**2018학년도 1학기 오픈소스SW프로젝트 최종보고서**

|  |
| --- |
| OCR  (딥러닝을 이용한  이미지-텍스트 변환 알고리즘 및 구현) |

**2018. 06. 19.**

**컴퓨터공학부 오픈소스SW프로젝트**

**탈주예비역 팀**

|  |
| --- |
| **목 차**  **1. 팀, 프로젝트 소개**  1-1. 팀명, 팀원 소개 (고정)  1-2. 팀원 역할 분담 소개 (고정)  **2. 개발과제의 개요**  2-1. 프로젝트의 개요 (고정)  2-2. 개발의 중요성 및 필요성 (고정)  2-3. 개발시 예상되는 파급효과 (고정)  **3. 개발 목표 및 내용**  3-1. 기술개발목표 및 기술의 특성 (고정)  3-2. 개발내용, 개발범위, 핵심개발내용 및 응용기술 (고정)  3-3. 국내외 관련기술 현황 (고정)  **4. 개발환경**    **5. 시스템구조**  **6. 주요기능**  **7. 결론**  **8. 첨부자료 (소스코드 등)**  **※ 소스코드는 필히 첨부 (양이 너무 많은 경우 별도로 압축하여 첨부 가능)** |

**1. 팀, 프로젝트 소개**

* 1. 팀명, 팀원 소개

팀명 : 탈주예비역

1-2. 팀원 소개 및 역할 분담

백승한 (팀장): 전체적인 프로그램 구성 및 Back End 개발

서준덕 (디자인): UI 디자인 및 보고서 작업

서덕진 (개발자): pyqt를 통한 Front End 개발

**2. 개발과제의 개요**

2-1. 개발의 중요성 및 필요성

* 기존에 시중에 사용되어지는 OCR 프로그램은 유료이거나 인식률이 매우 좋지 않은 경형이 있음
* 유료 버전임에도 pdf나 프린트된 서류 등을 문서로 바꿔주는 것이 전부이며 배경과 텍스트가 섞인 이미지의 경우에는 인식률이 좋지 않음
* 이를 보완하여 어떠한 이미지라도 텍스트인 부분을 추출하여 사용자에게 텍스트의 형태로 제공하는 프로그램이 필요함

2-2. 개발시 예상되는 파급효과 등

* 기존에 유료로 사용했어야 하는 OCR 프로그램을 오픈소스화 하는 것으로 인해 많은 일반인들이 혜택을 누릴 것이며 멀티 플랫폼을 지원함에 따라 다양한 환경에서 동작하는 프로그램으로 거듭날 수 있음
* 차후 개발이 계속 진행됨에 따라 웹에서 작동하는 웹 어플리케이 션을 통해 사용자가 브라우저 상에서 선택한 이미지에서 바로 텍스트를 추출하거나 다국어 번역기 등에서 사용될 여지가 있음

**3. 개발의 목표 및 내용**

3-1. 기술개발 목표 및 기술의 특성

OCR 기술은 사용자나 컴퓨터가 이미지 영상 데이터를 디지털화 된 텍스트로 변환하는 것을 기초로 하는데, 해당 프로젝트의 경우 더 나아가 배경과 글자가 결합된 이미지 속에서 텍스트인 영역만 추출하고 해당 텍스트가 어떠한 내용인지 분석하여 반환하는 것을 목표로 함

3-2. 개발내용, 개발범위, 핵심개발내용 및 응용 기술 등

사용자가 임의의 이미지 데이터를 입력하면 해당 이미지 데이터를 전처리하여 컴퓨터가 인식하기 쉽게 변환하여 반환하며 해당 이미지를 미리 만들어 둔 학습데이터를 기반으로 분류하여 텍스트인 이미지를 분리, 해당 이미지를 이용하여 디지털 기반 텍스트로 변환함

3-3. 국내외 관련기술 현황

현재 국내에서 OCR 프로그램을 운영하는 곳은 레티아 한 곳뿐이며 하루 일정횟수의 무료 변환을 지원하지만 회사측에 파일과 함께 메일을 보내야 하는 번거로운 과정이 필요함.

또한 일반 사용자가 아닌 기업을 대상으로 운영하는 곳이기 때문에 전체적으로 툴의 가격이 비싼편임



그림 1-1 국내 유일임을 강조하고 있는 레티아사의 OCR 서비스

해외로 넘어갈 경우 다양한 서비스가 존재하는데 대표적으로 Adobe 사의 Acrobat Pro인데 이것 또한 한달에 일정 금액을 지불하여 구독하는 식의 유료 어플리케이션이며 전반적인 인식률은 굉장히 뛰어난 편이지만 앞서 말했듯이 문서나 pdf파일에 대한 인식률이 좋은 것이지 배경이 섞인 이미지에서는 잘 인식하지 못함

**4. 개발환경**

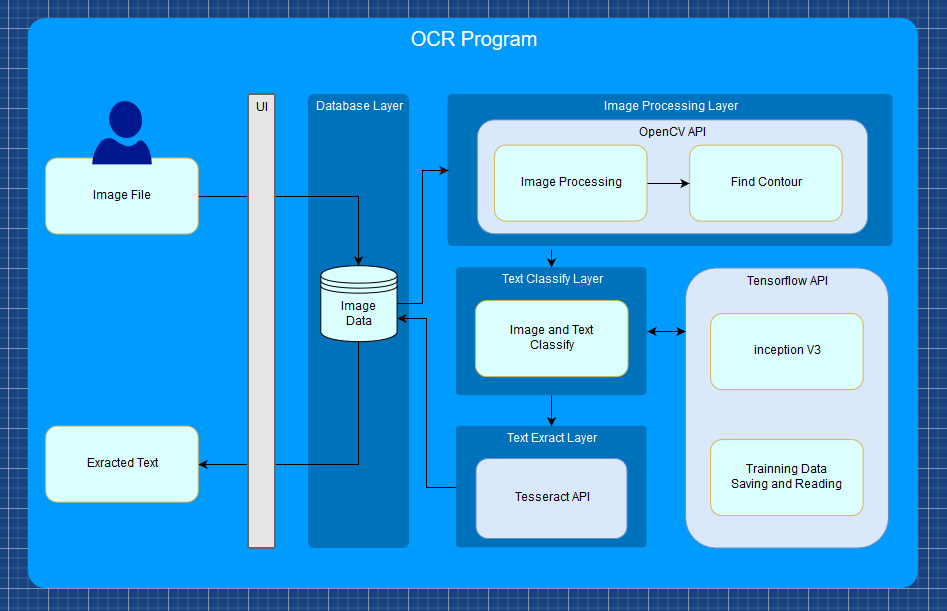
사용 언어 : Python

사용 툴 : Visual studio 2017

사용 라이브러리 :

* OpenCV
* Tensorflow
* PyQt
* Tesseract
* Inception v3

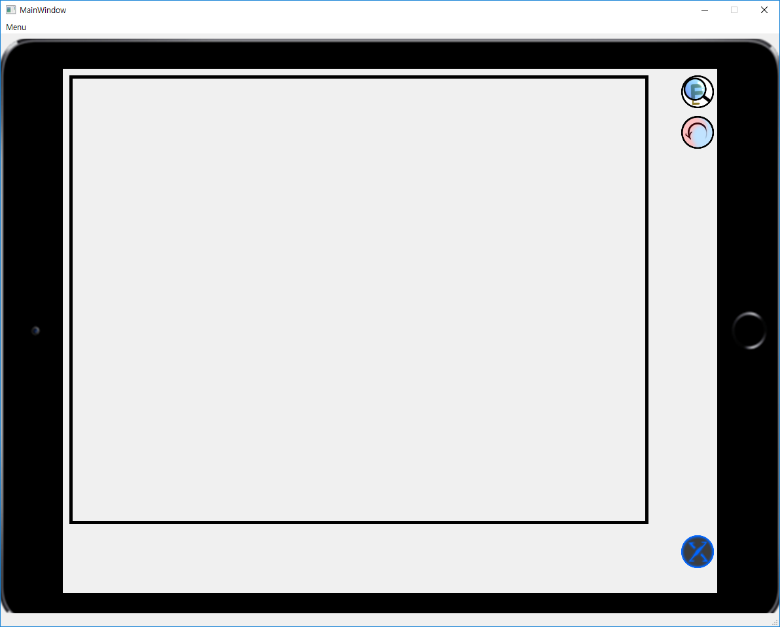
**5. 시스템 구조**



* 사용자가 임의의 이미지 파일을 입력하면 해당 파일을 저장함(User -> DB)
* 저장된 데이터를 OpenCV를 통해 전처리하고 윤곽선을 검출함(DB -> Image Processing)
* 검출된 윤곽선 영역을 Tensorflow로 학습하여 텍스트와 텍스트가 아닌 부분을 구분함(Image Processing -> Text Classify)
* 텍스트로 구분된 영역을 Tesseract를 이용하여 디지털 텍스트로 변환함(Text Classify -> Text Extract)
* 디지털 텍스트를 저장하고 사용에게 반환함(Text Extract -> DB -> User)

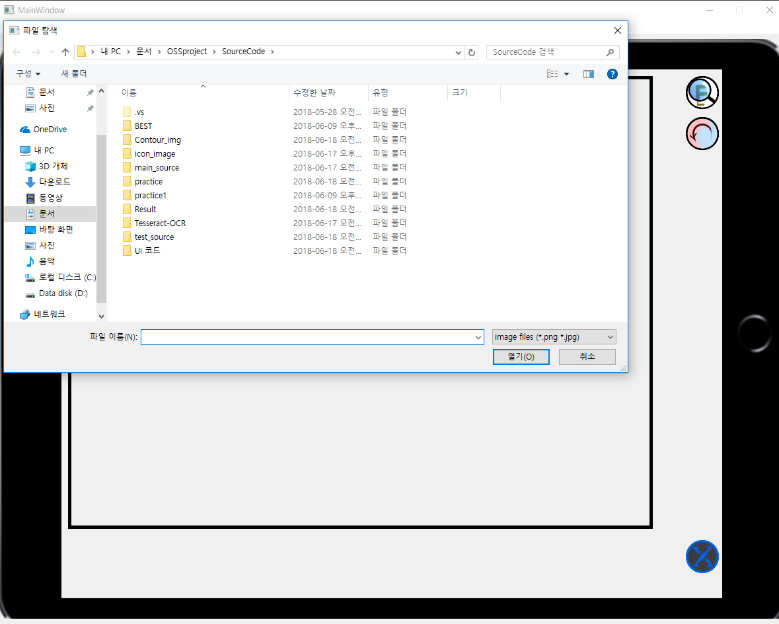
**6. 주요기능**

1) 최초 실행화면



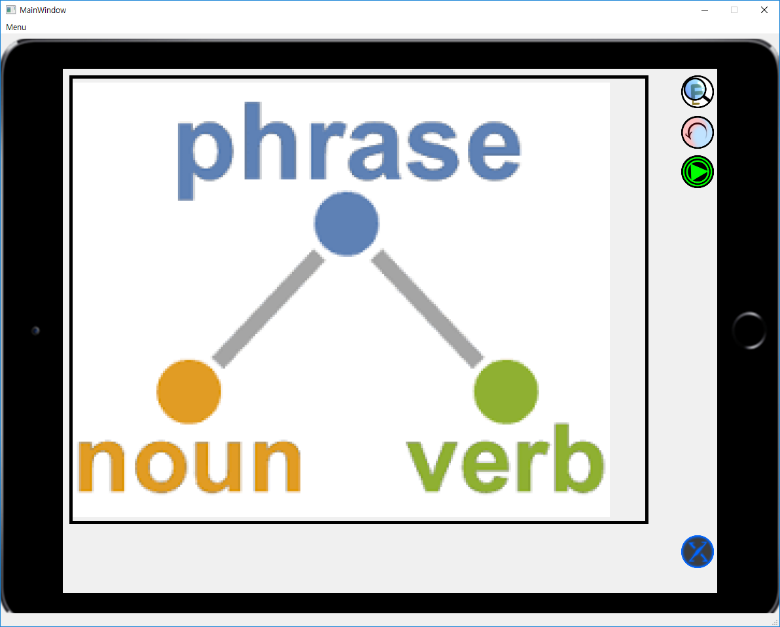
- 최초 실행 시, 좌측 이미지처럼 이미지가 출력될 영역과 ‘파일 탐색’, ’초기화’, ‘종료’ 버튼을 표시함

2) 파일 탐색



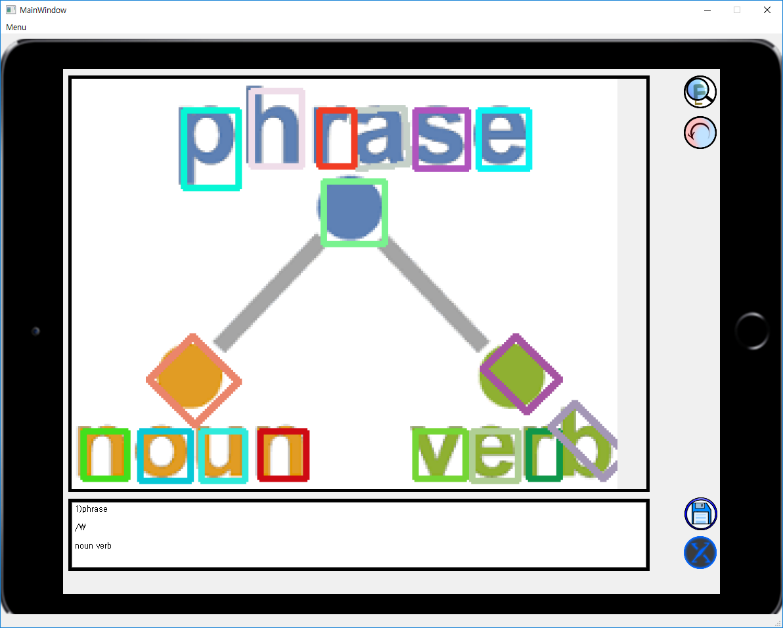
- 파일 탐색 버튼을 누르면 윈도우 파일 탐색기를 이용하여 파일을 선택함

3) 이미지 파일 확인 및 실행



- 사용자가 불러온 이미지를 표시하고 우측 버튼에 ‘실행’ 버튼을 활성화시킴

4) 실행 결과 화면



- 알고리즘이 텍스트로 구분해낸 영역을 화면에 보여주고 실제 텍스트로 인식한 부분을 하단의 텍스트 필드를 이용하여 출력해줌

**7. 결론**

프로젝트 기획 및 조사 단계에서 다양한 논문과 참고 자료를 읽으며 ‘왜 자료는 이렇게 많은데 실제 개발된 사례는 잘 없을까’ 하는 고민을 한적이 있었는데 실제 개발 과정에서 본 프로그램의 완성도를 높이기 위해 얼마나 많은 연구가 필요한지 느낄 수 있었다.

첫 번째로는 이미지의 해상도의 문제이다. 애초에 입력된 이미지의 해상도가 좋지 않으면 평범한 방법으로는 인식률이 대폭 감소하게 된다. 그렇지만 해당 이미지의 해상도를 향상시키기 위해서는 별도의 알고리즘이 요구되었고, 해당 알고리즘을 통해서도 손실된 프레임을 완전히 복구시킬 수 없다는 것을 알게 되었다.

두 번째는 컴퓨터 성능에 따른 연산 속도 문제이다. 컴퓨터 사양이 급격히 증가하는 과정 속에서 프로그램의 최적화에 대한 우려가 줄어드는 듯했으나 이번 프로젝트를 통해 복잡한 연산으로 컴퓨터의 메모리가 얼마나 소모되고 최적화를 통해 이를 보완해야 한다는 점을 알게 되었다.

마지막으로 한글의 인식률 문제이다. 애초에 자료 검색과정에서 생각보다 많은 양의 딥러닝을 통한 한글 인식에 대한 논문을 볼 수 있었다. 해당 논문에서는 보통 한글은 자모음과 받침으로 이루어져 기타 다른 언어와 달리 한 단어를 인식함에 있어서 큰 어려움을 겪는다 하였는데 실제로 윤곽선을 추출하는 과정에서 자모음 및 받침이 분리되는 경우가 많았고 이를 재결합시켜 주는 과정이 필요하였으며 손글씨의 경우 비슷하게 한 개 이상의 글자와 유사성을 띌 수 있기 때문에 곤혹을 겪었다.

최종적으로 개발에 많은 난항을 겪었지만 해당 기술의 응용과 발전에 따라 자동차 번호판 인식이나 다국어 번역기 등에서 보다 효율적이고 좋은 인식률을 보일 수 있을 것이며, 더 나아가 인공지능이 시각적 데이터를 통해 텍스트를 이해할 수 있을 것이다.