

# 저작권양도서

## (Copyright Transfer Form)

소속 : 인하대학교 정보통신공학과

성명 : 황인욱, 김규영

학번 : 12151642, 12151519

논문제목 : 컨볼루션 필터 초기화를 통한 한글인식 CNN 인식률 향상

본인은 상기 논문을 2020 학년도 2학기 정보통신프로젝트 최종 보고서 겸 결과 논문으로 제출하고자 합니다. 본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적이 없음을 확인합니다. 또한 공저자와 더불어 인하대학교 정보통신공학부에서 발간하는 논문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 정보통신공학부에 양도합니다.

2020 년 12 월 30 일

주저자 : 황인욱

공저자 : 김규영

황인욱

김규영

정보통신공학과장 귀하

## 컨볼루션 필터 초기화를 통한 한글인식 CNN 인식을 향상

### A Double-Column Format for the final report of ICE Project

황인옥, 김규영  
(InWook Hwang and GyuYoung Kim)



**요약:** 본 논문에서는 한글 OCR(Optical Character Recognition)에 사용되는 CNN(Convolution Neural Network)를 구성한다. 그 후 한글인식 CNN의 첫번째 convolution filter 일부의 weight 값을 특정한 값으로 초기화 해줌으로써 비교군 대비 성능의 향상을 얻는 기법을 제안한다. CNN의 경우 총 50만 736개의 데이터로 학습을 진행했고 자주 사용되는 한글 512개를 인식 및 구분한다. Test set으로 10240개의 데이터를 사용했고 비교군의 경우에서 95.33프로의 성능을 보인다. 초기화 해주는 Convolution Filter는 비교군으로 설정한 한글 인식 CNN에서 생겨난 오류를 분석해서 만든다. 한글 오류의 경우는 자음 인식 오류, 모음 인식 오류가 독립적으로 일어난다는 점을 주목해서 필터를 생성했다. 그 오류 분석 결과 ㄱ, ㄴ, ㄷ, ㄹ 형태 인식 강화 필터와 ㅁ의 직각 성분 강화 필터, ㅎ 인식 강화 필터를 생성했고 ㄱ 형태 필터와 ㅁ의 직각 강화 필터에서 95.33 성능 대비 각각 0.81, 1.02의 성능 향상을 이뤘다. 또한 이 필터들 적용 후 오류 분석 결과에서 의도한 부분에서 성능 향상을 이뤘음을 확인했다.

**키워드 :** 딥러닝, CNN, 한글인식 OCR, 컨볼루션 필터 초기화

**Abstract:** In this paper, we construct CNN, which is used in Korean OCR. We then propose a technique to obtain an improvement in comparison group performance by initializing the weight value of part of the first convolution filter of the Korean-language recognition CNN to a specific value. In the case of CNN, we trained the CNN with a total of 507,736 data and recognized and distinguished 512 Hangul in which is used frequently. It used 10240 data as a test set and showed the performance 95.33% in comparison groups. The convolution filter for initialization is made by analyzing errors generated by the Korean-language recognition CNN, set comparison. In the case of Hangul errors, a filter was created, noting that consonant recognition errors and vowel recognition errors occur independently. As a result of the error analysis, we created ㄱ, ㄴ, ㄷ, ㄹ shape recognition reinforcement filters, a ㅁ's perpendicular component reinforcement filter, and a ㅎ shape recognition reinforcement filter. Performance improvements of 0.81, 1.02 compared to 95.33 performance were shown in ㄱ shape recognition reinforcement filters and ㅁ's perpendicular reinforcement filters, respectively.

**Keywords:** Deep learning, CNN, Hangul Recognition OCR, Convolution Filter Initialization.

### 1. 서론

현재 많은 문서 및 도서들이 인쇄물이나 수기 형태로 존재한다. 전자책 구독 열풍으로 종이책의 수요가 줄어드는 추세이다. 따라서 자료의 효율적인 활용과 신뢰성 있는 저장을 위해 정보의 디지털 화가 요구된다. 따라서 데이터의 신뢰성 있는 변환을 위해 한글인식기의 높은 정확도가 요구된다. 하지만 한글의 경우, 실제로 사용되는 조합은 약 6300자이고 영어의 알파벳이 총 52종인것에 비해 구분해야 할 종류가 굉장히 많아 정확한 인식을 기대하기 힘들다. 또 신경망이 깊어짐에 따라 기울기 소실 문제가 발생하여 Input Image로부터 Feature를 추출하는 첫번째 Layer의 학습이 잘 이루어

지지 않는 문제가 있다.

따라서 본 연구는 첫번째 Convolution Layer의 Convolution Filter일부를 임의로 초기화하여 한글인식 CNN의 인식률을 향상시키는 Convolution Filter Initialization 기법을 제안한다. 비교군은 평균성능 95%를 보이는 CNN 모델이며 비교군 모델에 임의의 필터를 적용하고 그 외 모든 파라미터를 동일하게 해줌으로써 필터 초기화를 통해 인식률이 향상되었는지 분석할 것이다.

## 컨볼루션 필터 초기화를 통한 한글인식 CNN 인식률 향상

### A Double-Column Format for the final report of ICE Project

황인욱, 김규영  
(InWook Hwang and GyuYoung Kim)

**요약:** 본 논문에서는 한글 OCR(Optical Character Recognition)에 사용되는 CNN(Convolution Neural Network)를 구성한다. 그 후 한글인식 CNN의 첫번째 convolution filter 일부의 weight 값을 특정한 값으로 초기화 해줌으로써 비교군 대비 성능의 향상을 얻는 기법을 제안한다. CNN의 경우 총 50만 736개의 데이터로 학습을 진행했고 자주 사용되는 한글 512개를 인식 및 구분한다. Test set으로 10240개의 데이터를 사용했고 비교군의 경우에서 95.33프로의 성능을 보인다. 초기화 해주는 Convolution Filter는 비교군으로 설정한 한글 인식 CNN에서 생겨난 오류를 분석해서 만든다. 한글 오류의 경우는 자음 인식 오류, 모음 인식 오류가 독립적으로 일어난다는 점을 주목해서 필터를 생성했다. 그 오류 분석 결과 ㄱ, ㄴ, ㄷ, ㄹ 형태 인식 강화 필터와 ㅁ의 직각 성분 강화 필터, ㅎ 인식 강화 필터를 생성했고 ㄱ 형태 필터와 ㅁ의 직각 강화 필터에서 95.33 성능 대비 각각 0.81, 1.02의 성능 향상을 이뤘다. 또한 이 필터들 적용 후 오류 분석 결과에서 의도한 부분에서 성능 향상을 이뤘음을 확인했다.

**키워드 :** 딥러닝, CNN, 한글인식 OCR, 컨볼루션 필터 초기화

**Abstract:** In this paper, we construct CNN, which is used in Korean OCR. We then propose a technique to obtain an improvement in comparison group performance by initializing the weight value of part of the first convolution filter of the Korean-language recognition CNN to a specific value. In the case of CNN, we trained the CNN with a total of 507,736 data and recognized and distinguished 512 Hangul in which is used frequently. It used 10240 data as a test set and showed the performance 95.33% in comparison groups. The convolution filter for initialization is made by analyzing errors generated by the Korean-language recognition CNN, set comparison. In the case of Hangul errors, a filter was created, noting that consonant recognition errors and vowel recognition errors occur independently. As a result of the error analysis, we created ㄱ, ㄴ, ㄷ, ㄹ shape recognition reinforcement filters, a ㅁ's perpendicular component reinforcement filter, and a ㅎ shape recognition reinforcement filter. Performance improvements of 0.81, 1.02 compared to 95.33 performance were shown in ㄱ shape recognition reinforcement filters and ㅁ's perpendicular reinforcement filters, respectively.

**Keywords:** Deep learning, CNN, Hangul Recognition OCR, Convolution Filter Initialization.

#### 1. 서론

현재 많은 문서 및 도서들이 인쇄물이나 수기 형태로 존재한다. 전자책 구독 열풍으로 종이책의 수요가 줄어드는 추세이다. 따라서 자료의 효율적인 활용과 신뢰성 있는 저장에 대해 정보의 디지털 화가 요구된다. 따라서 데이터의 신뢰성 있는 변환을 위해 한글인식기의 높은 정확도가 요구된다. 하지만 한글의 경우, 실제로 사용되는 조합은 약 6300자이고 영어의 알파벳이 총 52종인것에 비해 구분해야 할 종류가 굉장히 많아 정확한 인식을 기대하기 힘들다. 또 신경망이 깊어짐에 따라 기울기 소실 문제가 발생하여 Input Image로부터 Feature를 추출하는 첫번째 Layer의 학습이 잘 이루어

지지 않는 문제가 있다.

따라서 본 연구는 첫번째 Convolution Layer의 Convolution Filter일부를 임의로 초기화하여 한글인식 CNN의 인식률을 향상시키는 Convolution Filter Initialization 기법을 제안한다. 비교군은 평균성능 95%를 보이는 CNN 모델이며 비교군 모델에 임의의 필터를 적용하고 그 외 모든 파라미터를 동일하게 해줌으로써 필터 초기화를 통해 인식률이 향상되었는지 분석할 것이다.

## II. 본론

### 1. 기존 연구

Cheng-Lin Liu의 연구에서 제안한 방향 특성 추출(Direction Feature Extraction)[3]은 그림 1과 같이 4, 8방향의 Edge Operator를 활용하여 Feature를 추출하였지만, 이는 중국어 손 글씨 인식(HCCR)을 위한 것이므로 한글에는 적합하지 않다. 실제로 한글 인식 CNN에 8-direction Edge Operator를 적용한 결과 인식률이 크게 향상되지 않았다. 따라서 본 연구에선 자주 사용되는 한글에 적합한 Convolution Filter의 초기값을 제안한다.

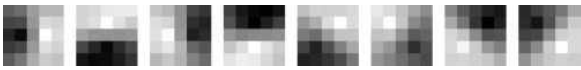


그림 1. Edge Operator

### 2. 데이터 구성

본 연구에선 자체적으로 생성한 학습 데이터와 테스트 데이터를 사용하였다. 학습 데이터셋의 경우 국립국어원 출처 가장 많이 사용되는 한글 512자[1]를 기준으로 163개의 폰트를 사용하였고 그림 2와 같이 데이터셋을 늘리기 위해 폰트별로 5번의 Elastic Distortion을 각각 적용하여 총 50만 736개의 한글 이미지를 사용하였다. 학습과 Validation의 비율은 85:15로 설정하였다.

테스트 데이터셋의 경우 학습 데이터와 마찬가지로 512자 한글을 기준으로 20개의 폰트를 사용하여 총 1만 240개의 한글 이미지를 사용하였다. 학습데이터셋과 테스트 데이터셋을 생성하는데 사용된 폰트는 독립적인 폰트를 사용하였다.



그림 2. Elastic Distortion 적용

### 3. CNN 구조

표 1. 비교군 CNN 구조 상세[4]

layers	type	# of feature maps	feature map size	window size	stride
C <sub>1</sub>	convolution	32	60x60	5x5	1
P <sub>2</sub>	max-pooling	32	30x30	2x2	2
C <sub>3</sub>	convolution	64	26x26	5x5	1
P <sub>4</sub>	max-pooling	64	13x13	2x2	2
C <sub>5</sub>	convolution	128	10x10	4x4	1
P <sub>6</sub>	max-pooling	128	5x5	2x2	2
C <sub>7</sub>	convolution	256	2x2	4x4	1
P <sub>8</sub>	max-pooling	256	1x1	2x2	1
F <sub>9</sub>	fully connected	384	1x1	N/A	N/A
F <sub>10</sub>	fully connected	520	1x1	N/A	N/A

표 2. 각 계층별 활성화 함수

Layer types	Activation Functions
Convolution Layers	ReLU
Classification Layers (hidden layer)	ReLU
Classification Layers (output layer)	Softmax (for cross entropy criterion)

앞서 설명한 비교군 CNN 구조는 표 1과 같이 구성하였다. 총 4쌍의 Convolution Layer와 Max-pooling Layer로 Feature를 추출하고, 2개의 Fully Connected Layer를 사용하고 마지막 층에서 Classification을 수행한다.

각 Layer별 활성화 함수는 표 2와 같고 손실 함수는 Cross Entropy를 사용하였다. 그 외 파라미터로 epoch는 15, batch size는 100, learning rate는 0.0001, Iteration은 63800이다.

비교군 CNN의 Convolution Filter Weight의 초기값은 표준편차가 0.28인 정규분포를 따르는 난수로 설정하였다. 첫 번째 Convolution Layer엔 총 32개의 Convolution Filter가 있는데, 그중 일부를 임의의 초기값으로 수정하고 그 외 모든 필터의 값은 비교군과 동일하게 설정한다.

### 4. 필터의 종류 선정

CNN을 활용한 한글인식기의 경우 인식오류 중 반복적으로 발생하는 대표적인 오류 케이스가 존재한다. 따라서 평균 성능 약 95%를 보이는 비교군 모델을 테스트한 후 생겨나는 오류를 분석하고 분류하였다[표 3, 4].

또한 한글의 경우 초성 중성으로 구성된 글자와 초성 중성 중성으로 구성된 글자 두 가지 케이스가 존재한다.

그리고 초성과 중성은 반드시 자음이 위치한다. 중성의 경우 반드시 모음이 위치한다. 따라서 자음의 위치에 있는 자음이 모음으로 인식되는 경우는 드물다. 또한 모음의 위치의 모음이 자음으로 인식되는 경우도 드물다. 이와 같은 이유로 그림 3을 참고하면 자음의 오류의 대부분은 다른 자음으로 인식을 한 경우이다. 마찬가지로 모음의 경우는 다른 모음으로 인식한 경우이다. 이러한 오류의 패턴을 반영해 자음 인식 오류 그리고 모음 인식 오류로 오류의 종류를 두 가지로 오류를 구분했다.

그리고 자음을 강화하는 필터 그리고 모음을 강화하는 필터를 각각을 개별적으로 구성하고 적용했다. 해당 성분만 지나치게 추출하는 필터가 되는 것을 막기 위해 각각의 5x5필터에 gaussian blur algorithm을 적용했다.

표 3. 자음 오류  
분류표

순서	유형	오류갯수
1	ㅇ	85
2	ㅎ	52
3	ㅁ	47
4	ㄴ	29
5	ㅂ	26
6	ㄹ	24
7	ㅅ	24
8	ㅍ	22
9	ㄷ	21
10	ㅈ	19
11	ㅊ	16
12	ㅌ	12
13	ㄱ	11
14	ㅋ	6
15	ㅍ	5
16	ㅈ	4
17	ㅊ	3
18	ㅌ	3
19	ㅍ	2
20	ㅈ	1
21	ㅊ	0
22	ㅌ	0
23	ㅍ	0
	총합	412

표 4. 모음 오류  
분류표

순서	유형	오류갯수
1	ㅏ	42
2	ㅓ	34
3	ㅗ	17
4	ㅡ	16
5	ㅜ	15
6	ㅛ	14
7	ㅝ	13
8	ㅞ	11
9	ㅟ	10
10	ㅠ	8
11	ㅢ	6
12	ㅣ	5
13	ㅤ	4
14	ㅥ	1
15	ㅦ	1
16	ㅧ	1
17	ㅨ	1
18	ㅩ	0
19	ㅪ	0
20	ㅫ	0
21	ㅬ	0
	총합	199

예측값(오답) : 음 / 예측 확률 : 0.94173014 / 정답 : 음  
현재까지 예러율 : 24.437575133092906 %  
파일 index number : 5822

음  
5822

그림 3. 자음 인식 오류 예시

자음 오류에 대한 분류표인 표 3를 보면 자음의 오류의 경우 ㅇ, ㅎ, ㅁ을 인식함에 있어서 생긴 오류의 경우가 가장 많다. 오류를 살펴보면 ㅇ인식을 ㅁ과 ㅎ으로 인식한 경우가 많다. 마찬가지로 ㅁ인식 오류의 경우 ㅎ과 ㅇ으로 인식한 경우가 많다. ㅎ 인식 또한 ㅁ과 ㅇ으로 인식한 경우가 많다.

따라서 그림 4처럼 육안으로도 구분하기 힘든 ㅇ과 ㅁ 인식 오류를 강화하는 필터를 그림 5와 같이 구성했다. 실제 사각형의 직각 성분의 경우 4개가 존재하기에 4가지 방향 각각을 강화하는 5x5 필터를 생성한 후 blur처리를 했다. 또한 4개의 필터의 구분을 위해 숫자로 각 필터를 구별하였다.

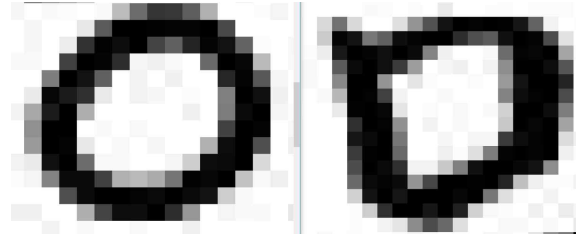


그림 4. ㅇ, ㅁ 오류

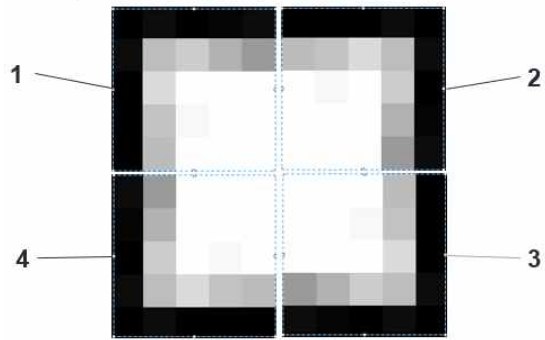


그림 5. 직각 강화 필터

ㅇ과 ㅎ의 가장 큰 특징은 ㅇ의 원형 모양과 ㅎ의 상단 부분 사이에 가로질러진 공백이 있다는 것이다. 따라서 이러한 특징에 착안하여 ㅎ인식 강화 필터를 그림 6과 같이 생성했다.



그림 6. ㅎ강화 필터

모음 오류에 대한 분류표인 표 4를 보면 단모음인 ㅏ, ㅓ, ㅗ, ㅜ, ㅡ의 오류가 가장 많다. 이 중 ㅏ, ㅓ, ㅗ, ㅜ의 오류의 경우가 지배적이다. 또한 ㅓ와 ㅗ의



오류의 경우도 1을 1로 인식하거나 -1을 1과 1로 인식하는 경우가 대다수였다. 따라서 그림 7과 같이 1, 1, 1, 1의 각각의 성분을 강화해주는 5x5 필터를 생성했다.

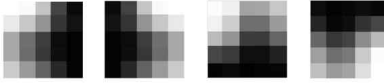


그림 7. 1, 1, 1, 1형 필터

## 5. 컨볼루션 필터 값 적용

평균 성능 95%를 보여주는 비교군 모델의 경우 활성화 함수로 ReLU 함수를 사용했다. 따라서 최적의 성능을 보이는 필터의 초기값의 범주는 He Initialization[수식 1]을 따른다.[2] 비교군의 경우 5x5의 filter size와 1개의 input 채널을 사용하므로 수식에 따라 0.28의 표준편차를 가지는 가우시안 정규분포를 따르는 난수로 필터 초기값을 가진다.

$$W \sim N(0, \sqrt{\frac{2}{n_{in}}})$$

$$n_{in}(fan\ in) = receptive\ field \times input\ channel \\ = (5 \times 5) \times 1 = 25$$

[수식 1] He initialization[2]

한글 이미지 파일에서 직접 특징을 추출하는 필터는 첫 번째 컨볼루션 레이어의 32개의 필터이다. 따라서 첫 번째 컨볼루션 필터 값 중 일부의 Weight 값을 수정함으로써 성능 향상을 이끌어 낸다. 수정해준 필터를 제외한 필터 값은 그림 9와 같이 95%의 성능을 보였던 비교군 모델과 동일하게 설정했다.

또한 임의로 만들어 적용해준 필터의 값 또한 He initialization의 데이터 경계 조건을 만족해야 한다. 그리고 실제로 학습이 끝난 필터 값들과 동떨어진 데이터 값은 성능 저하를 야기할 수 있다. 따라서 학습이 끝난 후 첫 번째 필터의 weight의 분포를 확인해보면 그림 8과 같다.

따라서 필터 Weight값은 그림 10과 같이 위의 조건을 만족하는 0.1 ~ -0.1의 범주로 Normalization을 했다.

최종적으로 실험에서 적용해준 필터의 종류는 그림 11과 같다.



그림 11. 전체 필터 종합

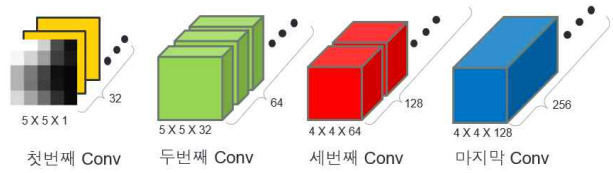


그림 9. 임의의 필터 적용



그림 10. Grey data Normalization

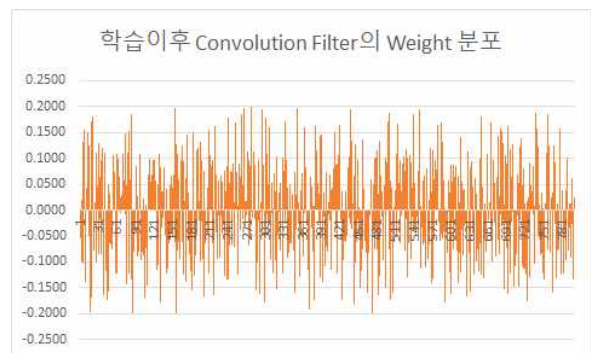


그림 8. 학습 이후 Filter Weight 분포

## 6. 실험 결과

표 5는 각 필터별로 필터 적용 모델을 10번씩 학습시킨 후 이를 평균을 낸 정확도를 종합한 것이다.

표 5. 필터 적용 결과 종합

Random	1	2	3	4	5	6	7
95.33%	94.73%	94.12%	96.14%	94.14%	93.36%	94.55%	94.84%

1번	2번	3번	4번	5번	6번
94.43%	96.35%	94.43%	94.76%	95.33%	94.25%

표 5의 결과를 살펴보면 95.33%의 성능을 보여주는 기존의 필터 값 대비 성능 향상을 보여주는 필터는 1모양 필터와 직각강화 2번 필터이다. 이외의 필터의 경

우는 오히려 성능이 저하 되었음을 확인할 수 있다. 이유는 특정한 부분에 대한 보강을 해줌으로써 필터가 다른 특징점을 찾아내는 역할에 악영향을 끼치기 때문이다. 또한 필터 초기화를 많이 적용해줄수록 성능이 저하되는 경향을 볼 수 있다. 특정한 영역의 성분 검출을 강화할수록 해당하는 성분에 과적합이 되는 경향을 보이는 것으로 예상된다.

자음 성분의 오류(표 6)를 살펴보면 직각 성분을 강화해준 직각 강화 2번 필터의 경우 의도한 대로 ㅇ과 ㅁ의 인식에서 높은 성능 향상을 보였다. 하지만 ㅈ과 ㅅ의 인식의 오류를 증가시키는 것을 볼 수 있다. 모음 성분의 오류(표 7)에서도 ㅓ 모양에서 직각성분이 강화되어 성능 향상을 확인할 수 있다. 결과적으로 의도한 부분에서 성능 향상을 이뤘음을 확인할 수 있다.

표 8, 9를 살펴보면 ㅓ 모양의 인식률이 상승했음을 확인할 수 있다. 또한 의도한 ㅓ 모양 인식 상승 이외에도 ㅇ과 ㅁ의 인식률 향상을 확인할 수 있는데 ㅓ 모양에 있는 직각 성분이 ㅁ의 특징점 추출을 강화한 결과로 예상이 된다. 이 경우에도 의도한 부분에서 성능 향상이 일어났음을 확인할 수 있다.

표 7. 직각2번적용후 랜덤 대비 오류 증감(모음)

모음유형	Random	직각 2번	오류 감소
ㅏ	42	25	17
ㅑ	34	19	15
ㅓ	15	1	14
ㅕ	8	3	5
ㅗ	6	3	3
ㅛ	4	1	3
ㅜ	1	0	1
ㅠ	1	0	1
ㅡ	10	10	0
ㅣ	1	1	0
ㅐ	0	0	0
ㅒ	1	1	0
ㅔ	0	0	0
ㅖ	0	0	0
ㅘ	0	0	0
ㅙ	5	7	-2
ㅚ	14	17	-3
ㅜ	11	14	-3
ㅝ	17	23	-6
ㅡ	16	24	-8
ㅞ	13	21	-8
오류 총계	199	170	29

표 6. 직각2번적용후 랜덤 대비 오류 증감(자음)

자음유형	Random	직각 2번	오류 감소
ㅇ	85	37	48
ㅂ	26	9	17
ㄴ	29	14	15
ㄷ	21	8	13
ㅁ	47	34	13
ㅍ	22	13	9
ㅋ	6	0	6
ㅅ	16	11	5
ㅈ	3	0	3
ㅊ	12	10	2
ㅊ	11	10	1
ㅌ	5	4	1
ㄴ	1	0	1
ㄹ	0	0	0
ㄴ	0	0	0
ㄷ	3	4	-1
ㅊ	52	55	-3
ㅈ	2	5	-3
ㄹ	24	28	-4
ㅈ	0	4	-4
ㅈ	4	9	-5
ㅈ	19	35	-16
ㅈ	24	44	-20
오류 총계	412	334	78

표 8. ㅓ 필터 적용후 랜덤 대비 오류 증감(자음)



자음유형	Random	ㅓ형	오류 감소
ㅁ	47	17	30
ㅇ	85	61	24
ㅂ	26	5	21
ㄷ	21	5	16
ㅍ	22	9	13
ㄴ	29	19	10
ㄱ	11	7	4
ㅌ	12	9	3
ㅈ	3	0	3
ㅊ	3	2	1
ㅌ	5	4	1
ㄴ	1	0	1
ㄴ	0	0	0
ㅈ	24	25	-1
ㅈ	0	1	-1
ㅈ	4	6	-2
ㄹ	24	27	-3
ㄹ	0	3	-3
ㅈ	2	7	-5
ㅋ	6	13	-7
ㅊ	52	63	-11
ㅈ	19	32	-13
ㅈ	16	41	-25
오류 총계	412	356	56

표 9. 1필터 적용후 랜덤 대비 오류 증감(모음)

모음유형	Random	1형	오류 감소
ㅏ	42	10	32
ㅑ	17	10	7
ㅓ	5	1	4
ㅕ	34	30	4
ㅗ	4	0	4
ㅛ	14	12	2
ㅜ	13	12	1
ㅠ	1	0	1
ㅡ	1	0	1
ㅗ	15	15	0
ㅛ	8	8	0
ㅜ	0	0	0
ㅠ	0	0	0
ㅡ	0	0	0
ㅏ	0	0	0
ㅑ	0	0	0
ㅓ	6	7	-1
ㅕ	1	2	-1
ㅗ	1	2	-1
ㅛ	10	14	-4
ㅜ	11	19	-8
ㅠ	16	32	-16
오류 총계	199	174	25

### III. 결론

표 10. 최종 결과

		
Random	1	2번
95.33%	96.14%	96.35%

본 연구에선 첫 번째 Convolution Layer의 Filter의 일부를 초기화 해줌으로써 한글 인식 CNN의 인식률을 향상시키는 방법을 제안한다. 다양한 형태의 필터를 조합 및 적용하여 테스트를 해 본 결과, 위 두 종류의 필터를 적용해줌으로써 유의미한 성능 향상이 일어났다.

또한 오류를 분석해 본 결과 우리가 의도한 대로 ‘ㅏ’ 인식이 강화되었고 직각이 강화됨에 따라 전반적인 인식률이 상승하였음을 알 수 있다. 또 두 필터를 적용한 모델 모두 인식률이 상승한 자모음이 비슷한 경향을 보였다. 이는 두가지 필터가 한글에서 직각 성분을 강화하는 역할을 유사하게 수행하였고 이는 직각 성분이 한글 인식의 핵심적인 요소임을 보여준다고 생각

된다.

하지만 Convolution Filter를 임의로 설정해준 부작용으로 ㅓ, ㅕ, ㅗ와 같은 다른 자모음에서의 인식률이 다소 낮아지는 것 또한 확인할 수 있었다. 향후 연구에서 이 부분을 조금 더 보완한다면 더 높은 인식률 향상을 기대할 수 있을 것으로 생각된다. 한글에 전반적으로 적용 가능한 성능향상 필터를 구성하기 위해 512개 한글 인식 CNN 뿐만이 아닌 다양한 조건의 글자 인식 CNN에 대한 필터 초기값에 대해서도 연구의 필요하다 생각된다.

### IV. 참고문헌

- [1] “현대국어 사용빈도조사 “, 국립국어원, 2020년 11월 13일 접속,  
URL:https://www.korean.go.kr/front/reportData/reportDataView.do?mn\_id=207&report\_seq=1
- [2] H.Kaiming, Z.Xiangyu, R.Shaoqing, S.Jian,“Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification“, IEEE International Conference on Computer Vision, 2015
- [3] H. Liu and X. Ding, “Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes,Proceedings.” 8thInternational Conference onDocument Analysis and Recognition,IEEE, 2005.
- [4] Kim. IJ, Xie. X, "Handwritten Hangul recognition using deep convolutional neural networks," , IJDAR, pp. 1-13, 2015.



김 규 영

2015 ~ 현재 인하대학교  
정보통신공학과 학사과정  
재학 중.  
관심분야는 임베디드 시스템, 디지털 시스템, 인공지능



황 인 옥

2015 ~ 현재 인하대학교  
정보통신공학과 학사과정  
재학 중.  
관심분야는 인공지능, 영상처리