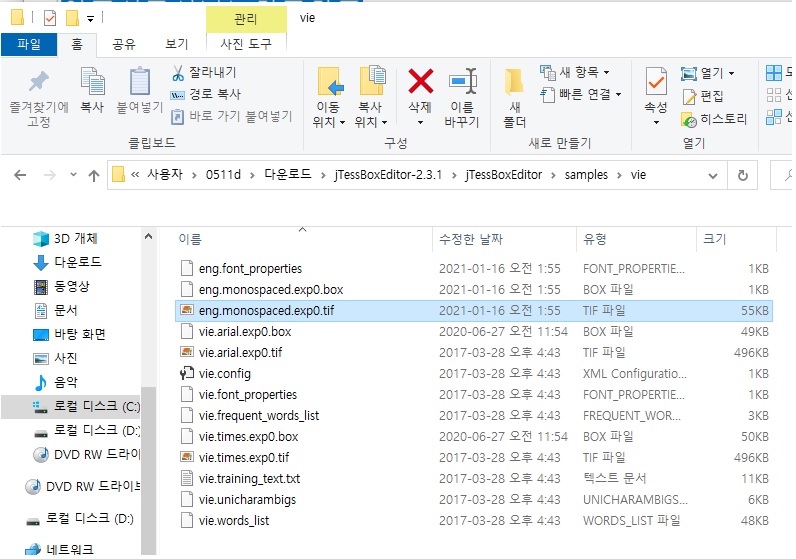
텍스트 추출 결과 쌍자음의 인식률이 떨어지는것을 확인할 수 있다.



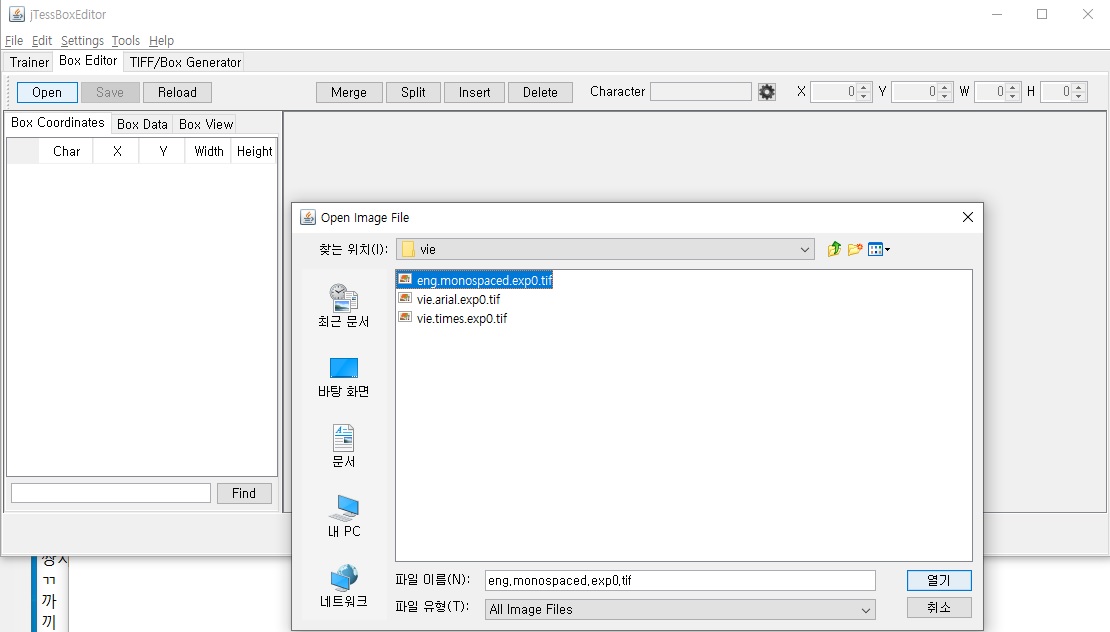
8-2. 한글 쌍자음 학습 딥러닝(LSTM)

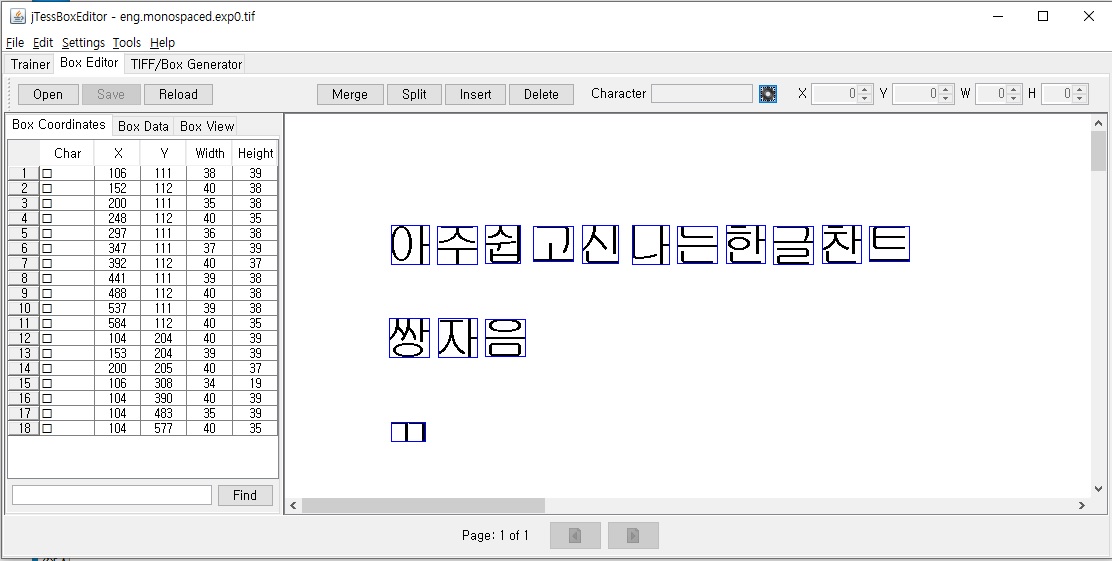
* 윈도우 환경에서 딥러닝(LSTM)을 하기 위해 jTessBoxEditor 선택
* <https://sourceforge.net/projects/vietocr/files/jTessBoxEditor> 사이트에서 jTessBoxEditor 다운로드
* 학습 시키고자 하는 한글을 메모장에 입력한 후 UTF-8로 인코딩 선택 후 저장
* JtessBoxEditor가 설치된 디렉토리에서 train.bat 배치파일 실행
* TIFF/Box Generator 탭 선택 후 하단의 항목 선택
  + 저장될 위치
  + 언어코드
  + 폰트
  + 생성될 tif 파일명 입력
* Input 버튼 클릭하여 저장된 메모장 선택
* Genertate 버튼 클릭 하여 tif 파일 생성



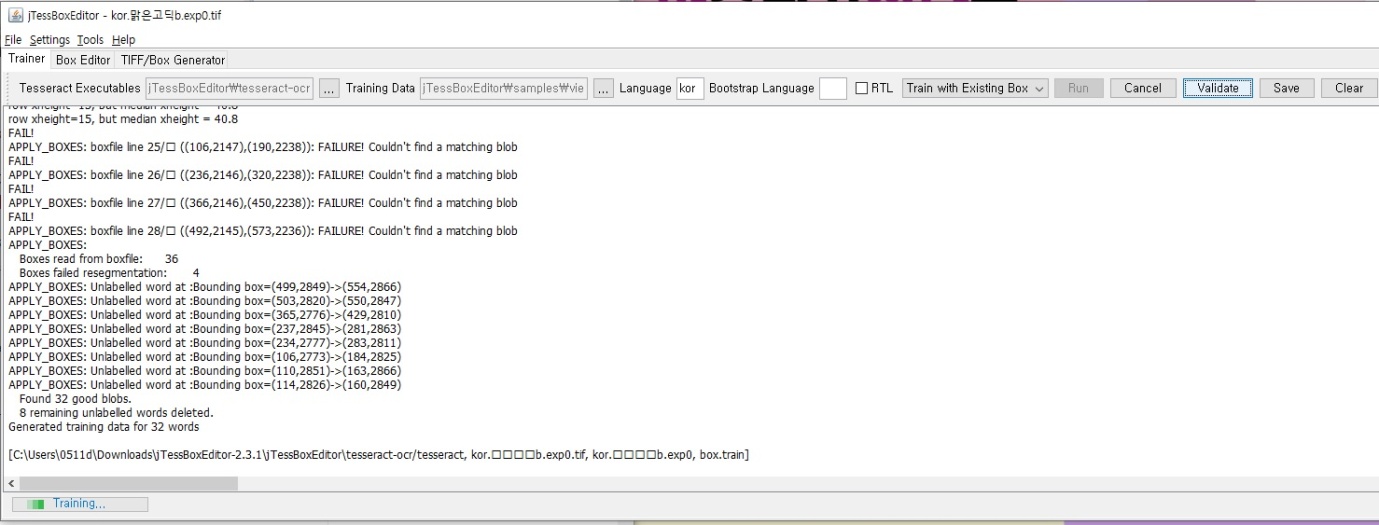


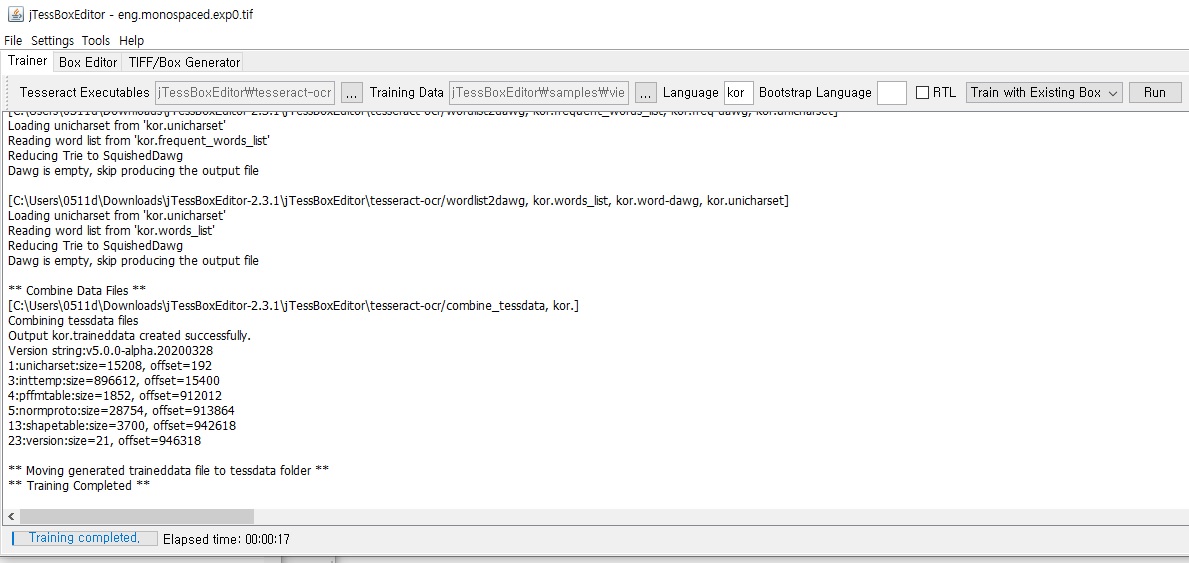
* Box Editor 탭 선택 후 Open 버튼을 클릭하여 생성된 tif 파일 열기



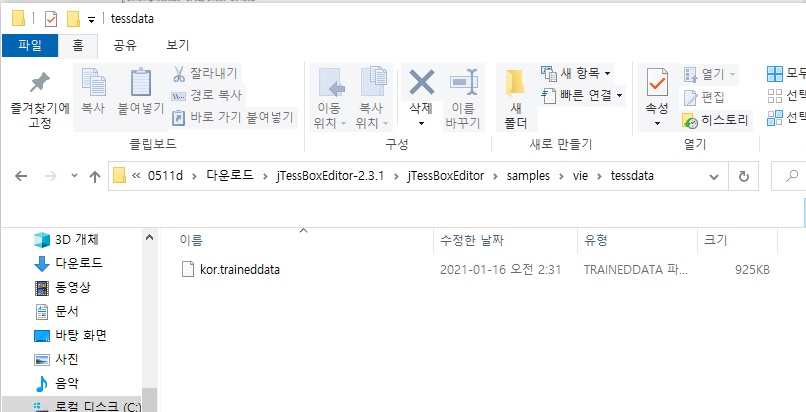


* Trainer 탭 선택 후 원하는 옵션을 선택 한 후 Run 버튼 클릭

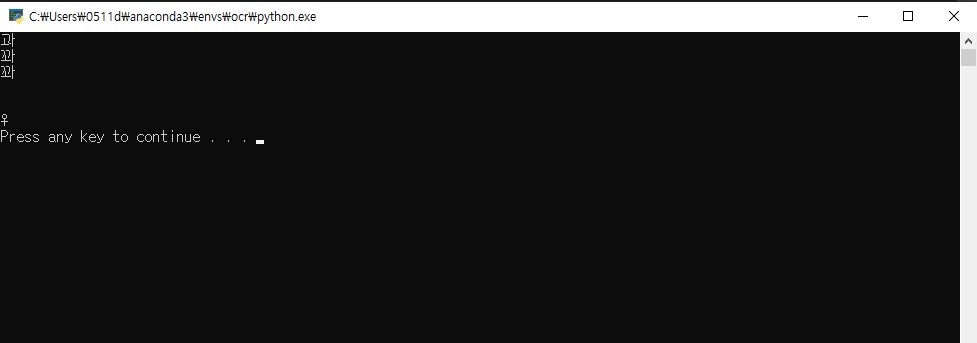




* 새로 생성된 kor.traineddata 파일



* 새로 생성된 kor.traineddata 파일을 사용하여 쌍자음을 인식시켜 본 결과



8-3. 결론

* + JtessBoxEditor를 이용하여 딥러닝(LSTM) 학습시키려는 텍스트를 추가하는것이 가능한 것으로 보여졌으나, 학습 결과로 생성된 “언어. traineddata” 파일로 확인 결과 예상과 다르게 동작함을 확인
  + <https://tesseract-ocr.github.io/tessdoc/Data-Files>에서 다운받은 “언어. traineddata” 파일은 정상동작하는 것에 비해 JtessBoxEditor를 이용한 학습은 비정상동작하는 것으로 판단됨
  + 개인의 학습 딥러닝(LSTM)의 현주소를 좀더 알아보고, 추후 더 자세히 알아본 후 Tesseract Command 명령어 사용 방법으로 재시도 필요

1. 참고자료

* <https://ko.wikipedia.org/wiki/%EA%B4%91%ED%95%99_%EB%AC%B8%EC%9E%90_%EC%9D%B8%EC%8B%9D>
* <https://pypi.org/project/pytesseract/>
* <https://github.com/tesseract-ocr/tesseract>
* <https://tesseract-ocr.github.io/>
* http://colah.github.io/posts/2015-08-Understanding-LSTMs/