

한글 인식을 위한 CNN 기반의 간소화된 GoogLeNet 알고리즘 연구

김연규 · 차의영*

Streamlined GoogLeNet Algorithm Based on CNN for Korean Character Recognition

Yeon-gyu Kim · Eui-young Cha*

Department of Computer Science Engineering, Pusan National University, Pusan 46241, Korea

요 약

CNN(Convolutional Neural Network)을 사용한 심화 학습이 다양한 분야에서 진행되고 있으며 관련 연구들은 이 미지 인식의 많은 분야에서 높은 성능을 보이고 있다. 본 논문에서는 한글 인식을 위해 대규모 한글 데이터베이스를 학습할 수 있는 CNN 구조의 간소화된 GoogLeNet을 사용한다. 본 논문에 사용된 데이터베이스는 대규모 한글 데이터베이스인 PHD08로 총 2,350개의 한글 문자에 대해 각 2,187개의 샘플을 가져 총 5,139,450개의 데이터로 구성되어 있다. 간소화된 GoogLeNet은 학습의 결과로 학습 종료 시점에서 PHD08에 대해 99% 이상의 Top-1 테스트 정확도를 보였으며 실험의 객관성을 높이기 위해 PHD08에 존재하지 않는 한글 폰트로 이루어진 한글 데이터를 제작하여 상용 OCR 프로그램들과 분류 성능을 비교하였다. 상용 OCR 프로그램들은 66.95%에서 83.17%의 분류 성공률을 보인 반면, 제안하는 간소화된 GoogLeNet은 평균 89.14%의 분류 성공률을 보여 상용 OCR 프로그램들보다 높은 분류 성공률을 보였다.

ABSTRACT

Various fields are being researched through Deep Learning using CNN(Convolutional Neural Network) and these researches show excellent performance in the image recognition. In this paper, we provide streamlined GoogLeNet of CNN architecture that is capable of learning a large-scale Korean character database. The experimental data used in this paper is PHD08 that is the large-scale of Korean character database. PHD08 has 2,187 samples for each character and there are 2,350 Korean characters that make total 5,139,450 sample data. As a training result, streamlined GoogLeNet showed over 99% of test accuracy at PHD08. Also, we made additional Korean character data that have fonts that are not in the PHD08 in order to ensure objectivity and we compared the performance of classification between streamlined GoogLeNet and other OCR programs. While other OCR programs showed a classification success rate of 66.95% to 83.16%, streamlined GoogLeNet showed 89.14% of the classification success rate that is higher than other OCR program's rate.

키워드 : 분류, 컨볼루션 뉴럴 네트워크, 심화 학습, 한글 인식

Key word : Classification, Convolutional Neural Network, Deep Learning, Korean Character Recognition

Received 25 April 2016, Revised 27 April 2016, Accepted 09 May 2016

* Corresponding Author Eui-Young Cha(E-mail:eycha@pusan.ac.kr, Tel:+82-51-510-2878)

Department of Computer Science Engineering, Pusan National University, Pusan 46241, Korea

Open Access <http://dx.doi.org/10.6109/jkice.2016.20.9.1657>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

모든 정보가 전산망을 통해 생성되고 공유되는 현대 사회에서 출력된 문자를 다시 디지털 데이터로 복원하는 시스템은 우리 사회에서 가장 기본적인 시스템이다. 하지만 한글 필기체 인식 문제의 경우 한글이 가질 수 있는 많은 불규칙한 곡선과 구조로 인해 명쾌히 해결되지 않고 있다.

하지만, 최근 영상 인식 분야에 CNN(Convolutional Neural Network)이 본격적으로 사용되기 시작하면서 영상 인식에 존재하는 다양한 방법의 성능이 향상되고 있다[1-3]. 문자 인식 분야에서도 CNN을 적용하여 인식 성능을 개선하는 연구가 진행되고 있으며 특히 중국어 필기체 인식 연구에서 뚜렷한 성과를 보여주고 있다. Zhang[4]은 CNN의 깊이가 다른 몇 개의 CNN 네트워크를 통해 중국어 필기체 인식의 정확도를 비교 실험하였고 Zhong[5] 등은 학습 데이터마다 방향에 따른 특징 맵을 생성한 후 기존의 학습 데이터와 함께 CNN에 학습시켜 향상된 학습 정확도를 보였다. Yang[6] 등은 한문의 도메인 정보를 이용하여 새로운 학습 데이터를 생성한 후, CNN에 적용하여 기존보다 나아진 한문 인식 정확도를 얻을 수 있었다.

본 논문에서는 함대성[7] 등이 제작한 대규모 한글 데이터베이스인 PHD08과 CNN 구조를 사용하여 한글 인식 실험을 진행하였다. 총 5,139,450개의 데이터를 가진 PHD08은 현재 존재하는 한글 데이터베이스 중 가장 큰 규모의 데이터베이스이며 글자마다 폰트, 해상도, 품질, 회전, 각도 등이 모두 다른 샘플을 가져 많은 학습 데이터를 필요로 하는 CNN 구조에 매우 유용하다. PHD08을 학습 데이터와 테스트 데이터로 분리하여 본 논문에서 사용한 간소화된 GoogLeNet에 학습시킨 후 학습 횟수에 따른 테스트 정확도를 비교한다.

논문 구조는 다음과 같다. 2장에서 한글 인식과 관련된 기존 연구들을 소개한 후 3장에서 CNN의 간단한 소개와 함께 간소화된 GoogLeNet의 자세한 구조를 보인다. 그 후 4.1장에서 간소화된 GoogLeNet이 PHD08을 학습하는 실험을 보인 후 학습 단계에 따른 테스트 정확도를 보인다. 4.2장에서는 성능 평가의 객관성을 위해 PHD08에서 사용하지 않는 폰트를 사용한 한글 데이터를 분류시켜 보는 실험을 진행하고 온라인에 존재하는 상용 OCR 프로그램들과 분류 성공률을 비교하여

본 논문에서 사용한 간소화된 GoogLeNet의 분류 성능이 더 우수함을 보인다. 그 후 4장에서 결론으로 정리한다.

II. 관련 연구

2.1. HMM(Hidden Markov Model)

한글 인식 분야는 국내에서 90년대부터 다양한 방법으로 연구가 진행되었다. 최근 CNN이 등장하기 전까지 HMM을 중심으로 한글 인식 연구가 활발하였다. 신봉기[8] 등은 한글의 자소별로 모형을 하나씩 설계하고 이들을 제자 원리에 따라 연결해 하나의 글자 네트워크 모형을 구성하여 실험하였다. 또한, 하진영[9] 등은 실험을 통해 HMM의 상태 수를 어떻게 정할 것인가에 대해 연구하였다. 이와 같은 HMM을 사용한 연구들은 높은 인식률을 보였지만 실시간으로 현재 문자가 써지고 있는 위치 정보를 가지는 온라인 한글에만 목적이 있어 이미 완성된 문자를 인식하는 오프라인 한글 인식 분야에는 적용하는 데 한계가 있다.

2.2. 기타

HMM을 사용한 한글 인식 외에 황도찬[10] 등은 원 이미지로부터 특징 지점을 찾아내고 이들을 이용하여 획을 결정하고 자모를 분리하여 각각 의사 결정 나무를 통해 인식하는 방법을 제안하였다. 하지만 제안된 특징 지점을 찾아내는 알고리즘을 무한히 많은 한글 필기체의 패턴에 적용하는 데 한계가 존재한다.

최근 김인정[11] 등은 앞서 언급한 CNN을 통한 중국어 인식 연구와 유사하게 CNN을 사용하여 필기체 한글을 인식할 수 있도록 연구를 진행하였다. 직접 CNN을 구성하여 필기체 한글을 학습시켰으며 기존의 연구보다 한글에 존재하는 수많은 패턴에 대응할 가능성을 보였다. 하지만 실험의 검증 과정이 사용된 학습 데이터베이스 내부에서만 이루어져 다른 한글 데이터가 제시되었을 때, 높은 인식률을 보인다는 보장이 없다. CNN의 구성도 단 3개씩의 컨볼루션 계층과 풀링 계층만 사용되어 비교적 단순하다. 본 논문에서는 CNN을 학습시키기 위해 사용한 학습 데이터 이외에 임의로 얻은 필기체 한글 데이터로 실험을 검증하여 전체적인 실험의 신뢰성을 향상한다. 또한, 최근 국제대회에서 수상

한 CNN 구조를 사용하여 효과적인 한글 학습이 이루어질 수 있도록 하였다.

III. 한글 인식을 위한 CNN 구조 설계

3.1. Convolutional Neural Network 의 소개

입력 데이터의 지역적 특징 정보를 추출하여 학습에 사용하는 CNN은 숫자를 인식하기 위해 LeCun[12]이 설계한 LeNet부터 본격적으로 사용되기 시작했다. 컨볼루션, 풀링, 완전 연결과 같은 계층을 반복적으로 사용하는 신경회로망의 일종인 CNN은 이미지 인식을 위한 심화학습에서 가장 많이 사용되는 구조로 자리매김했다. 최근 비선형 활성화 함수인 ReLU(Rectified Linear Unit) 계층과 Drop-out 계층이 등장하면서 오버피팅을 줄이고 성능이 더욱 향상되는 모습을 보이고 있다. CNN의 컨볼루션 계층에 데이터가 입력되면 출력 값은 다음 식 (1)과 같이 계산되어 다음 계층의 노드로 전달된다.

$$x_i^l = \sum x^{l-1} * k_i^l + b_i^l \quad (1)$$

여기서 x_i^l 는 l^{th} 계층의 i^{th} 노드가 가지는 연산 결과 값이며 이는 l^{th} 계층에 연결된 i^{th} 커널 맵과 $l-1^{th}$ 계층의 연결된 노드 값을 곱해 누적한 결과에 바이어스 b_j^l 를 더한 값으로 계산된다. CNN의 최종 계층에서 결과가 출력되면 출력 값이 실제 값과 얼마나 가까운지 나타내는 값으로 흔히 Loss 또는 Error 라는 용어를 사용하며 전체 데이터셋 D의 모든 데이터 객체 |D|에 대한 평균 Loss값은 다음 식 (2)와 같이 계산된다.

$$L(W) = \frac{1}{|D|} \sum_i f_w(X^{(i)}) + \lambda r(W) \quad (2)$$

W 는 현재 네트워크의 가중치 매개변수 맵이며 $f_w(X^{(i)})$ 는 데이터 객체 $X^{(i)}$ 의 Loss 값이다. $r(W)$ 는 가중치 상수 λ 와 함께 사용되는 제약 조건이다. 식 (2)에서 Loss 값의 계산이 완료되면 매 학습마다 Loss 값의 변화에 대해 가중치 매개변수 맵을 새로 업데이트해

야 하며 이는 식 (3)과 같이 SGD(Stochastic Gradient Descent) 방식을 사용하여 수행된다. 최종적으로 Loss 값을 최소화하는 것이 학습의 목표이다.

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \quad (3)$$

$$W_{t+1} = W_t + V_{t+1}$$

먼저, 업데이트된 가중치 매개변수 맵인 W_{t+1} 은 $t+1$ 번째 학습단계에서 새로 바뀐 업데이트 값을 뜻하는 V_{t+1} 에 의해 계산된다. V_{t+1} 을 계산하기 위해 μ 라는 모멘텀 상수가 존재하며 V_t 는 이전 학습 단계에서의 업데이트 값을 의미한다. 마지막으로 α 는 현재 학습 단계에서의 학습 율을 의미하며 $\nabla L(W_t)$ 는 이전 학습 단계에서의 가중치 맵에 대한 Loss값의 음의 변화 경사를 의미한다. 실험에 사용되는 상수들의 구체적인 값은 3장에서 제시한다.

3.2. 한글 인식을 위한 간소화된 GoogLeNet

LeCun 의 LeNet 이후 성능을 더욱 개선하기 위해 많은 CNN 구조가 제시되었다. 제시된 많은 CNN 구조 중, IMAGENET[13]이 개최하는 이미지 인식 대회 ILSVRC-2012의 이미지 인식 분야에서 우승한 Krizhevsky 등의 AlexNet[14] 과 ILSVRC-2014 대회의 같은 분야에서 우승한 Szegedy 등의 GoogLeNet[15]이 많은 주목을 받았다. 본 논문에서는 한글 인식을 위해 가장 최신의 CNN 구조 중 하나인 GoogLeNet에 기반하여 간소화된 GoogLeNet을 설계하였다. CNN 구조를 가진 GoogLeNet은 기존의 CNN 구조들보다 훨씬 더 깊은 구조를 가진다. 가중치 매개변수를 가지는 계층이 22개로 8개의 가중치 매개변수 계층을 가지는 AlexNet보다 2배 이상 깊다. GoogLeNet의 가장 큰 특징은 인셉션 모듈을 가진다는 점이다. 인셉션 모듈은 Arora[16]의 방법을 응용하여 설계되었으며 네트워크 안의 네트워크 구조로 설계되어 기존의 1차원 직렬 구조만을 가지는 CNN과는 차이가 있다. GoogLeNet과 간소화된 GoogLeNet에서 동일하게 사용되는 인셉션 모듈은 그림 1에서 확인할 수 있으며 깊이가 2 다. 깊이란 가중치 매개변수를 가지는 계층이 몇 개가 이어져 있는지를 나타낸다. 인셉션 모듈은 공간의 지역적 특징을 가장 효과적으로 표현하기

위해 도입된 방식이다.

1×1, 3×3, 5×5 크기의 커널로 공간의 지역적 특징을 세분화하여 분리한 후 계산하여 인셉션 모듈의 마지막 계층에서 모두 결합시킨다. 단, 인셉션 모듈 내부의 3×3 과 5×5 컨볼루션 계층의 계산 비용을 줄이기 위해 1×1 컨볼루션 계층이 연결되어 사용된다. 1×1 컨볼루션 계층에는 ReLU 활성화 함수가 사용되어 3×3, 5×5 컨볼루션 계층 계산의 복잡함을 감소시킬 수 있다. 기존의 GoogLeNet이 9개의 인셉션 모듈을 사용하는 것과 달리 간소화된 GoogLeNet은 3개의 인셉션 모듈만을 사용하였다.

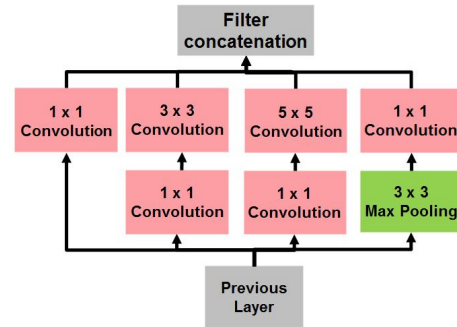


Fig. 1 Inception module for GoogLeNet

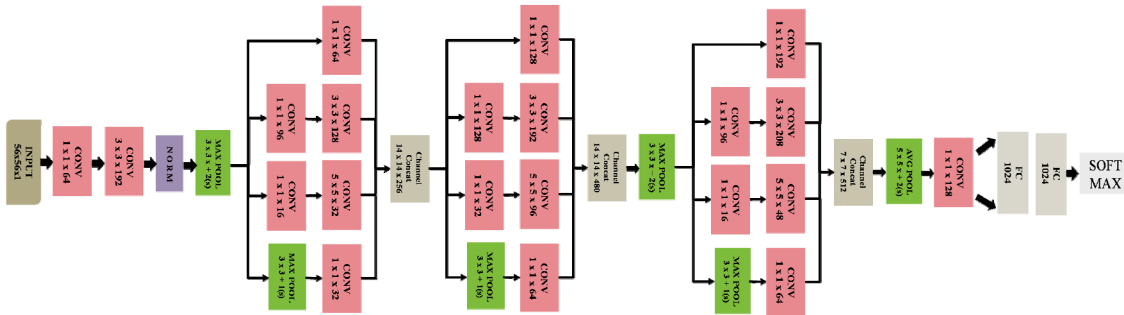


Fig. 2 The architecture of streamlined GoogLeNet

Table. 1 Streamlined GoogLeNet incarnation of the Inception architecture

Type	Patch size / Stride	Output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	Pool Proj.
Input	56 × 56 × 1		0						
Convolution	3 × 3 / 1	28 × 28 × 192	2		64	192			
Max Pooling	3 × 3 / 2	14 × 14 × 192	0						
Inception(a)		14 × 14 × 256	2	64	96	128	16	32	32
Inception(b)		14 × 14 × 480	2	128	128	192	32	96	64
Max-Pool	3 × 3 / 2	7 × 7 × 480	0						
Inception(c)		7 × 7 × 512	2	192	96	208	16	48	64
Average Pooling	7 × 7 / 1	1 × 1 × 512	0						
Convolution	1 × 1 / 1	1 × 1 × 128	1						
Fully-Connected		1 × 1 × 1024	1						
Drop-out (70%)		1 × 1 × 1024	0						
Fully-Connected		1 × 1 × 2350	1						
Softmax		1 × 1 × 2350	0						

이는 기존의 GoogLeNet은 256×256×3 크기의 자연 이미지를 분류하는 데 목적이 있지만 간소화된 GoogLeNet은 56×56×1 크기의 한글 이미지 데이터를 분류하는 데 목적이 있기 때문이다. 간소화된 GoogLeNet의 구조는 그림 2와 같으며 본 논문에서 사용하는 입력 데이터의 크기에 따라 수정된 각 계층 별 패치 크기와 출력 크기를 표 1에서 확인할 수 있다.

IV. 실험 결과

본 장에서는 두 가지의 실험에 관해 설명하고 결과를 분석한다. 먼저 4.1장에서는 PHD08을 간소화된 GoogLeNet에 학습시킨 후 학습 단계에 따른 테스트 정확도를 비교한다. 4.2장에서는 학습을 끝낸 간소화된 GoogLeNet에 PHD08에 존재하지 않는 폰트를 가진 한글 데이터를 입력하여 분류 성능을 확인하고 실험의 객관성을 평가하기 위해 상용 OCR 프로그램인 ABBY FineReader12[17], ABC-OCR[18], Office Lens[19]와 분류 결과를 비교한다.

4.1. 한글 데이터베이스 PHD08의 학습 실험

4.1.1. 실험 데이터

학습에 사용된 한글 데이터베이스인 PHD08은 다양한 조건으로 생성된 한글 문자를 인쇄한 후 스캔한 이미지를 바이너리 형태로 저장한 데이터베이스이다. 표 2와 같이 KS 완성형 문자인 2,350개의 한글 문자에 대해 문자별로 2,187개의 샘플을 가지고 있으며 샘플들은 각각 서로 다른 폰트, 글자 크기, 스캔 해상도, 노이즈 레벨, 회전, 각도 등을 가지고 있다.

Table. 2 The composition of PHD08

PHD08	Characters	Samples	Type
	2,350	2,187	Binary
Fonts	Size(pt)	Rotation	Noise
9-fonts	12, 13, 14	-3°, 0°, 3°	3-level

실험은 PHD08의 바이너리 데이터를 이진 이미지로 바꾸어 CNN의 입력 데이터로 사용한다. 단, CNN의 입력 데이터는 모두 같은 크기의 데이터를 사용해야 한다는 제약사항이 있으므로 기존 다양한 크기를 가진

PHD08 데이터들을 일괄적으로 56×56 크기로 변경한 후 사용한다. 크기를 변형하는 방법으로는 선형 보간법 [20]이 사용되었다. 최종적으로 사용되는 한글 입력 데이터의 예시는 그림 3과 같다.

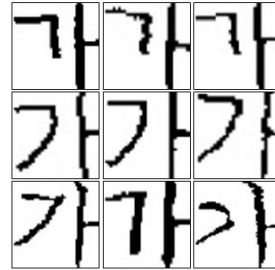


Fig. 3 Example of Input data transformed from PHD08

본 논문에서는 학습 데이터의 수가 많을 때와 작을 때 학습 단계에 따른 테스트 정확도의 차이를 확인하기 위해 표 3 과 같이 5개의 실험 데이터 set을 구성하였다. PHD08이 가지는 총 5,139,450개의 데이터를 학습데이터 수와 테스트 데이터의 비율로 1:0.5 ~ 1:8까지 총 5종류의 Set로 구성하였다. Set마다 학습을 진행하여 간소화된 GoogLeNet에서 보이는 학습 단계에 따른 테스트 정확도를 확인한다.

4.1.2. 실험 환경

학습에서 가장 중요한 역할을 하는 가중치 매개변수를 수정하는 방법은 3장에서 제시된 바와 같이 SGD방식을 사용하며 매개변수 값으로 초기 학습률은 0.01, 모멘텀 상수로 0.9를 사용한다. 또한, 학습률은 매 학습 단계 10,000번 마다 0.96 배씩 작아진다.

Table. 3 5 sets of experimental data

Set	Ratio	Training		Testing	
		Count	Percentage	Count	Percentage
Set-1	1:0.5	1,458 × 2,350 = 3,426,300	729 × 2,350 = 1,713,150		
Set-2	1:1	1,093 × 2,350 = 2,568,550	1,094 × 2,350 = 2,570,900		
Set-3	1:2	729 × 2,350 = 1,713,150	1,458 × 2,350 = 3,426,300		
Set-4	1:4	437 × 2,350 = 1,026,950	1,750 × 2,350 = 4,112,500		
Set-5	1:8	243 × 2,350 = 571,050	1,944 × 2,350 = 4,568,400		

학습 단계마다 입력 데이터로 사용되는 이미지의 개수를 뜻하는 배치 사이즈는 학습 단계에서 56, 테스트 단계에서 50이 모든 실험에서 동일하게 사용되었다. 실험에는 빠른 연산을 위해 그래픽카드 GTX-970 (CUDA v7.0)을 통한 병렬 연산이 사용되었으며 공개 프레임워크인 CAFFE[21]를 통해 진행되었다.

4.1.3. 실험 분석

표 3의 입력 데이터 Set1 ~ 5를 간소화된 GoogLeNet에 학습시키면서 나타난 학습 단계에 따른 테스트 정확도를 그림 4에 보였다. 학습 데이터의 비율이 상대적으로 5번째 실험과 같이 적더라도 충분한 학습 횟수가 진행되면 임의의 시점에서 테스트 정확도가 상승하였다. 그림 4에서 확인할 수 있는 것처럼 간소화된 GoogLeNet은 최소 30,000번의 학습 단계가 지난 후에 테스트 정확도가 상승하였다. 이는 기존 CNN 구조들과는 다르게 지역 특징을 인셉션 모듈에서 더 조밀하게 분리한 후 다시 결합하는 GoogLeNet의 특성과 관련이 있다. 인셉션 모듈이 가지는 격층 구조를 학습시키기 위해서는 많은 학습 단계가 소모되고 제시된 인셉션 모듈 내부의 매개 변수가 학습이 완료되면 테스트 정확도는 크게 상승하게 된다. 표 4는 학습이 완료된 시점에서의 테스트 정확도를 Top-1, 2, 5로 나타낸 것이다. Top-1은 CNN이 영상을 분류할 때, 주어진 전체 클래스 중 가장 높은 확률로 분류된 것이 실제와 일치한 확률을 백분율로 나타낸 것이며, Top-5는 CNN이 출력한 가장 높은 5개의 확률에 대해 실제 찾고자 하는 값이 5개 중에 있을 확률을 백분율로 나타낸 것이다.

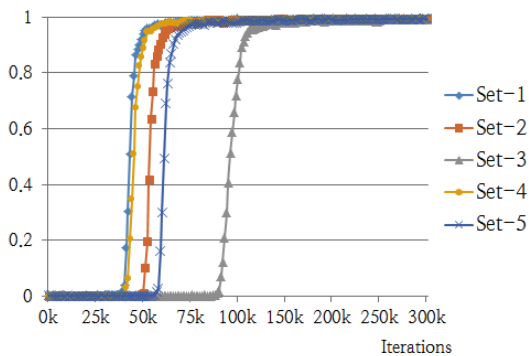


Fig. 4 Test Accuracy curve graph of streamlined GoogLeNet depends on training iteration

Table. 4 Test Accuracy of streamlined GoogLeNet about Top-1, Top-2 and Top-5

Network	Top-1 (%)	Top-2 (%)	Top-5 (%)	
Streamlined GoogLeNet	Set-1	99.41	99.73	99.89
	Set-2	99.56	99.86	99.93
	Set-3	99.14	99.74	99.90
	Set-4	99.66	99.88	99.94
	Set-5	99.40	99.83	99.90
AVG.	99.43	99.80	99.91	

또한, 본 실험의 결과에서 학습 데이터의 수와 테스트 데이터의 숫자와 무관하게 학습 데이터를 구성하는 클래스마다 최소한의 데이터만 보장이 되면 학습 결과에 큰 영향이 없는 것을 알 수 있다. 단, 모든 실험에 대해 학습 단계의 소요시간을 측정한 결과, 한 번의 학습 단계가 수행되는데 최소 0.45초, 최대 0.51초가 소요되었다.

4.2. 기존 프로그램들과의 분류 성능 비교 실험

PHD08을 통해 간소화된 GoogLeNet의 학습을 완료한 후 PHD08에 존재하지 않는 폰트를 가진 한글 데이터를 제작하여 간소화된 GoogLeNet에 입력해 보고 분류 성공률을 확인한다. 또한, 본 논문에서는 ABBYY FineReader 12, ABC-OCR, Office Lens 프로그램과 분류 성공률을 비교하여 제시한다.

4.2.1. 실험 데이터

대한민국의 국가 1~4절과 후렴구의 한글 문자를 이미지로 추출하여 실험데이터로 사용한다. 애국가 1~4절은 각 28자를 포함하며 후렴은 24자를 포함한다. 총 136자(28×4+24)가 존재하지만 중복되는 문자를 제외하면 82자의 서로 다른 문자가 존재한다. 서로 다른 82자를 한 세트로 하여 10개의 서로 다른 폰트를 적용하여 총 820개의 분류를 위한 한글 데이터를 생성한다. 본 실험에서 추가 한글 데이터를 생성하는 데 사용되는 폰트는 표 5의 New-1 ~ New-10과 같으며 PHD08에서 사용하는 폰트를 제외한 임의의 폰트가 적용되었다. 단, 본 실험에서는 그림 5와 같은 과정을 거친 후 분류를 위한 입력 데이터로 사용한다. 각 글자의 영역을 주어진 이미지에서 구분시킨 후 분리된 문자 이미지들을 모두 가로, 세로 56×56의 크기로 변형시켜서 이진 이미지로

변형한다. 마지막으로, 본 실험에서는 학습된 가중치 파라미터 맵으로 4.1장의 실험 중 Set-1을 학습한 간소화된 GoogLeNet의 가중치 파라미터 맵을 사용하였다.

Table. 5 Font name and shape examples of PHD08 and newly created data

Set	Font Name	Shape examples of font
PHD 08	Ba-da	동해물과 백두산이 마르고 닳도록
	Dot-um	동해물과 백두산이 마르고 닳도록
	Go-thic	동해물과 백두산이 마르고 닳도록
	Han-yang-hae-seo	동해물과 백두산이 마르고 닳도록
	Head-line	동해물과 백두산이 마르고 닳도록
	Myung-jo	동해물과 백두산이 마르고 닳도록
	Na-mu	동해물과 백두산이 마르고 닳도록
	Saem-mul	동해물과 백두산이 마르고 닳도록
	Yeop-seo-Han-gul	동해물과 백두산이 마르고 닳도록
	New-1	Gul-im
New-2	Gung-seo	동해물과 백두산이 마르고 닳도록
New-3	Ba-tang	동해물과 백두산이 마르고 닳도록
New-4	HY-Mok-pan-L	동해물과 백두산이 마르고 닳도록
New-5	OI	동해물과 백두산이 마르고 닳도록
New-6	Yang-jae-in-chang-jae-M	동해물과 백두산이 마르고 닳도록
New-7	MD-Gae-sung-cha	동해물과 백두산이 마르고 닳도록
New-8	Human-pyeon-jichae	동해물과 백두산이 마르고 닳도록
New-9	Ga-neun-an-sang-soo-cha	동해물과 백두산이 마르고 닳도록
New-10	HY-crystal-M	동해물과 백두산이 마르고 닳도록

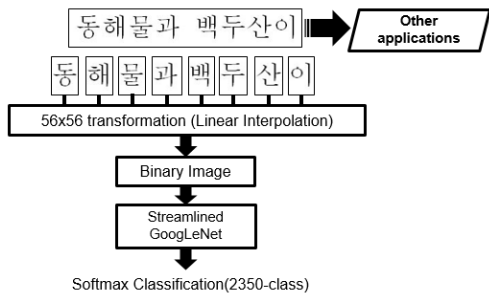


Fig. 5 Transformation of the input data using for classification with streamlined GoogLeNet

4.2.2. 실험 분석

본 논문에서 사용한 간소화된 GoogLeNet과 다른 상용 OCR 프로그램들의 한글 데이터 분류 성공률을 표 6에서 확인할 수 있다.

실험에 사용된 한글 데이터 셋, New-1부터 New-10까지 총 820개의 구분된 한글 데이터에 대해 3개의 어플리케이션인 ABBYY, ABC-OCR, Office Lens에서 각각 72.07%, 83.17%, 66.95%의 평균 분류 성공률을 보였다. n/a는 프로그램에서 입력된 데이터의 어떤 글자도 인식하지 못했거나 오류가 난 경우를 의미한다. 이에 반해 간소화된 GoogLeNet에서는 평균 89.14%의 분류 성공률을 보였다. 실험에 사용된 간소화된 GoogLeNet 및 상용 OCR 프로그램들은 사람의 필기체와 유사한 형태의 New-5, New-8, New-9에서 공통적으로 낮은 인식 성공률을 보였다. 이는 학습에 사용된 데이터베이스인 PHD08만으로 다양한 폰트가 가지고 있는 모든 패턴에 대응하는 것에는 한계가 있다는 점을 의미한다.

Table. 6 Classification performance comparison between streamlined GoogLeNet and other applications

Set	ABBYY	ABC OCR	Office Lens	Streamlined GoogLeNet
New-1	80/82	82/82	78/82	80/82
New-2	80/82	60/82	76/82	82/82
New-3	82/82	81/82	82/82	79/82
New-4	74/82	77/82	69/82	73/82
New-5	42/82	61/82	n/a	64/82
New-6	76/82	77/82	76/82	82/82
New-7	50/82	71/82	60/82	77/82
New-8	30/82	52/82	33/82	64/82
New-9	n/a	40/82	n/a	58/82
New-10	77/82	81/82	75/82	72/82
AVG.	72.07 %	83.17 %	66.95 %	89.14 %

V. 결론

본 논문에서는 한글 인식의 성능을 개선하기 위해 CNN 구조를 가진 간소화된 GoogLeNet을 설계하여 PHD08에 대해 학습을 수행하였다. PHD08을 학습한 결과 간소화된 GoogLeNet 에서 99% 이상의 높은 테스트 정확도를 보여주었다. 또한, 실험의 객관적인 평

가를 위해 PHD08에 존재하지 않는 폰트를 가진 한글 데이터를 생성하여 상용 OCR 프로그램과 성능을 비교하였다. 기존 상용 OCR 프로그램들보다 간소화된 GoogLeNet에서 인식률이 더 높은 것을 확인하여 다양한 폰트를 가진 한글 데이터에 대한 인식 성능을 보장하였다. CNN을 사용하여 문자를 인식한 기존 논문이나 연구들은 CNN을 학습시키기 위해 사용한 데이터베이스 내부에서만 인식 정확도를 비교하고 있다. 하지만, 한글과 같은 문자의 필기체에는 필기자에 따라 무한히 많은 패턴이 나타날 수 있다. 따라서, 학습에 사용한 데이터베이스 내부에서만 테스트를 진행하는 것은 실생활 응용에 한계가 존재한다. 본 논문에서는 CNN의 학습에 사용하지 않은 추가적인 데이터로 테스트를 진행하여 성능을 확인하는 과정으로 실생활에 응용할 가능성을 확인하였다.

추후 해결해야 할 과제로는 어떤 시스템에서도 필기체와 유사한 폰트를 가진 한글 데이터에서는 높은 인식 성공률을 보이지 못했다는 점이다. 따라서 한글 필기체에 관한 데이터베이스를 확보한 후 추가하여 학습을 진행하거나 필기체 인식을 위한 새로운 기술이 연구되어야 할 것이다.

ACKNOWLEDGMENTS

This work was supported by a 2-Year Research Grant of Pusan National University and BK21PLUS, Creative Human Resource Development Program for IT Convergence.

REFERENCES

[1] D. Cireşan, U. Meier, J. Masci and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333-338, Aug. 2012.

[2] N. Kalchbrenner, E. Grefenstette and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," arXiv preprint arXiv:1404.2188, 2014.

[3] P. Callet, C. Viard-Gaudin and D. Barba, "A Convolutional Neural Network Approach for Objective Video Quality

Assessment," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1316-1327, Sep. 2006.

[4] Y. Zhang, "Deep Convolutional Network for Handwritten Chinese Character Recognition," *University of Stanford, CS231N course project*.

[5] Z. Zhong, L. Jin and Z. Xie, "High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature map," in *Document Analysis and Recognition(ICDAR), 13th International Conference on IEEE*, pp. 846-850, May 2015.

[6] W. Yang, L. Jin, Z. Xie and Z. Feng, "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge," in *Document Analysis and Recognition (ICDAR), 13th International Conference on. IEEE*, pp. 551-555, May 2015.

[7] D. Ham, D. Lee, I. Jung and I. Oh, "Construction of Printed Hangul Character Database PHD08," *Journal of the Korea contents association*, vol. 8, no. 11, pp.33-40, Nov. 2008.

[8] B. Sin and J. Kim, "On-line Handwritten Character Recognition with Hidden Markov Models," in *proceeding of the Korea Information Science Society*, pp. 533-542, Oct. 1992.

[9] J. Ha and B. Shin, "Optimization of Number of States in HMM for On-line Hangul Recognition," in *Proceeding of the Korea Information Science Society*, vol. 25, no. 2, pp. 372-374, Oct. 1998.

[10] T. Hwang and S. Kim, "Hangul Recognition Using The Path Following Algorithm," *IE interfaces*, vol. 3, no. 2, pp. 53-62, Oct. 1990.

[11] I. Kim and X. Xiaohui, "Handwritten Hangul recognition using deep convolutional neural network," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 18, no. 1, pp. 1-13, Mar. 2015.

[12] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in *Proceeding of the IEEE 86.11*, pp. 2278-2324, 1998.

[13] ImageNet Large Scale Visual Recognition Challenge [Internet]. Available: <http://image-net.org/>.

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Network," in *Proceeding of the Neural Information Processing Systems 25(NIPS)*, pp. 1097-1105, 2012.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *Proceeding of the*

- Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.
- [16] S. Arora, A. Bhaskara, R. Ge and T. Ma, "Provable Bounds for Learning Some Deep Representations," CoRR (Computing Research Repository), abs/1310.6343, 2013.
- [17] Recognition Technologies and Innovation Ace (RETIA) ABBYYFineReader12 [Internet]. Available: <http://www.retia.co.kr/cnt/products/products.html?category=1&uid=24&name=finereader-12&tab=1>.
- [18] ABC-OCR [Internet]. Available: <https://itunes.apple.com/us/app/scanner-ocr-optical-character/id777913435?mt=8>.
- [19] Office Lens [Internet]. Available: <https://itunes.apple.com/en/app/office-lens/id975925059?mt=8>.
- [20] Linear Interpolation [Internet]. Available: https://en.wikipedia.org/wiki/Linear_interpolation.
- [21] CAFFE(Convolutional Architecture for Fast Feature Embedding) [Internet]. Available: <http://caffe.berkeleyvision.org/>.



김연규(Yeon-Gyu Kim)

2015년 2월 부산대학교 정보컴퓨터공학부 학사
2015년 3월 ~ 현재 부산대학교 컴퓨터공학과 석사과정
※관심분야 : 기계 학습, 심화 학습, 컴퓨터비전, 인공 지능



차의영(Eui-Young Cha)

1979년 경북대학교 전자공학과 학사
1982년 서울대학교 전자계산학과 석사
1998년 서울대학교 컴퓨터공학과 박사
1981년 ~ 1985년 한국전자기술연구소 연구원
1995년 ~ 1996년 University of London 방문교수
1985년 ~ 현재 부산대학교 컴퓨터공학과 교수
※관심분야 : 컴퓨터비전, 신경망, 웨이블릿