

Handwritten mathematical symbols OCR

2023-1 인공지능과딥러닝 6조

Term-project

융합기계공학과

지민우 권준형

기계공학부

박준우

I. Background

□ Optical character recognition (OCR)

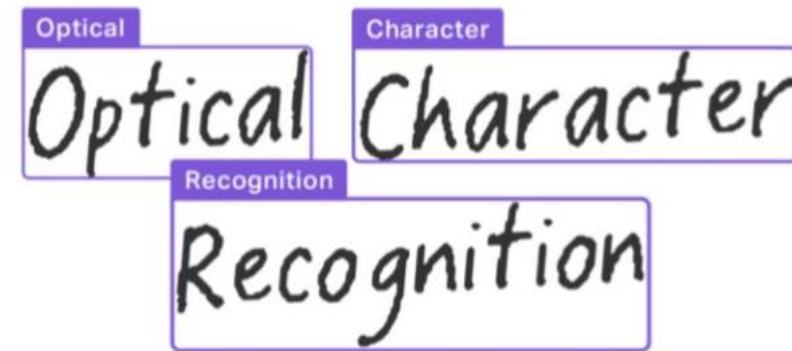
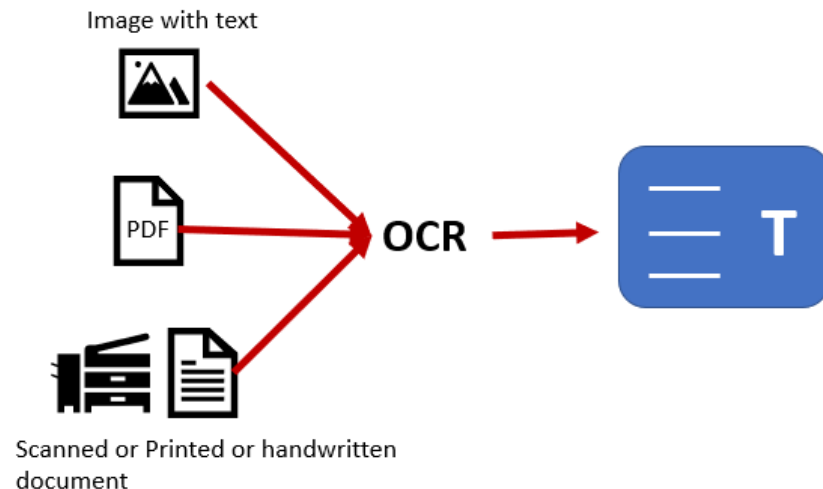
➤ Handwritten text, Printed text → **Machine-encoded format (digital form)**

➤ OCR technology

- **Automation** of text conversion
- Text conversion into **searchable information**

➤ OCR technology 활용 사례

- Papago - 이미지 번역 서비스
- 주민등록증, 사업자등록증 등 행정문서 처리 자동화
- 물류 시스템 자동화



Examples of OCR technology

I. Background

➤ OCR system

- Extraction of features
- Classification of features (based on patterns)

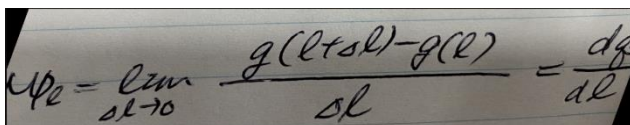
❑ Mathematical symbols OCR

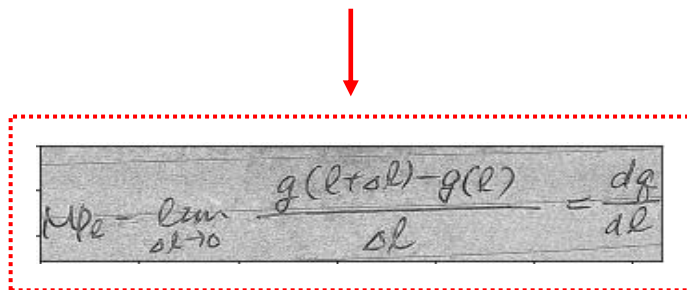
➤ 일렬로 정렬된 문자가 아닌 수식을 Recognition

- Printed 된 수식을 지원하는 상업용 OCR 프로그램
- 교육 현장 등 넓은 분야에 활용 가능

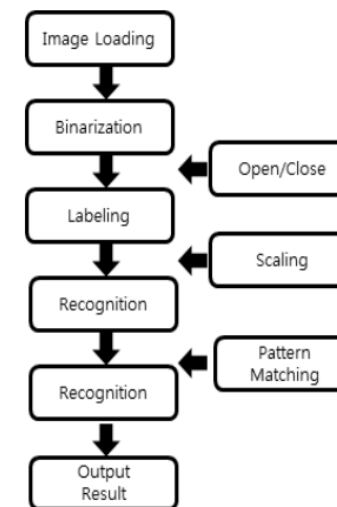
➤ Handwritten mathematical symbols OCR

- 인식 정확도가 높지 않아 상업적 활용이 부진함
- 회전, 필체 등 내-외부적 Noise가 존재하여 보다 정교한 이미지 전처리가 요구됨

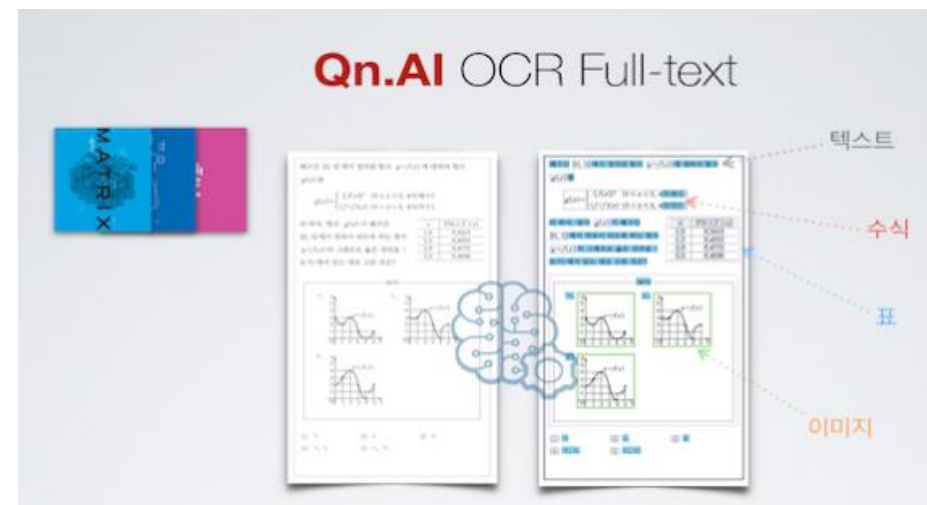




Example of preprocessing



Flow chart of mathematical symbols OCR



Commercial mathematical symbols OCR program

II. Research objectives

□ Handwritten mathematical symbol OCR

- 수식 Image를 Recognition하여 Text로 변환
- 숫자만으로 구성된 수식에 한하여 OCR을 수행한 선행 연구가 존재함
- 본 Term project에서는, **문자를 포함한 수식**을 인식하여 OCR을 수행할 수 있도록 하였음
- Overall OCR process
 - Image preprocessing (w/ OpenCV) + Recognition
 - Model training
 - Prediction

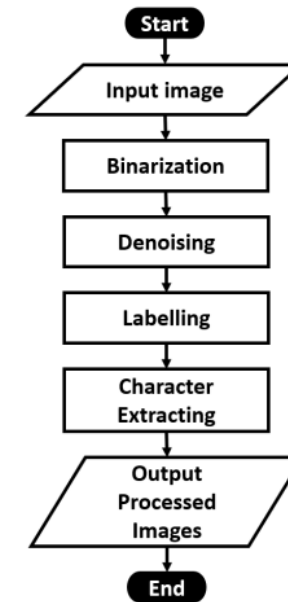


Fig. 2 Image Segmentation Flowchart

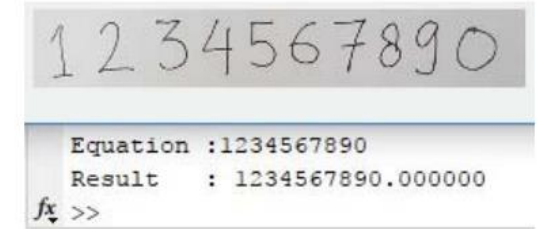


Fig. 16 Testing for Sample Image 1

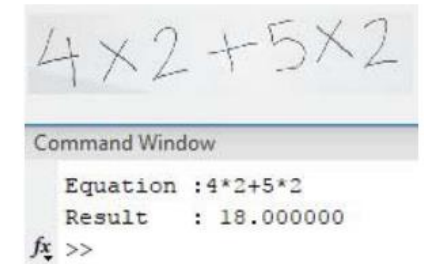
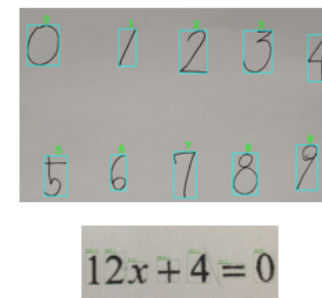


Fig. 17 Testing for Sample Image 2

Prior research related to handwritten mathematical symbol OCR



Input image data
(OpenCV)

Math symbol
OCR



$12x + 4 = 0$

Output text

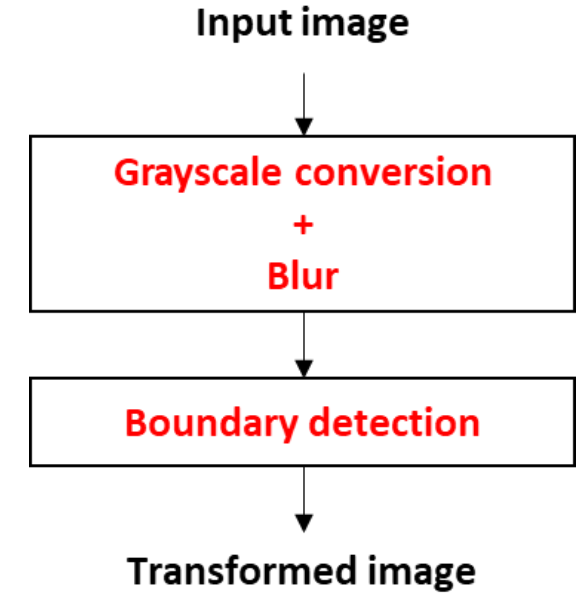
Referenece : Lee, Jhonson, Bobbi Winema Yogatama, and Hans Christian. "Optical character recognition for handwritten mathematical expressions in educational humanoid robots." *2018 IEEE 8th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2018.

Example of research objectives

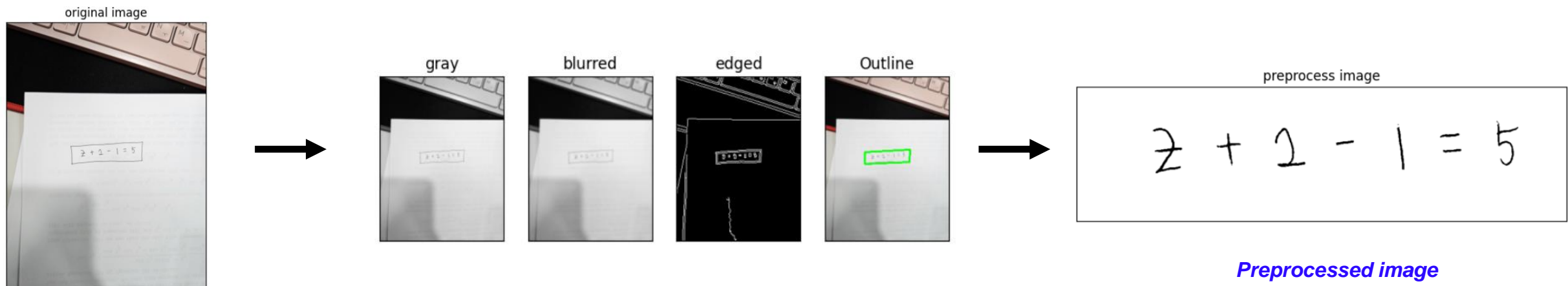
III. Methodology

□ Preprocessing

- Model prediction에 앞서, Dataset의 Noise를 줄이기 위한 Image Preprocessing이 요구됨
- 수식 영역을 정의하기 위하여, 수식 주변의 Box를 이용함
- Box를 기준으로 Truncated된 Image를 **Grayscale**로 변경하고, **Gaussian blur**를 적용 후 **Edge**를 찾기 위한 이미지 연산을 진행함
- Box를 기준으로 Denoised 된 Image를 얻을 수 있었음



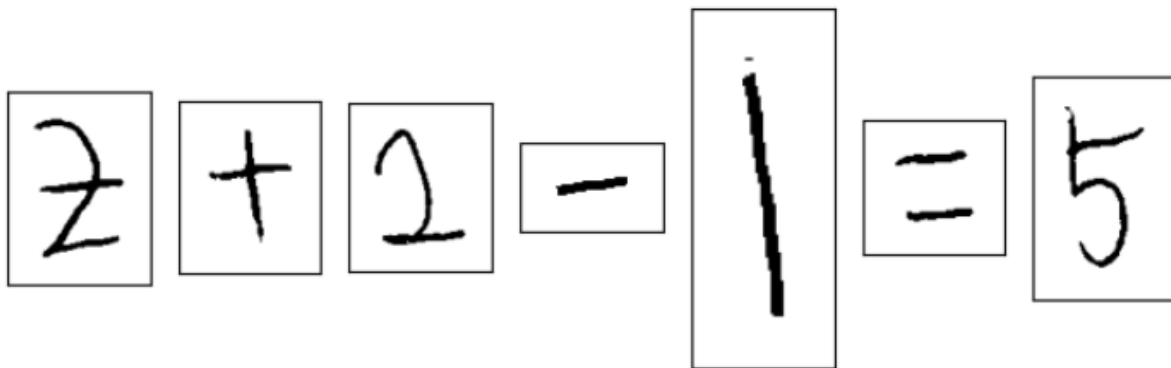
Schematic diagram of postprocessing



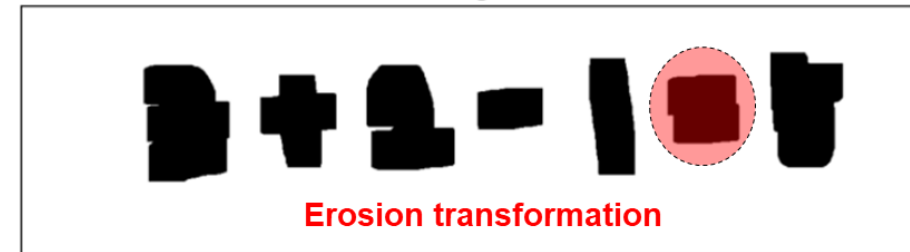
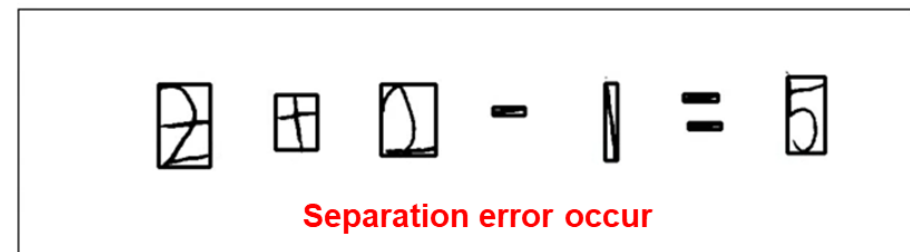
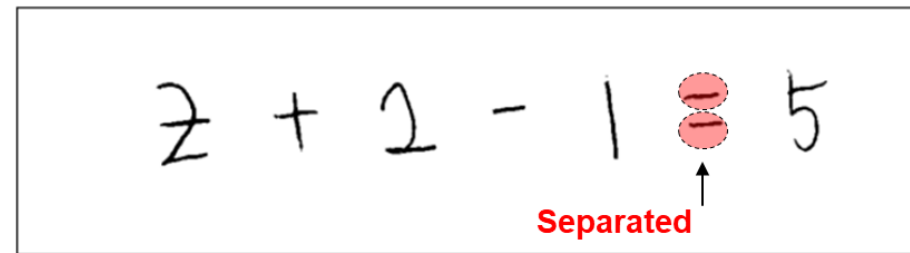
III. Methodology

□ Preprocessing - erosion

- 앞선 Preprocessing을 통하여 Image의 Noise를 줄일 수 있었으나, 등호 (=)와 같이 세로 폭이 존재하는 개체가 분리되어 인식되는 문제가 발생함
- 세로 폭을 갖는 개체가 하나의 개체로 인식될 수 있도록 폭을 조정함
- OpenCV의 Filter 중 'Dilation'과 'Erosion'이 해당 기능을 수행할 수 있으며, 개체의 색이 검정색임을 고려하여 'Erosion'을 이용하였음
- 통일된 Size로 Image를 처리하기 위하여, 'Padding'을 이용하여 Resizing을 진행하였음



Improved result of recognition

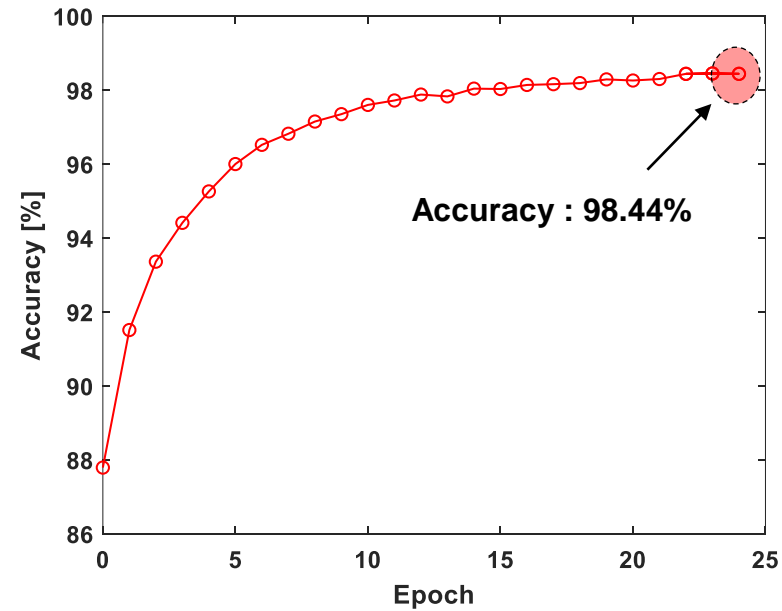
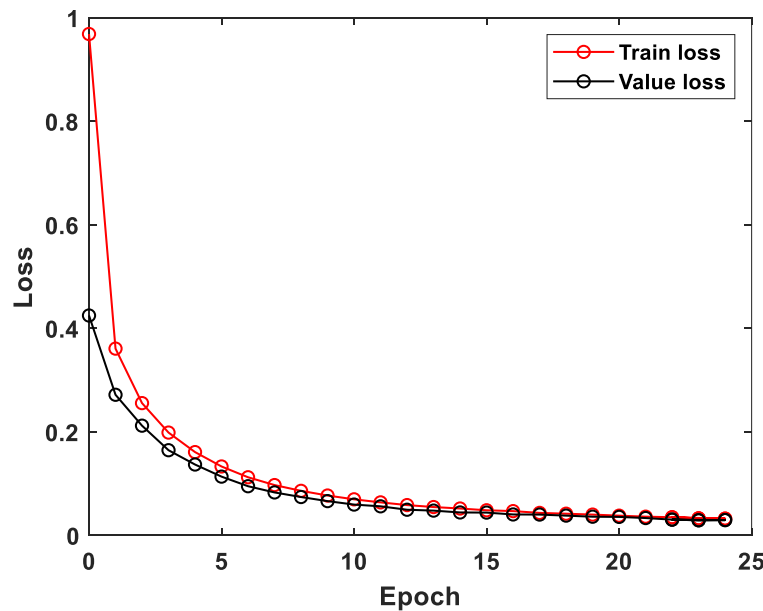


Preprocessed image with erosion

III. Methodology

□ Training

- Kaggle의 'Handwritten math symbol dataset'을 활용하여, Training 및 Testing을 진행하였음 (Train size = 338,376, Test size = 37,598, Training epoch = 25, Batch size = 256)
- Dataset에서 추출한 Mathematical symbol을 **CNN model**을 사용하여 학습
- Adam optimizer를 사용하여 Loss function의 Optimization을 수행하였으며, 그 결과는 아래와 같음

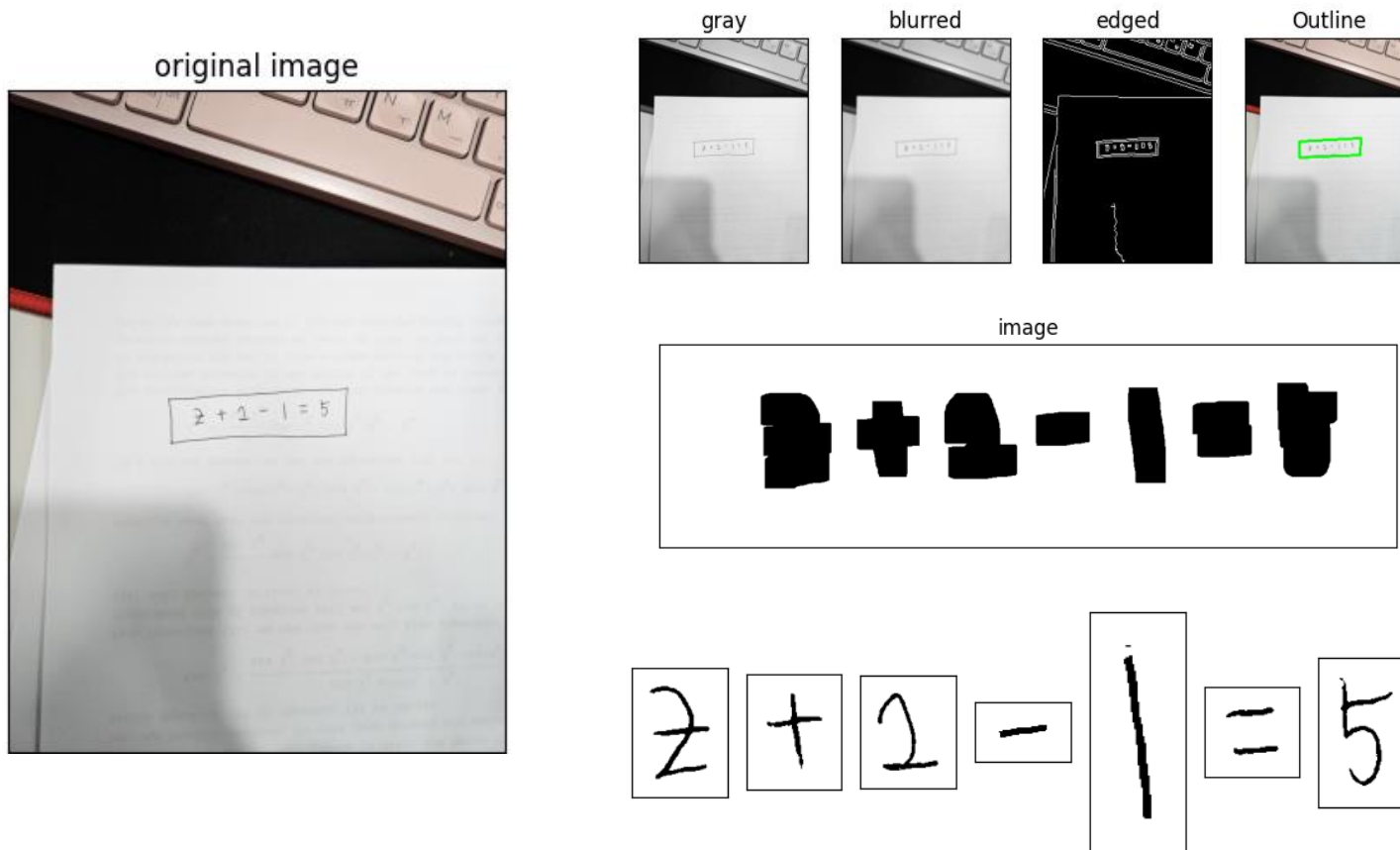


Result of optimization

IV. Experiment

Experimental setup

- 우측 표의 환경에서, 학습 소요 시간은 약 45분이었음.
- 본 모델에서 Test case에 대한 Prediction을 수행한 결과는 아래와 같음



Input image (test case)

Image recognition

Experimental setup	
OS	WSL
CPU	AMD Ryzen 5 5600X
RAM	16 GB
GPU	NVIDIA GeForce RTX 3060
TOOL	VSCode

Experimental setup

```
predict: z
predict: +
predict: 1
predict: -
predict: 1
predict: =
predict: 5
```

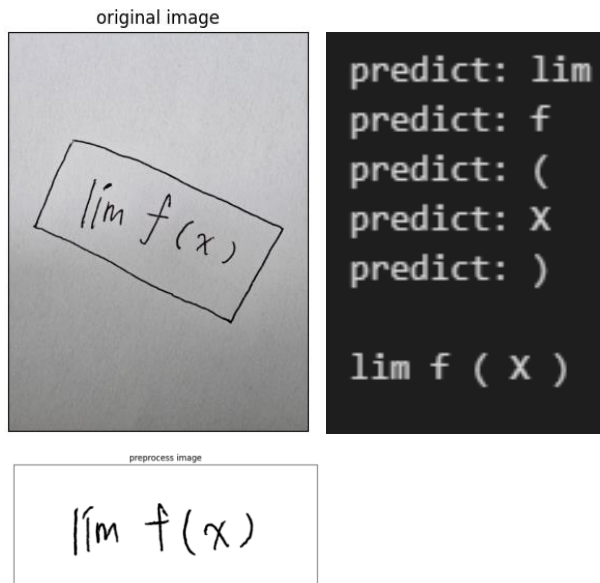
z+1-1=5

Results

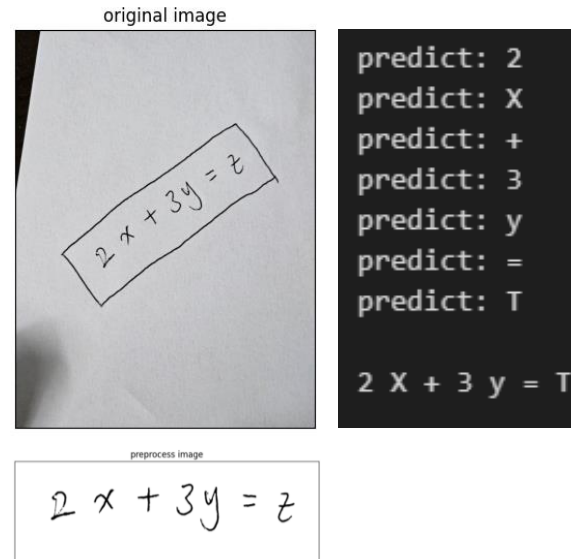
IV. Experiment

Quantitative evaluation

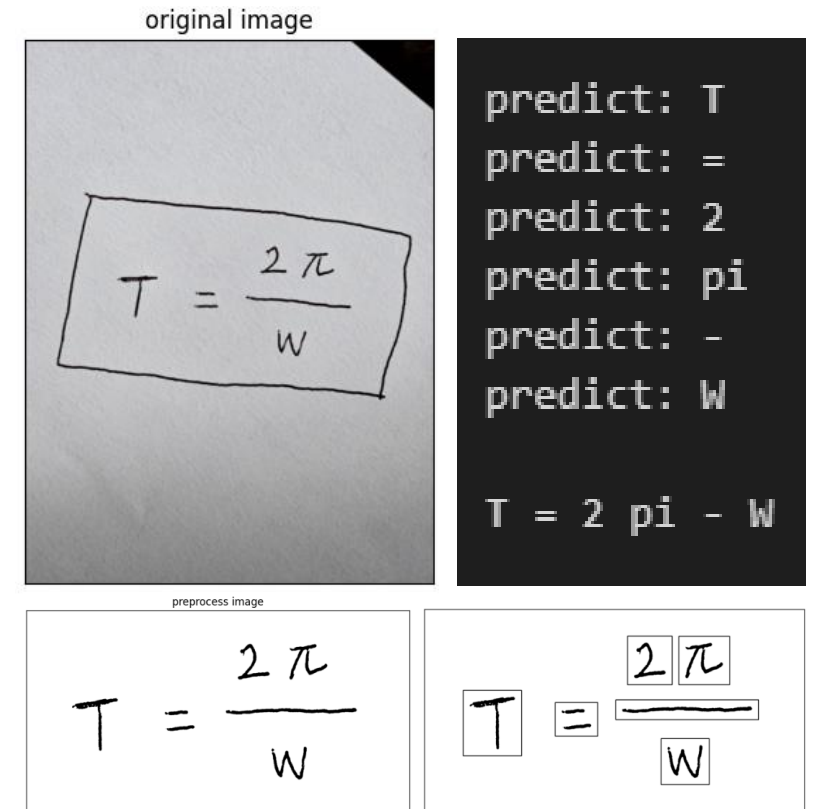
- 수기로 작성한 40개의 수식을 인식하는 Quantitative evaluation을 진행하였음
- 40개의 수식 중 인식에 성공한 수식은 30개로, 평균 인식 정확도는 **75%**이었음
- Rotation에 대해서는 비교적 강건한 인식 성능을 보였으나, Shear 및 필체 차이에 따른 인식 정확도 감소 Issue가 존재하였음. 더불어, 2차원적으로 작성된 수식(분수식)을 가로 방향으로 그대로 읽는 Issue가 존재하였음



Correct prediction



Incorrect prediction
(handwriting difference)



Incorrect prediction
(2-D equation)

V. Conclusion & Future work

□ Conclusion

- Image preprocessing을 이용하여 Noise를 최대한 제거함으로써 수식 인식 정확도를 향상시킬 수 있었음.
- 일렬로 정렬된 1-D 수식 인식 과정에서, 일부 기호가 분리된 객체로 인식되는 Issue가 발생하였으며, 'Erosion'을 통한 Preprocessing을 통하여 이를 해결할 수 있었음.
- 본 모델은 수식을 1차원적으로 처리하기 때문에 분수 등과 같이 2차원적으로 작성된 수식에 적용할 수 없다는 한계를 지님.

□ Future work

- 2차원적으로 작성된 수식의 경우, 수식 내 각 개체의 좌표를 2차원적으로 인식하는 이미지 처리 방식이 필요하다고 판단됨.
- 더불어, 필체 차이에 따른 인식 정확도 감소 문제를 해결하기 위하여 추가적인 학습 모델 개선 및 이미지 전처리 과정 보완이 필요하다고 판단됨.
- 향후 지수, 미적분 등이 포함된 다양한 수식을 인식하여 Text로 출력이 가능하도록 모델을 보완할 예정임.

Q&A