**테스트 보고서**

**1. Tesseract-ocr 기본**

**2. 기본글꼴 학습**

**3. 두 줄 인식**

**4. Noise제거**

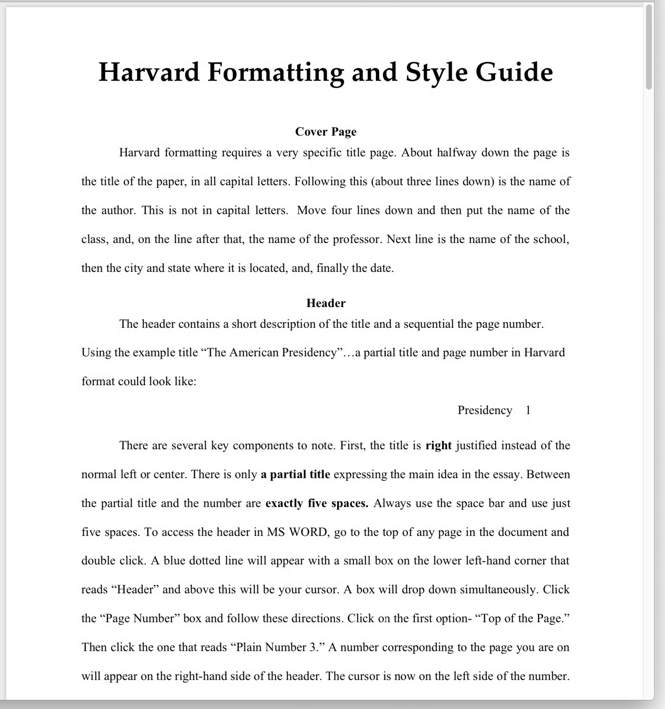
**5. 영상처리코드**

**6. Image Box 파일 학습**

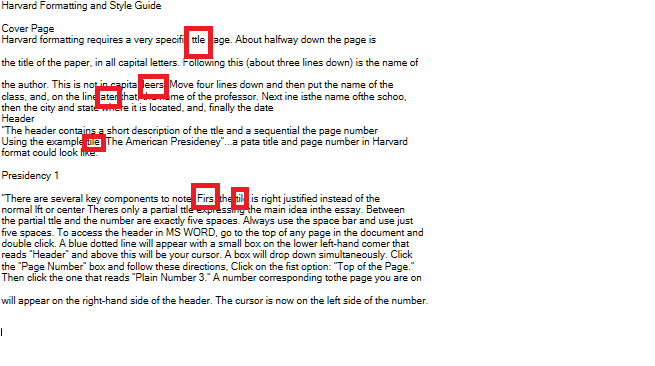
**1. 기본 Tesseract-ocr**

**1.1 영어**

**원본이미지(기본 문서 이미지)**

****

**OCR 후**

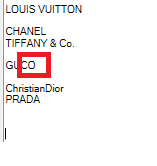
****

**원본 이미지(영어 포스터)**

**건물, 실외, 인도, 표지판이(가) 표시된 사진

자동 생성된 설명**

**OCR 후**

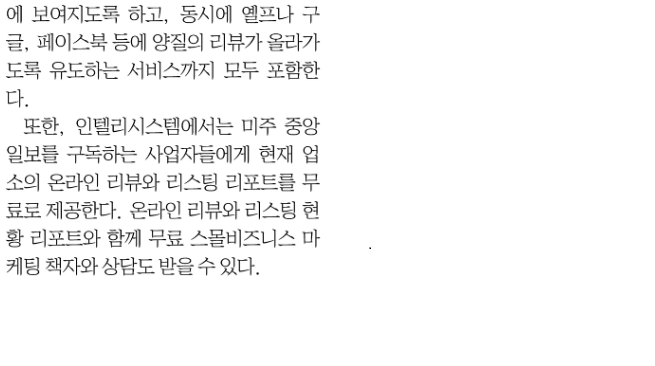
****

**영어 기반의 OCR 프로그램이라 그런지 위 사진들의 체크포인트 몇가지를 제외하면 영어의 인식률은 매우 높았습니다.**

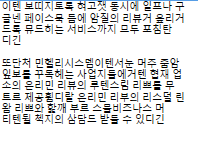
|  |  |
| --- | --- |
| **상 태** | **인식률** |
| **기본 tesseract ocr** | **89%** |

**1.2 한 글**

**원본 이미지**

****

**Ocr 후 이미지**

****

**영어 기반의 OCR프로그램이기 때문에 한글 인식에 굉장히 취약하였다.**

|  |  |
| --- | --- |
| **상 태** | **인식률** |
| **기본 tesseract-ocr** | **46%** |

**2. 기본 글꼴 학습**

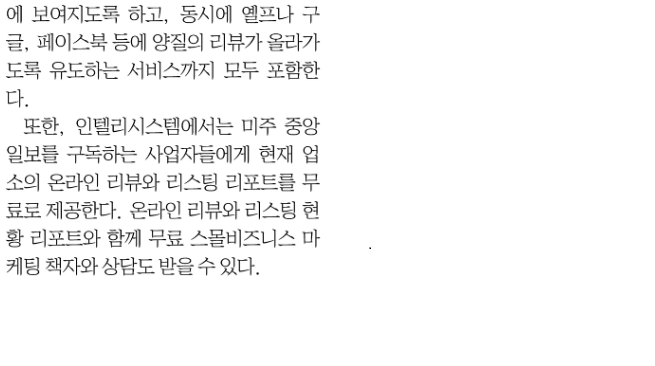
**2.1 영어**

**결과 : 영어의 경우 글꼴 추가의 효과는 보지 못하여 오인식의 문제가 글꼴의 문제가 아니라는 것을 확인하였습니다.**

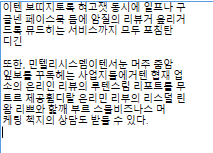
|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **89%** |
| **글꼴 추가** | **89%** |

**2.2 한글**

**원본이미지**

****

**OCR 후**

****

**결과 : 한글의 경우 인식률이 살짝 올라갔지만 기본 tesseract 데이터에 글꼴이 없어서 인식률이 낮게 나왔던 경우를 제외 하고는 글꼴의 추가 효과 보다는 글자들을 데이터에 입력시킨 학습효과 때문에 인식률이 올라간 것으로 보인다.**

|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **46%** |
| **글꼴 추가** | **56%** |

**3. 두 줄 인식을 한 줄 인식으로 변경**

**3.1 영어**

**영어는 한 줄로 인식되기 때문에 효과가 없었다.**

**3.2 한글**

**한글은 받침이 있기 때문에 가끔 두 줄로 인식되는 경우가 생긴다.**

**해결 > block라인에서 라인을 구분하는 부분에서 강제로 한 줄로 인식하게 만들었다.**

**결과 : 두 줄로 인식되어 인식이 잘 되지 않던 부분들이 한 줄로 인식되어 인식률이 올라갔다.**

|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **46%** |
| **글꼴 추가** | **56%** |
| **한 줄 인식** | **64%** |

**4. noise제거**

**이미지 파일의 조그마한 노이즈나 티끌에 인식의 오류가 생긴다.**

**해결> 해결 : blob내의 outline을 확인하여 noise성분을 제거하는 방법**

**이미지 전체의 blob에서 hole을 제외한 outline의 합에서 평균을 낸 후 noise성분만 제거 할 수 있을 정도의 값인 4로 나누었다.**

**1. 영어**

|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **89%** |
| **글꼴 추가** | **89%** |
| **Noise 제거** | **93%** |

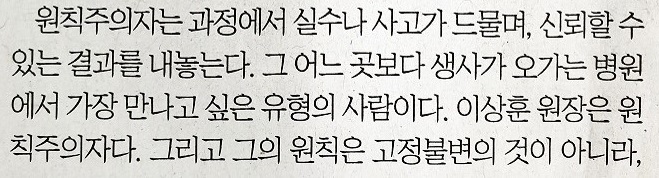
**결과 : ‘l’이 ‘i’로 인식되거나 ‘j’로 인식되는 문제가 존재하였으나 노이즈 제거로 인해 이 부분이 보완되어 인식률이 올라갔다.**

**2.한글**

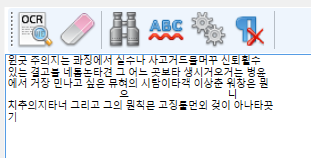
|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-cor** | **46%** |
| **글꼴 추가** | **56%** |
| **한 줄 인식** | **64%** |
| **Noise 제거** | **69%** |

**결과 : 한글의 경우 ‘ㅈ’이 ‘ㅊ’으로 바뀌거나 ‘ㅣ’가 ‘ㅏ’나 ‘ㅓ’로 바뀌는 문제가 존재하였으나 노이즈 제거 후 인식률이 상승하였다. 노이즈 제거 최소 크기를 ‘ㅊ’이나 자음의 점들보다 높게 기준을 잡아 완벽한 제거는 되지않았지만 인식률이 상승하였다.**

**원본 이미지파일**

****

**오류 수정 전**

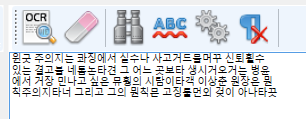
****

**위의 이미지를 보면 ‘유형’, 원장’ ‘원칙주의자’의 단어에서 두 줄로 인식되서**

**이미지 변환이 두 줄로 된 경우를 볼 수 있다.**

**또 ‘원장’이나 ‘가장’의 ‘ㅈ’이 ‘ㅊ’으로 인식된 것도 볼 수 있다.**

**오류 수정 후**

****

**3,4번 과정을 통해 두줄 인식이나 노이즈제거가 효과적으로 적용된 것을 볼 수 있다.**

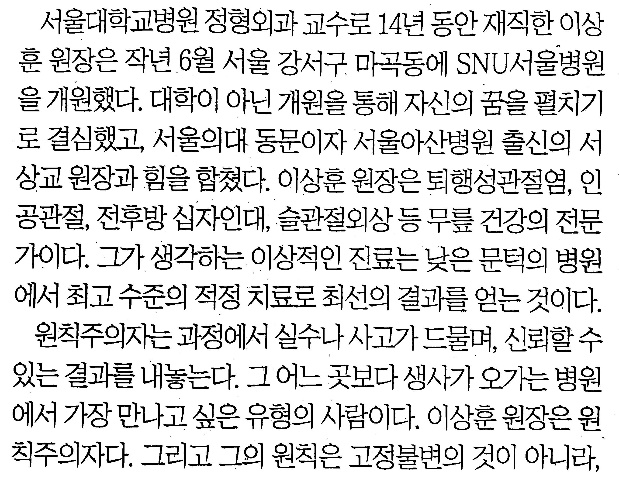
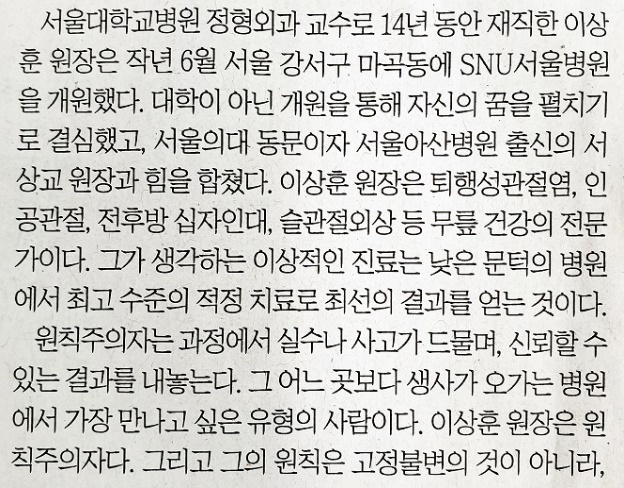
**5. 영상 처리 코드**

**이미지 파일 중 영상이 어둡거나 배경의 색깔 때문에 글자를 구분하기 힘든 경우가 존재하였다.**

**> 해결 : 1. 히스토그램 스트레칭: 이미지의 명암 대비 향상**

**2. 이미지 이진화: 배경은 흰색, 글자는 검정색으로 이진화**

**영상처리 전 후**

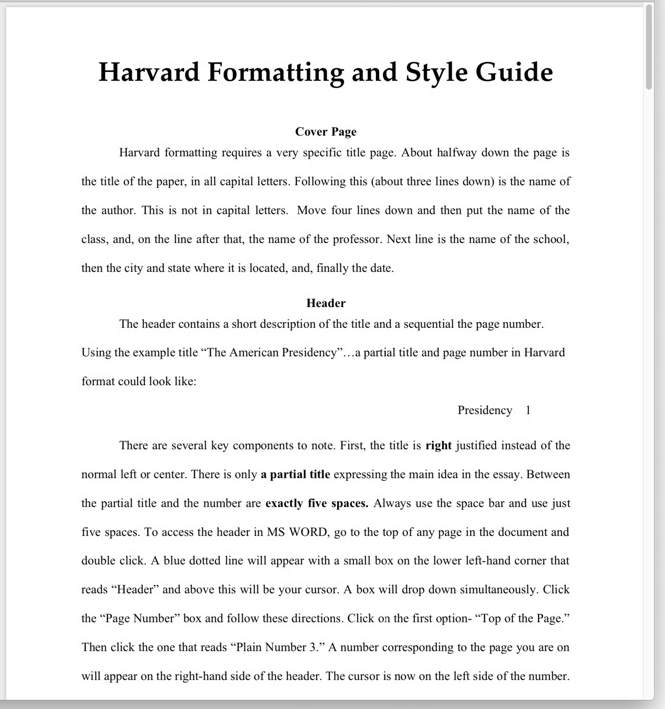


**결과 : 영상이 어둡거나 배경 때문에 글자를 인식하지 못했던 이미지들의 글자 인식이 가능해졌다.**

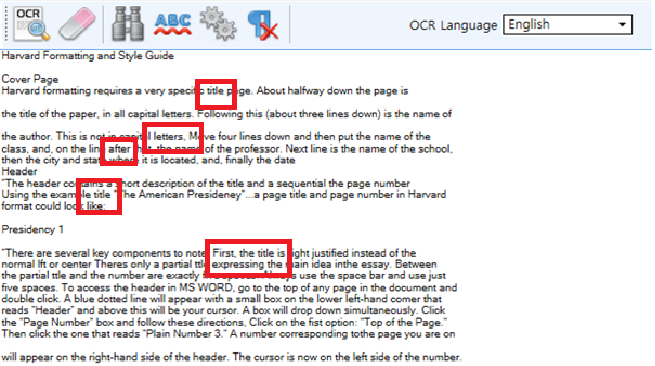
**6. 이미지 Box 파일 학습**

**Jtessboxeditor에 없던 신문 글꼴 같은 추가 글꼴들이나 추가적인 이미지파일의 글자들을 입력하고 수정시켜 학습하였다. 이 과정에서 학습 파일에 단어 사전도 등록하였다.**

**6.1 영어 : 원본 이미지**

****

**수정 후 OCR**

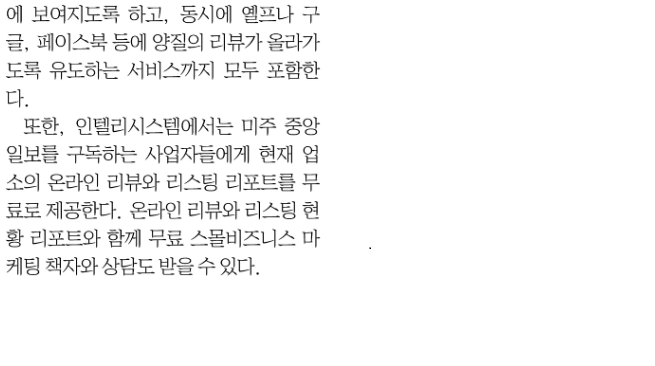
****

**결과 : 앞부분과 비교하여 오류도 거의 다 사라졌고 반복된 학습과 사전등록으로 인식률이 굉장히 올라갔다.**

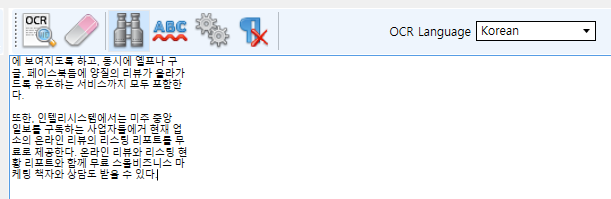
|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **89%** |
| **글꼴 추가** | **89%** |
| **Noise 제거** | **93%** |
| **이미지 학습** | **97%** |

**6.2 한글**

**원본 이미지**

****

**수정 후 OCR**

****

**결과 : 몇가지 오류 빼고는 대부분 학습이 되었고 학습과 사전등록으로 인해 인식률이 많이 올라갔다. 하지만 글자가 좀 작거나 비슷한 경우 약간의 오류가 발생하였다.**

|  |  |
| --- | --- |
| **상태** | **인식률** |
| **기본 tesseract-ocr** | **46%** |
| **글꼴 추가** | **56%** |
| **한 줄 인식** | **64%** |
| **Noise 제거** | **69%** |
| **이미지 학습** | **92%** |