딥러닝을 이용한 대규모 한글 폰트 인식

양진혁O, 곽효빈, 김인중

한동대학교, 전산전자공학부
yjh2067@gamil.com, gyqls1494@gmail.com, ijkim@handong.edu

Large-Scale Hangul Font Recognition Using Deep Learning

Jin-Hyeok YangO, Hyo-Bin Kwak, In-Jung Kim
School of Computer Science and Electrical Engineering, Handong Global University

요 약

본 연구에서는 딥러닝을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 폰트는 디자인 분야에 있어서 필수적인 요소이며 문화적으로도 중요하다. 한글은 영어권 언어에 비해 훨씬 많은 문자를 포함하고 있기 때문에 한글 폰트 인식은 영어권 폰트 인식보다 어렵다. 본 연구에서는 최근 다양한 영상 인식 분야에서 좋은 성능을 보이고 있는 CNN을 이용해 한글 폰트 인식을 수행하였다. 과거에 이루어진 대부분의 폰트 인식 연구에서는 불과 수 십 종의 폰트 만을 대상으로 하였다. 최근에 이르러서야 2000종 이상의 대용량 폰트 인식에 대한 연구결과가 발표되었으나, 이들은 주로 문자의 수가 적은 영어권 문자들을 대상으로 하고 있다. 본 연구에서는 CNN을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 많은 수의 폰트를 인식하기 위해 두 가지 구조의 CNN을 이용해 폰트인식기를 구성하고, 실험을 통해 이들을 비교 평가하였다. 특히, 본 연구에서는 3300종의 한글 폰트를 효과적으로 인식하면서도 학습 시간과 파라미터의 수를 줄이고 구조를 단순화하는 방향으로 모델을 개선하였다. 제안하는 모델은 3300종의 한글 폰트에 대하여 상위 1위 인식률 94.55%, 상위 5위 인식률 99.91%의 성능을 보였다.

주제어: 한글폰트인식, 딥러닝, CNN, ResNet

1. 서론

폰트는 디자인 분야에 있어서 필수적인 요소이며 문화적으로도 중요하다. 한글 폰트 인식은 우리의 문자인 한글의 아름다움과 중요성을 보존하고 홍보하기 위해 유용한 기술이다. 폰트를 구분하기 위해서는 문자 영상에 존재하는 지역적인 세부 형태를 효과적으로 구분해야 한다. 폰트 인식 연구는 해외에서 영어권 언어나 중국어를 중심으로 진행되었다. 최근 Google, Adobe, Snapchat 등의 글로벌 IT 기업들은 딥러닝을 폰트 인식에 적용해 수 천 종의 폰트에 대해서도 좋은 성과를 얻었다[1]. 그러나, 한글은 영어권 언어보다 문자 수가 훨씬 많기 때문에 한글 폰트 인식은 영어권 폰트 인식보다 어렵다. 과거에 이루어진 대부분의 한글 폰트 인식 연구들은 불과 수 십 종의 폰트 만을 인식 대상으로 하고 있다.

본 연구에서는 최근 영상인식 분야에서 좋은 성능을 보이고 있는 CNN을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 두 가지 다른 구조의 CNN을 이용해 폰트인식기를 구성하고, 실험을 통해 이들을 비교 평가하였다. 특히, 본 연구에서는 폰트 인식에 필요한 지역적 세부 특징을 효과적으로 추출하면서도 학습 시간과 파라미터의 수를 줄이는 방향으로 모델을 개선하였다. 제안하는 모델은 3300종의 한글 폰트에 대하여 상위 1위 인식률 94.55%, 상위 5위 인식률 99.91%의 성능을 보였다.

2. 관련 연구

최근 해외에서는 딥러닝을 이용한 폰트 인식 방법들이 제안되었다. [1]에서는 영상인식에 성능이 우수한 딥러닝 모델인 CNN(Convolutional Neural Network)을 이용해 2383종의 폰트를 인식하였다. 학습 데이터로는 기본 폰트 영상에 다양한 변형을 적용함으로써 더 많은 영상을 추가해 사용하였다. 또한, 원활한 학습을 위해 SCAE (Stacked Convolutional Auto-Encoder)를 이용해 사전 학습을 수행한 후, 폰트 인식을 위한 교사 학습을 진행하였다.

해외에서는 폰트 인식 연구가 활발히 이루어졌으며, 딥러닝을 비롯한 첨단 기술들도 많이 적용되었다. 그러나, 국내에서는 폰트 인식 연구가 많이 이루어지지 않았을 뿐 아니라 주로 전통적인 특징 추출 알고리즘에 의존하고 있다. 한글은 2350개의 문자를 포함하고 있으며 현재 한국에서는 3000종이 넘는 한글 폰트들이 사용되고 있다. 문자 및 폰트의 종류가 많을수록 폰트를 정확히 인식하는 것은 어렵다. 따라서, 한글 폰트를 인식하는 것은 영어권 폰트 인식에 비해 난도가 높다. 지금까지 수행된 대부분의 한글 폰트 인식 연구들은 불과 수 종의 폰트 만을 인식 대상으로 하였다. 1997년도에 진행된 연구에서는 한글 문서의 폰트를 MLP(Multi-layer Perceptron)를 이용해 학습하였다[2]. MLP를 학습시키기 위해서 몇 가지 특징을 사용했는데, 문서에서 일정한 크기의 블록을 추출해서 수직방향과 수평방향으로 FFT(Fast Fourier Transform)를 수행한 후, 각 방향에 대해서 평균을 취하고, 그 결과 중에서 64개의 특징 값을 추출했다. 이와 같이 추출한 특징을 이용해 명조체, 신명조체, 견명조체, 고딕체, 중고딕체, 견고딕체, 궁서체, 샘물체, 필기체, 그래픽체라는 한글 문서의 기본이

되는 10가지 폰트를 인식하여 평균 95.19%의 인식률을 얻었다.

한글 문자 인식 연구는 폰트 인식보다는 많이 수행되었다. [3]에서는 CNN을 이용해 필기 한글을 인식하였는데, 4개의 컨볼루션계층과 4개의 맥스풀링(max-pooling)계층, 그리고 2개의 완전연결계층(fully-connected)으로 구성된 CNN을 이용하였다. 이 연구에서는 520자의 조합에 대한 성능은 97.67%, 2350자의 조합에 대한 성능은 96.34%의 정확도를 보였다.

3. 시스템 구성

3.1. 데이터 구성

본 연구는 한글 폰트 3300종을 인식 대상으로 한다. 각 폰트는 한글 2350자를 포함하고 있으므로 총 7,755,000(3300x2350)가지 폰트-문자 조합이 존재한다. CNN의 학습과 평가에는 48x48 크기의 문자 영상들을 사용하였다. 총 7,755,000개의 문자 영상들을 학습 데이터, 검증 데이터, 평가 데이터로 나누었으며, 각각의 데이터 셋은 전체 데이터의 80%, 10%, 10%의 비율로 랜덤 분할하였다.

3.2. 폰트 인식 CNN의 구성

본 연구에서는 두 가지 모델을 사용하였는데, 각 모델의 구성은 그림 1과 같다. 폰트를 효과적으로 인식하기 위해서는 문자 영상의 지역적인 세부 특징을 추출해야 한다. 이를 위해 본 연구에서 사용한 CNN은 다음과 같은 특징을 갖는다.

3.2.1. CNN 기본 모델

CNN의 상위 계층에서는 고수준 특징(high-level feature)을 추출하고 하위 계층에서는 저수준 특징(low-level feature)을 추출한다[4]. 고수준 특징은 영상의 전역적인 특성을 잘 표현할 뿐 아니라, 변이에 강한 장점이 있다. 반면 저수준 특징은 지역적인 세부 형태를 잘 반영하는 장점이 있다. ImageNet 데이터 등 복잡한 영상에 사용되는 VGG, GoogLeNet 등의 CNN들이 매우 많은 수의 계층으로 구성된다[5][6]. 그러나, 본 연구에서는 지역적 세부 형태를 잘 추출하기 위해 비교적 적은 수의 계층으로 이루어진 [3]의 모델을 기본 모델로 택하였다. 적은 수의 계층을 사용할 경우 폰트 인식에 필요한 저수준 특징을 잘 반영할 수 있을 뿐 아니라 학습 시간과 파라미터의 수를 줄이는 데에도 바람직하다. 그러나, [3]의 모델은 한글 인식을 위해 설계되었으므로 폰트 인식에 좀 더 적합하도록 다음과 같은 변형을 적용하였다.

3.2.2. 모두 컨볼루션으로 구성된 네트워크

과거의 한글 인식 연구에서는 맥스풀링(max-pooling)계층을 통해 컨볼루션 계층에서 추출한 특징 벡터를 추상화하고 특징맵의 크기를 축소하였다[3]. 맥스풀링계층은 차원 축소 및 추상화 과정에서 특징의 위치 변이를 흡수하는데, 그 결과 폰트 인식에 필요한 지역적 세부 형태 정보가 소실된다. 본 연구에서는 이러한 문제점을 극복하기 위해 [7]과 같이 맥스풀링계층을 모두 동일한 크기의 커널과 보폭을 갖는 컨볼루션계층으로 대체했다.

3.2.3. 잔류 연결(Residual Connection)

폰트 인식에는 획에서의 미세한 차이로도 폰트의 종류가 달라지기도 한다. 따라서, 폰트를 효과적으로 인식하기 위해서는 추상화 수준이 높은 고수준 특징뿐 아니라 세부 형태를 반영하는 저수준 특징들도 요구된다. 이를 위해 본 연구에서는 저수준 특징들이 정보를 보존한 상태로 상위 계층까지 전달되기 위해 잔류 연결(residual connection)을 적용했다. 심층신경망이 잔류 연결을 포함할 경우 얕은 네트워크를 병렬적으로 연결한 것과 유사한 효과를 얻을 수 있는데[8], 그로 인해 하위 계층들이 추출한 저수준 특징들을 상위 계층까지 잘 전달할 수 있다.

3.2.4. 전역 평균 풀링(Global Average Pooling)

CNN의 완전연결계층은 파라미터의 수가 매우 많아 과적합(over-fitting)이 많이 발생하는 것으로 알려져 있다. 이를 완화하기 위해 본 연구에서는 [9]과 같이 완전연결계층 대신 CCCP(Cascaded Cross Channel Pooling) 계층과 전역평균풀링(global average pooling)을 사용하였다.

3.2.5. 폰트 인식 CNN의 학습

CNN의 학습 알고리즘으로는 RMSProp(Root Mean Square Propagation) 최적화 알고리즘과 모멘텀(momentum) 최적화 방법을 결합한 ADAM 최적화(ADAptive Momentum estimation optimizer) 알고리즘 [12]을 사용하였다. ADAM 최적화는 최근 많은 연구에서 좋은 성능을 보이고 있다.

또한, Xavier 초기화와 배치정규화를 함께 사용함으로써 학습 속도를 개선하였다. 최근에는 많은 수의 계층으로 구성된 CNN에서는 He 초기화를 Xavier 초기화 알고리즘보다 더 많이 사용하는 추세이다[10][11]. 그러나, 본 연구에서 두 알고리즘을 적용해 본 결과 Xavier 초기화 알고리즘이 근소하게 좋은 성능을 보였다. 이는 본 연구에서 사용한 CNN이 비교적 적은 수의 계층으로 구성되었기 때문으로 추정된다.

다수의 계층으로 구성된 CNN의 학습에는 내부 공변량 이동(internal covariate shift)문제가 발생한다. 이를 해결하기 위한 방법으로 배치정규화가 널리 사용된다[13]. 3300가지 폰트를 분류해야 하기 때문에, 초기값과 학습률을 세부적으로 고려하여 설정해주지 않으면 경사도(gradient)가 폭발적으로 증가하거나 소실될 수 있다. 배치정규화는 각 배치(batch)와 계층마다 정규화를 수행함으로써 이러한 문제들을 해결해주며 학습 시간도 크게 단축시킨다. 동일한 모델을 학습 할 경우에도 배치정규화를 사용할 경우 3배 이상의 수렴속도를 보였다.

동일한 폰트 내의 문자 영상들은 지역적 형태가 매우 유사하기 때문에 폰트인식기의 학습에서는 학습 데이터들의 배열에 섬세한 주의가 요구된다. 또한, 학습에

사용되는 배치정규화(batch normalization) 알고리즘은 각 배치별로 특징들의 분포를 추정하기 때문에, 모든 폰트 3300종에 대해 한글 조합 2350자가 고르게 섞이도록 하는 것이 중요하다. 본 연구에서는 많은 수의 문자 영상들을 고르게 분포하도록 하기 위해 모든 폰트-한글 조합에 대한 인덱스를 만들고, 매 반복마다 고르게 섞은 후 각 인덱스가 가리키는 폰트-문자 영상을 읽어와 학습에 사용하였다.

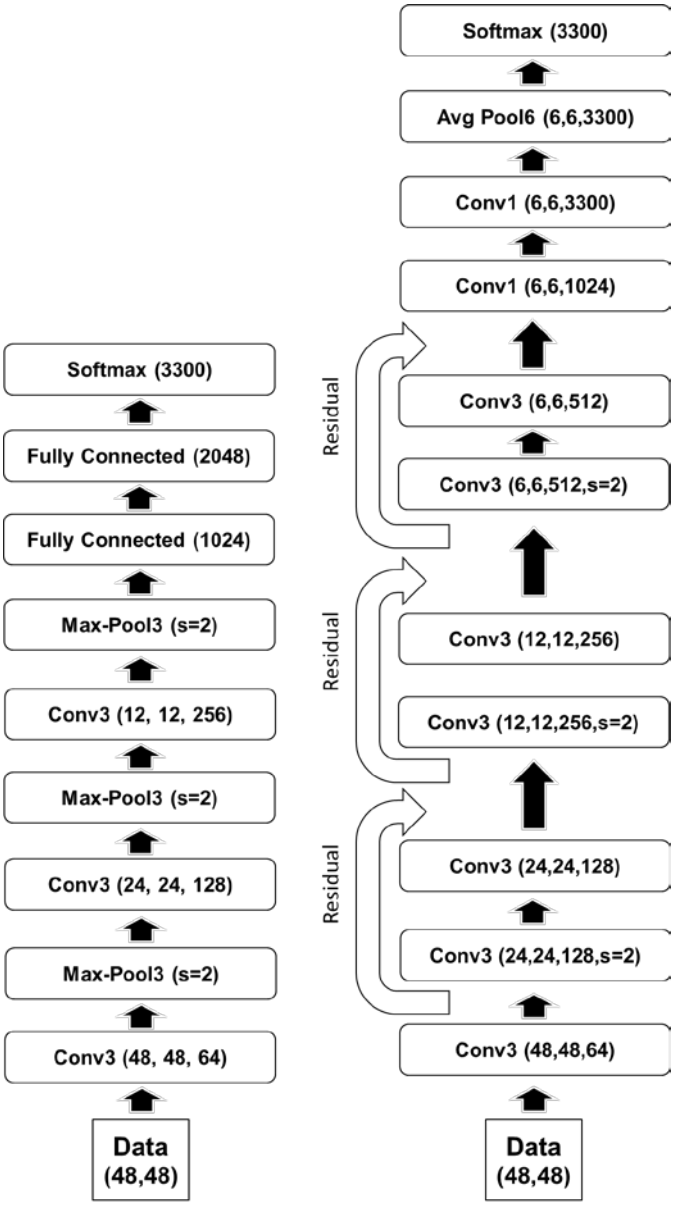

그림 1 모델 구성. 기본 모델 A(좌), 제안하는 모델 B(우).

각 상자 안의 이름은 계층의 종류, 그 옆의 숫자는 커널의 크기, 괄호안의 숫자는 차례대로 이미지의 높이, 넓이, 채널(혹은 노드)의 개수이다. 모든 은닉계층의 활성화 함수는 ReLU를 사용했다.

4. 실험 결과

표 1 두 모델의 폰트 인식률

	기본 모델 (A)	제안하는 모델 (B)	상대적 오차감소율
상위 1위 인식률	88.29%	94.55%	53.46%
상위 5위 인식률	99.03%	99.91%	90.72%

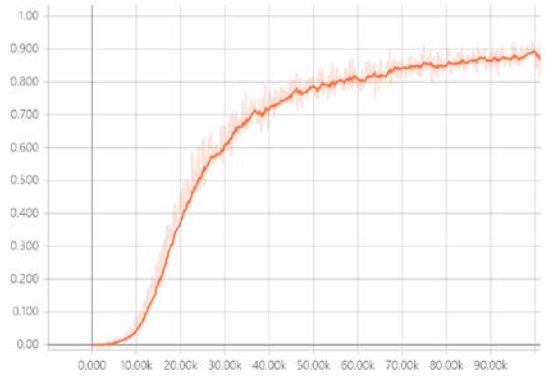

그림 2 모델 A 학습 과정의 인식률 변화

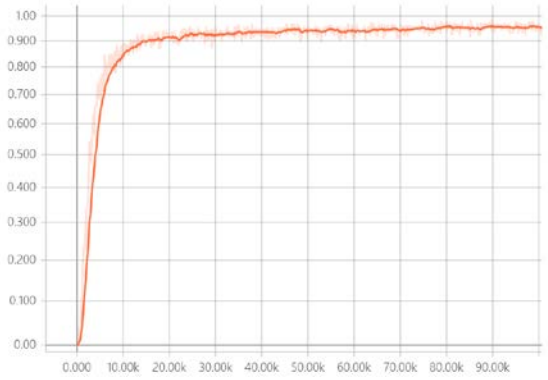

그림 3 모델 B 학습 과정의 인식률 변화

4.1. 실험 환경

학습을 위한 실험 환경으로 Intel i7-6700K 4.00GHz CPU, GeForce GTX-1080 GPU 2개, SSD, 32GB Ram를 사용하여 실험을 진행했다. 두 가지 모델에 대해 학습을 위한 배치의 크기는 256, 학습의 횟수는 10만 번으로 동일하게 실험했다. 모델 A는 학습률을 0.001로 설정하고 학습하였고, 모델 B는 배치정규화를 이용했기 때문에 높은 학습률인 0.1로 설정하고 학습했다. 배치정규화를 적용하지 않은 모델 A에 높은 학습률을 적용할 경우 학습이 진행되지 않았다.

4.2. 모델 별 실험 결과

그림 2와 그림 3은 모델 A와 모델 B의 학습 횟수에 따른 인식률의 변화를 나타낸 그래프이다. x축은 학습 횟수이며, y축은 인식률(%)이다. 모델 B의 수렴속도가

모델 A의 수렴속도보다 훨씬 빨랐다. 특히, 배치정규화를 적용하고 학습률을 높게 설정한 것이 수렴 속도에 큰 영향을 미쳤다.

표 1은 두 모델에 대해 상위 1위 인식률과 상위 5위 인식률에 대한 실험 결과이다. 모델 A보다는 모델 B가 더 우수한 성능을 보였다. 1위 인식률과 5위 인식률의 절대적 오류 감소는 각각 6.26%와 0.88%였다. 그러나, 상대적 오차 감소율은 1위 인식률이 53.46%와 5위 인식률이 90.72%로 나타나 3장에서 기술한 방법들이 폰트 인식 성능 개선에 효과적이었음을 확인할 수 있었다.

그림 4 폰트 인식 결과 예. 인식률 상위 2위의 폰트가 정답 폰트와 얼마나 유사한지를 나타내는 예시. 폰트 별로 3개의 이미지가 있으며, 점선을 기준으로 왼쪽은 정답 폰트와 인식률, 오른쪽은 해당 폰트에 대한 예측률이 높은 상위 2개의 폰트와 예측률. 파란색 글자는 정답 폰트와 일치하는 예측 폰트. 3300가지의 폰트에 대한 예측 시각화 중 12가지의 샘플 예시.

오인식 분석 결과 거의 같은 모양을 가진 폰트가 많았는데, 이러한 폰트들은 육안으로도 구분하기 어려웠다. 그림 4는 폰트 인식 결과의 예이다. 일부 폰트들은 형태가 단순함에도 불구하고 인식률이 낮게 측정되었는데, 이러한 폰트들은 다른 폰트와 매우 유사하거나 아예 동일한 형태를 갖는 경우가 많았다. 이러한 폰트들은 사실상 구분이 매우 어려웠다. 반면, 형태가 복잡한 그림체 폰트들은 동일한 형태의 폰트가 없어서 높은 인식률을 보이는 경우도 있었다.

5. 결론

본 논문에서는 CNN을 이용해 3300종의 한글 폰트를 인식하였다. 지역적인 세부 특징을 효과적으로 추출하기 위해 비교적 적은 수의 계층을 사용하였으며, 최상단 계층 외에는 모두 컨볼루션 계층으로만 구성된 CNN을 사용하였다. 또한, 저수준 특징이 최상단까지 잘 전달되도록 하기 위해 잔류 연결을 적용하였다. 그 결과 3300종의 폰트 전체에 대하여 1위 인식률 94.55%, 5위 인식률 99.91%의 높은 정확도를 보였다.

감사의 글

- 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2017년도 문화기술 연구개발 지원사업으로 수행되었음 (2017000130)

- 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음

참고문헌

[1] Z. Wang, et. al., "Deepfont: Identify your font from an image," Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015.

[2] 박문호, et. al., "인공지능: 인쇄된 한글 문서의 폰트 인식.", 정보처리학회논문지 제4권 8호, pp. 2017-2024., 1997.

[3] I. Kim, C. Choi, and S. Lee, "Improving discrimination ability of convolutional neural networks by hybrid learning," International Journal on Document Analysis and Recognition (IJDAR), vol. 19, no.1, pp. 1-19, 2016.

[4] Ranjan, Rajeev, Vishal M. Patel, and Rama Chellappa. "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," arXiv preprint arXiv:1603.01249, 2016.

[5] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[6] C. Szegedy, et. al., "Going deeper with convolutions," Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[7] J. Springenberg, et al. "Striving for simplicity: The all convolutional net," arXiv preprint arXiv:1412.6806, 2014.

[8] A. Veit, M. Wilber, and S. Belongie, “Residual Networks Behave Like Ensembles of Relatively Shallow Networks,” Advances in Neural Information Processing Systems. 2016.

[9] M. Lin, Q. Chen, S. Yan, “Network In Network,”

arXiv preprint arXiv:1312.4400, 2014.
[10] X Glorot, Y Bengio, “Understanding the difficulty of training deep feedforward neural networks,” PMLR, pp. 249-256, 2010.
[11] K. He, X. Zhang, S. Ren and J. Sun, “Delving deep into rectifiers surpassing human-level performance on imagenet classification,” arXiv preprint arXiv:1502.01852, 2015.
[12] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[13] S. Ioffe, and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning, 2015.