

CNN기반 딥러닝을 이용한 Kuzushiji-MNIST/49 분류의 정확도 향상을 위한 학습 방안

박병서¹ · 이승영² · 서영호^{3*}

Training Method for Enhancing Classification Accuracy of Kuzushiji-MNIST/49 using Deep Learning based on CNN

Byung-Seo Park¹ · Sungyoung Lee² · Young-Ho Seo^{3*}

¹Graduate Student, Department of Electronic Material Engineering, Kwangwoon University, Seoul, 01897 Korea

²Professor, Ingenium College of Liberal Arts, Kwangwoon University, Seoul, 01897 Korea

^{3*}Professor, Department of Electronic Material Engineering, Kwangwoon University, Seoul, 01897 Korea

요 약

본 논문에서는 고대 및 중세 시대의 일본 문자에 대한 데이터셋인 Kuzushiji-MNIST와 Kuzushiji-49를 정확하게 분류하기 위한 딥러닝 학습 방법에 대해서 제안한다. 최신의 합성곱 신경망 네트워크들을 분석하여 가장 적합한 네트워크를 선별하고, 이 네트워크를 이용하여 Kuzushiji-MNIST와 Kuzushiji-49 데이터셋을 분류하기 위한 학습 횟수를 선정한다. 또한 Mixup과 Random Erase 등의 학습 방법을 적용하여 높은 정확도를 갖도록 학습을 진행한다. 학습 결과를 살펴보면 MNIST에 대해서는 99.75%, K-MNIST에 대해서는 99.07%, 그리고 K-49에 대해서는 97.56%의 정확도를 보임으로써 제안한 학습 방법이 높은 성능을 보일 수 있음을 증명하였다. 이와 같은 딥러닝 기반의 기술을 통해 동아시아와 서양의 역사, 문학, 그리고 문화를 연구하는 다양한 연구자들에게 좋은 연구 기반을 제공할 것으로 사료된다.

ABSTRACT

In this paper, we propose a deep learning training method for accurately classifying Kuzushiji-MNIST and Kuzushiji-49 datasets for ancient and medieval Japanese characters. We analyze the latest convolutional neural network networks through experiments to select the most suitable network, and then use the networks to select the number of training to classify Kuzushiji-MNIST and Kuzushiji-49 datasets. In addition, the training is conducted with high accuracy by applying learning methods such as Mixup and Random Erase. As a result of the training, the accuracy of the proposed method can be shown to be high by 99.75% for MNIST, 99.07% for Kuzushiji-MNIST, and 97.56% for Kuzushiji-49. Through this deep learning-based technology, it is thought to provide a good research base for various researchers who study East Asian and Western history, literature, and culture.

키워드 : 합성곱 신경망, 딥러닝, Kuzushiji-MNIST, 학습, 분류

Keywords : convolutional neural network, deep learning, Kuzushiji-MNIST, training, classification

Received 26 October 2019, Revised 11 December 2019, Accepted 14 February 2020

* Corresponding Author Young-Ho Seo (E-mail:seo@kw.ac.kr, Tel:+82-2-940-8362)

Professor, Department of Electronic Material Engineering, Kwangwoon University, Seoul, 01897 Korea

Open Access <http://doi.org/10.6109/jkiice.2020.24.3.355>

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

최근의 인공지능은 심층 신경망의 형태로 구성되어 있는데 이를 학습시키고 활용하는 것을 딥러닝(deep learning)이라 한다. 대용량의 데이터베이스가 확보되면 딥러닝을 이용하여 인공지능을 학습시킬 수 있다. 인공지능을 학습시키는 방법은 다양한데, 본 논문에서는 가장 널리 사용되는 지도학습법을 사용한다. 지도학습은 인공지능에게 문제와 정답(데이터셋)을 알려주어 스스로 문제로부터 정답을 풀 수 있는 능력을 갖도록 만드는 것(훈련)이다. 좋은 문제와 정답(데이터셋)이 많으면 많을수록(빅데이터) 스스로 문제를 푸는 능력은 향상될 수 있다.

딥러닝은 다양한 학문 분야에서 활용될 수 있다. 일례로 최근 2018년에 일본, 캐나다 및 미국의 연구자들은 공동연구로 그림 1과 같은 Kuzushiji-MNIST(K-MNIST) 및 Kuzushiji-49(K-49)라는 데이터베이스를 구축하였다. Kuzushiji 문자는 약 1,000여 년 동안 사용해 왔음에도 불구하고 현대 일본에서는 독해의 곤란과 관련 전문가의 부재 등으로 가르치고 있지 않다. 그 결과 대부분의 일본인들은 필기체로 작성된 약 150년 이전의 Kuzushiji 서적들을 읽지 못하는 실정이다. 데이터베이스에 대한 연구의 착수는 이러한 문제의식으로부터 촉발되었다[1].

한국의 문자가 아님에도 일본의 고대 및 중세 문자가 중요한 이유는 고대 및 중세 일본이 서양의 문물을 먼저 다양하고 빈번하게 받아들임으로써 그 문자로 작성된 다양한 문헌에 동아시아와 서양의 역사 및 문화를 설명하고 있는 정보를 다수 담고 있기 때문이다. 기록되어 있는 역사적 문서들은 우리에게 과거를 엿볼 수 있게 해준다. 우리는 이전 시대의 세상을 엿볼 수 있도록 해주고, 우리 자신의 문화, 규범, 그리고 우리에게 반영된 가치를 간접적으로 유추해 볼 수 있다.

일본의 경우, 1868년의 메이지 유신을 통해 문화의 현대화를 위한 교육 시스템의 개혁을 시도하여 필기 및 인쇄 시스템에도 커다란 변화를 맞이하였다. 일본어 현대화로 인해 공식 학교 교과 과정에서 필기체인 Kuzushiji 스크립트도 더 이상 가르치지 않게 되었다. Kuzushiji는 1000년 이상 사용되었지만 결과적으로 오늘날 대부분의 사람들은 150년 전에 쓰이거나 출판된 문헌들을 해석할 수 없는 상황이다[2][3].

상기와 같은 역사적 배경 하, 중세 문자에 대한 연구와 함께 그 문자들을 인식하기 위해서는 본 연구에서와 같은 딥러닝의 특성을 숙지하고 본 연구에 적용할 것이 필요하다. 이 경우, 본 연구에서는 합성곱 심층 신경망(convolutional neural network, CNN) 형태의 딥러닝 기법을 이용하고자 한다.

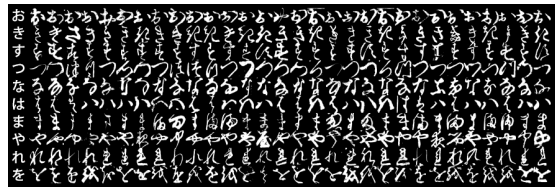


Fig. 1 Example of Kuzushiji-MNIST database

2012년 AlexNet[4]이 도입된 이후 CNN은 이미지 분류의 지배적인 접근 방식이 되었다. 그 이후로 VGG[5], NiN[6], Inception[7][8], ResNet[9], DenseNet[10] 및 NASNet[11]을 포함하여 다양한 새로운 네트워크가 제안되었고 이와 함께 정확도도 꾸준한 개선 추세를 보였다. 본 논문에서는 K-MNIST에 적합한 다양한 학습 방법을 적용함으로써 모델 정확도는 향상시키지만 계산 복잡도는 거의 변경하지 않는 학습 방법에 대해 제안하고자 한다. 이러한 방법들은 대체로 모델의 선택을 포함하여 학습률을 조정하거나, 전처리 기법을 적용하는 것과 같은 일종의 “트릭”에 해당한다. 그러나 이러한 사소한 조정을 통해서 학습의 정확도를 가져올 수 있는 것이 딥러닝의 특성 중의 하나이다[12].

본 논문에서는 서장에 이어 2장에서는 K-MNIST와 주로 사용될 네트워크에 대해서 설명한다. 3장에서는 제안한 학습 기법에 대해서 설명하고 4장에서 실험 결과 및 비교 결과를 제시한다. 마지막으로 5장에서는 결론을 맺는다.

II. CNN 기반의 딥러닝

2.1. Kuzushiji-MNIST의 소개

K-MNIST 데이터 세트는 NIJL(National Institute of Japanese Literature, 국립 일본 문학원)에서 만들어졌고, CODH(Center for Open Data in the Humanities, CODH)에서 최종적으로 완성되었다[1]. NIJL과 다른 기관들은

2014년에 약 30만권의 오래된 일본 서적을 디지털화하여 그 중 일부를 번역하고 국제 협력 증진을 위한 공개 데이터로 공유하는 국가 프로젝트를 시작했다. 필사 과정에서 각 글자에 대한 데이터베이스가 만들어졌지만, 활용되지 않고 있다가 기계 학습 관점에서 여러 기계 학습 문제의 기초로 사용하기 위해 CODH에 의해 별도의 데이터 세트로 만드는 것이 제안되었고, 이로 인해 Kuzushiji 데이터 세트가 2016년 11월에 발표되었다. 전체 데이터 세트에는 3,999개의 문자 유형과 403,242개의 문자가 포함되어 있다[13].

이 데이터세트는 머신러닝과 고대 및 중세 일본어의 해석이 쉽게 가능하도록 만들어진 것으로 18세기에 인쇄된 35권의 고전 서적에서 스캔한 문자를 사전 처리하여 3개의 데이터 세트로 구성되었다. 첫 번째는 MNIST[14] 데이터 세트를 대체하는 K-MNIST, 두 번째는 48개의 히라가나 문자와 하나의 히라가나 반복 마크를 포함하는 훨씬 크지만 불균형한 K-49이다. 마지막으로 표본이 거의 없는 희귀 문자를 포함하여 3,832개의 한자 문자로 구성된 Kuzushiji-Kanji이다. K-49와 Kuzushiji-Kanji는 데이터세트의 구성이 불균형하기 때문에 딥러닝 결과에 대한 신뢰도가 낮다. 특히 Kuzushiji-Kanji 데이터는 그 구성이 매우 불균형하기 때문에 실험에서 제외하였다.

Table. 1 Content of dataset

Item	The Number of Dataset	
	Kuzushiji-MNIST	Kuzushiji-49
Training images	60,000	232,365
Training labels	60,000	232,365
Testing images	10,000	38,547
Testing labels	10,000	38,547

MNIST는 10개의 클래스로 제한되며 Kuzushiji 히라가나를 완전히 표현하는데 필요한 클래스 49개 보다 클래스의 수가 훨씬 적기 때문에 Kuzushiji-MNIST를 만들 때 히라가나의 10행 각각을 나타내는 하나의 문자를 선택하였다. 표 1 과 같이 K-49에는 49개의 클래스 (266,407개 이미지)가 있다. Kuzushiji-Kanji는 총 3,832개의 클래스 (140,426개 이미지)가 있으며, 1,776의 이미지로 구성된 클래스부터 단일 이미지로 구성된 클래스까지 다양한 데이터 분포를 보인다. Kuzushiji-MNIST

및 K-49는 MNIST 데이터 세트와 일치하는 28×28 픽셀 해상도의 그레이 스케일 이미지로 구성되어 있으며, Kuzushiji-Kanji 이미지는 64×64 픽셀의 해상도가 더 큰 이미지로 구성되어 있다.

K-MNIST는 클래스 간 데이터 구성의 균형을 유지하는 반면, K-49는 적은 수의 데이터로 구성된 클래스(예: ~“츠”만 있는 400개 이미지)와 희귀 문자로 이루어진 클래스로 인해 불균형한 데이터 구성을 가지고 있다. Kuzushiji-Kanji 또 한 Kuzushiji 문헌에 나타나는 Kanji의 고유 빈도로 인한 클래스간 불균형한 데이터 구성을 가지고 있다.

각 데이터 세트의 트레이닝 세트와 테스트 세트는 동일한 35권의 책으로 부터 샘플링 되므로 일관된 데이터 분포를 가지고 있다. K-MNIST의 MNIST를 기반으로 하는 데이터 세트의 구성은 Fashion-MNIST[15]로부터 영감을 얻어 개발되었다.

2.2. 딥러닝 네트워크

2014년 ILSVRC에서 1등을 수상한 GoogLeNet의 인셉션 모듈(Inception Module)이 참신한 구조 변화를 보여주었다. GoogLeNet은 비록 분류의 측면에서는 좋은 성능을 내었지만 구조의 복잡성으로 인해 활용도는 낮았다. 이러한 이유로 오히려 2등을 수상한 VGGNet이 더 많이 활용되었다. 이후 2015년에 은닉층을 많이 갖는 딥러닝 구조에서 가장 큰 문제점이었던 기울기 소실 문제를 다소 해결한 ResNet이 제안되었다. ResNet은 2015년 ILSVRC에서 오류율 3.6%로 1등을 차지하였다.

AlexNet이 처음 제안된 이후로 CNN구조의 은닉층은 점점 더 깊어졌고, 층이 깊어질수록 역전파 시 기울기 값이 중간에 소실되어 학습이 제대로 이루어지지 않는 문제(gradient vanishing)가 발생했다. ResNet의 핵심 알고리즘은 그림 2의 잔차블록(residual block)이다. 또한 기울기 값이 잘 전파될 수 있도록 일종의 지름길(shortcut, skip connection)을 만들었다. ResNet의 성능이 좋은 이유는 기울기 문제 외에 잔차블록이 앙상블(ensemble) 모델을 구축한 것과 비슷한 효과를 내기 때문이다. 잔차블록의 건너 뛰기 연결(skip connection)로 인해 입력데이터와 기울기가 오갈 수 있는 통로가 크게 늘어난다.

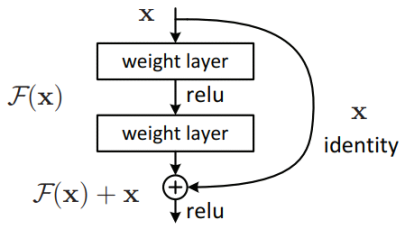


Fig. 2 Residual block of ResNet

GoogLeNet 연구팀은 인셉션의 구조를 개량하여 다음 버전의 인셉션 구조를 꾸준히 발표하였다. VGGNet의 개념을 발전시켜 합성곱 커널에 인수분해 개념을 포함시켜 은닉층을 더욱 깊게 만들었다[8]. 인셉션의 첫 번째 버전에서 사용되었던 배치 정규화를 기본적으로 사용하면서 인수분해와 라벨 스무딩, 그리고 보조 분류기(Auxiliary classifier)를 포함시켜 Inception-V3를 제안하였다. 그림 3에 인셉션의 은닉층의 노드를 나타내었다. 세 번째 인셉션은 ResNet[9]과 거의 유사한 성능을 갖고 있다.

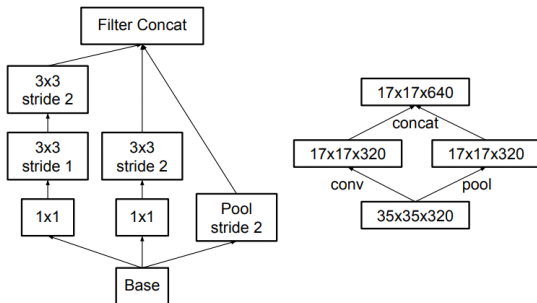


Fig. 3 Convolution of Inception-V3

DenseNet[10]은 ResNet에서 한발 더 나아가 전체 네트워크의 모든 층과 통하는 지름길을 추가하였다. DenseNet의 전체적인 아키텍처는 그림 4와 같다. ResNet은 네트워크에 건너 뛰기 연결(skip connection)을 추가해서 기울기 소실 문제(degradation)를 해소하고자하였는데 DenseNet에서는 고밀도 연결(Dense connectivity)이라는 새로운 개념을 추가하였다. 고밀도 연결 패턴(Dense connectivity pattern)은 몇 가지 중요한 장점을 갖는다. 첫 번째는 모든 피쳐 맵(feature map)들을 쌓아오기 때문에 레이어와 레이어 사이 최대한 가치 있는 정보가 이동할 수 있다. ResNet같은 경우는 네트워크 시작 지점 레이어의 정보가 깊은 망을 거치면서 뒤쪽

으로 갈수록 희미해지는 것을 건너 뛰기 연결(skip-connection)로 어느 정도 해결을 하는 반면에, 밀도 연결 패턴(dense connectivity pattern)을 사용하면 아예 초반 레이어의 정보를 쌓아가며 뒤쪽 레이어 까지 전달할 수 있다. 즉, 오차를 다시 역전파 할 때도 더 효율적으로 전달할 수 있기 때문에 학습에 더욱 용이하다.

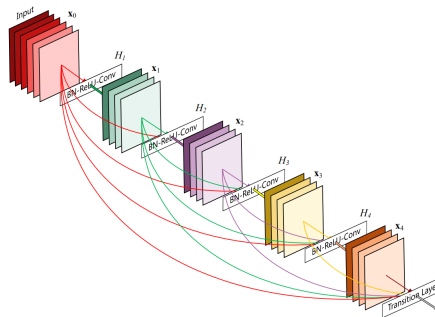


Fig. 4 Network structure of DenseNet

두 번째는 다른 네트워크보다 파라미터의 수를 많이 줄일 수 있다는 것이다. DenseNet의 레이어들은 다른 네트워크들에 비해 좁다. 한 레이어당 대략 12개 정도의 필터를 가지며 이전의 피쳐 맵들을 계속 쌓아가면서 전체적인 네트워크 내부의 정보들을 효율적으로 전달한다. 이는 마지막 분류기(classifier)에 네트워크 내부의 전체 피쳐 맵을 골고루 입력할 수 있게 하여 전체 파라미터의 개수가 줄었음에도 불구하고 네트워크가 충분히 학습이 가능하게 한다. 이것은 저수준 피쳐와 고수준 피쳐를 효율적으로 섞어서 사용하는 것이라고 볼 수 있다. 마지막으로 고밀도 연결(dense connection)이 정규화 효과를 가지고 있어서 작은 데이터 셋에서도 과적합되는 것을 줄여준다.

본 논문에서 우리는 이러한 주요 CNN 구조를 이용하여 K-MNIST를 분류하기 위한 학습을 수행할 것이고, K-MNIST에 가장 적합한 모델을 선정하고 실험을 수행할 것이다.

III. 제안한 훈련 방법

3.1. 기본 학습 방법

본 논문에서는 미니 배치를 이용한 확률적 경사 하강법을 사용하고 있으며, 신경망을 학습시키는 알고리즘

을 그림 5에 나타냈다[12]. 각 반복에서 무작위로 b개의 이미지를 선택하고, 전처리(Preprocess)를 수행했다. 다음에 순전파를 시킨 후에(Forward) 경사도를 계산한 다음(Loss) 네트워크 파라미터를 업데이트(Update)한다. 모든 데이터를 이용하여 학습을 한번 수행한 것을 epoch라 한다. epoch는 데이터셋의 조건과 학습 상황에 따라서 결정된다.

```

initialize(net)
for epoch = 1 ~ K do
  for batch = 1 ~ #(images/b) do
    Images ← Uniformly random sample b images
    X, y ← Preprocess(Images)
    z ← Forward(net, X)
    ℓ ← Loss(z, y)
    Grad ← Backward(ℓ)
    Update(net, grad)
  end for
end for
    
```

Fig. 5 Training algorithm

3.2. 학습 길이의 결정

가장 먼저 학습의 길이, 즉 몇 번 학습할 것인지 epoch 수를 결정하기 위한 실험을 진행한다. 이때 기본 CNN에 두 가지 기법을 적용하였다. 첫 번째는 학습률 Warmup인데 이 방식은 초기에 학습률을 0으로 설정하고 이를 일정 기간동안 선형적으로 증가시키는 방식이다. 두 번째는 No bias decay 기법으로 L2 정규화와 가중치 및 편차의 정규화에 모두 적용한다. 가중치에만 decay를 주는 것이 과학습을 방지하는데 효과적이라는 연구결과에 따라서 본 논문에서는 다소 경험적으로 적용하였다. KMINST에 가장 적합한 CNN 모델을 선정하기 위해서 기본적인 기법들을 적용하고, 학습횟수에 따른 정확도를 측정하였다. CNN 모델에는 MobileNet[16], ShuffleNet[17], ShuffleNet-V2[17], Resnet-50[18], Inception-V3[8], 그리고 DenseNet[10] 등을 이용하였다. 표 2에 다양한 CNN을 이용한 epoch에 따른 학습 결과를 정리하였다. 표 2의 결과에서 나타나는 것과 같이 DenseNet이 가장 좋은 성능을 보였고, epoch은 300이 되어야 최고의 정확도를 보였다.

Table. 2 Training result according to epoch by using various CNN

Network [1]	K-MNIST (100 epoch)	K-MNIST (300 epoch)
MobileNet [16]	91.73%	
ShuffleNet [17]	95.73%	
ShuffleNet-V2 [17]	96.34%	
Resnet-50 [18]	97.49%	98.59%
Inception-V3 [8]	97.48%	98.12%
DenseNet [10]	98.19%	98.77%

3.3. 학습 기법

본 논문에서는 몇 가지 기법을 활용하여 학습 효과의 상승을 시도하였다. 학습 효율을 높이는 방법은 다양하다. 최근 아마존에서 발표한 CNN 학습 방법에 대한 논문[12]에서는 Cosine Learning Rate Decay, Label Smoothing, Knowledge Distillation, 그리고 Mixup Training 기법들이 최근 CNN 모델들에 가장 적합한 학습방법이며, 그 효과의 우수성을 주장하였다. 본 연구팀은 이러한 기법들 중 Mixup 훈련 방법[19]과 Random Erase 기법[20]을 적용하여 K-MNIST 데이터 세트의 학습에 성능 향상을 확인하였다.

섞기(Mixup) 훈련 방법은 데이터셋을 무작위로 추출한 후에 2개 혹은 그 이상의 데이터를 가중 선형 보간 방법을 이용하여 새로운 데이터셋을 만든 후에 이것을 이용하여 학습을 수행하는 방법이다. 무작위로 선택된 두 개의 데이터를 (x_i, y_i) 와 (x_j, y_j) 라고 하면, 이들을 이용하여 만든 새로운 데이터는 식 (1)과 (2)와 같이 (\hat{x}, \hat{y}) 가 된다. 여기서 λ 는 0과 1에서 추출한 난수이다. 즉 Mixup 훈련 방법은 무작위로 샘플링 된 두 개의 이미지를 블렌딩 하여 하나의 입력 이미지로 만들어 데이터의 다양성을 확보하는 방법이다. K-MNIST 데이터셋에 Mixup 방법을 적용하여 생성한 데이터 예시를 그림 6에 나타냈다.

$$\hat{x} = \lambda x_i + (1 - \lambda)x_j \tag{1}$$

$$\hat{y} = \lambda y_i + (1 - \lambda)y_j \tag{2}$$

Mixup은 두 개의 샘플을 혼합한 결과에 대해 예측을 하게 하는 알고리즘이기 때문에 어떤 측면에서는 정규화로 볼 수 있고, 어떤 측면에서는 데이터 확대(data augmentation)이라고 볼 수도 있다. 이 기법은 영상에 대

해서는 잘 적용되나 다른 정보에 대해서는 잘 적용되지 않는 특성이 있다. 이것을 극복하기 위해서 나온 기법은 Manifold Mixup이다[21]. 이 방식은 입력 데이터셋을 미리 가중 선형적 합을 구하지 말고, 은닉층에서 선형적으로 보간하는 방식이다.

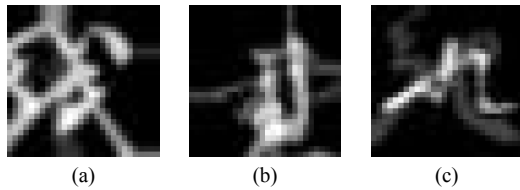


Fig. 6 Mixup result of K-MNIST

Mixup과는 다소 다른 방법이지만 효과가 유사한 잘라내기(Cutout)와 지우기(Erase) 기법이 있다[20]. 이 두 방법은 기본적으로 동일한 방식인데 삭제한 부분을 처리하는 방법이 다소 다르다. 잘라내기(Cutout)는 지운 부분을 0으로 처리하고, 지우기(Erasing)는 랜덤값으로 처리하는 것이 차이이다. 이와 같이 매우 유사하지만 두 방식의 제안 목적은 다소 다르다. 잘라내기(Cutout)는 정규화를 위한 방법으로, 지우기(Erasing)는 데이터 확대(Augmentation) 방법으로 제안되었다. 그러나 실제로 정규화와 데이터 확대(Augmentation)이 사용상에 있어서 큰 차이를 보이지는 않는다. 본 논문에서는 K-MNIST에 더욱 좋은 성능을 높이는 무작위 지우기(Random Erase) 방식을 채택하였다. Mixup과 Random Erase를 K-MNIST에 적용한 결과를 그림 7에 나타냈다.

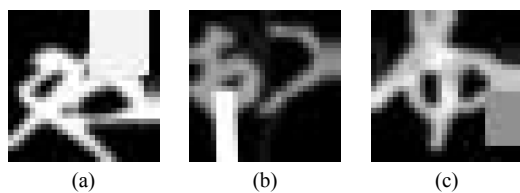


Fig. 7 Mixup result and Random Erase of K-MNIST

표 3에는 DenseNet을 이용하여 Mixup과 Random Erase를 MNIST, K-MNIST, 그리고 K-49에 각각 적용한 결과를 나타냈다. MNIST의 경우에는 99.75%의 매우 높은 정확도를 보였고, K-MNIST에 대해서도 99.07%의 매우 높은 결과를 나타냈다.

Table. 3 Training result through training technique

DenseNet 121	Accuracy
MNIST (300 Epoch + Mixup + Random Erase)	99.75%
K-MNIST (300 Epoch + Mixup + Random Erase)	99.07%
K-49 (300 Epoch + Mixup + Random Erase)	97.56%

표 4에는 DenseNet을 이용하여 학습 기법을 적용함에 따라서 K-MNIST에 대한 정확도가 높아지는 경향을 나타냈다.

Table. 4 Training result for K-MNIST

DenseNet 121	Accuracy
100 Epoch	98.19%
300 Epoch	98.77%
300 Epoch + Random Erase	98.88%
300 Epoch + Mixup	99.01%
300 Epoch + Mixup + Random Erase	99.07%

IV. 실험 결과

학습 기법들에 대한 분석 및 적용을 통해서 본 논문에서는 K-MNIST 데이터셋을 분류하는 문제를 해결하는 데에 DenseNet이 가장 적합하다는 것을 확인하였기 때문에 DenseNet을 이용하여 주로 실험을 수행하였다. 그림 8은 K-MNIST 데이터베이스를 이용한 분류 결과를 예시하였고, 그림 9는 K-49 데이터베이스를 이용한 분류 결과를 예시하였다.

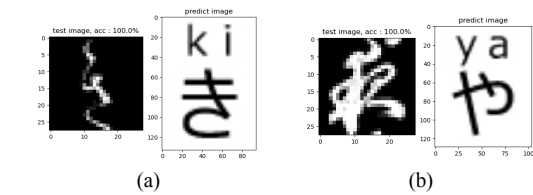


Fig. 8 Classification result of K-MNIST

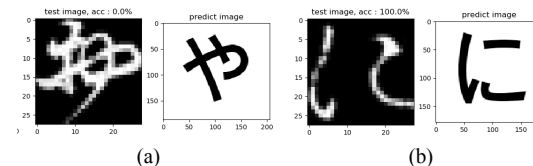


Fig. 9 Classification result of K-49

그림 10에는 DenseNet을 이용하여 학습을 수행한 과정을 그래프로 나타냈다. 그림 10(a)는 기본 모델을 사용한 것으로 epoch을 100까지 수행하여 결과를 나타냈다. 앞서 나타낸 것과 같이 이 방법의 경우에는 최대 98.19%까지의 정확도를 보인다. 그림 10(b)는 Mixup만을 적용한 것이고, 그림 10(c)는 Randm Erase만을 적용한 것이고, 그림 10(d)는 둘 다 적용한 것이다. 특이한 점은 Mixup을 적용할 경우 정확도는 향상되지만 학습 시 loss 값은 높아진다는 것으로, Loss 값이 0.26정도로 비교적 높지만 분류의 정확도는 높다는 특성을 보인다. 또한 Random Erase를 적용한 경우, 학습의 안정화가 빨리 이루어진다는 점을 확인할 수 있다. 그림 10(b), (d)에서 “Validation Accuracy” 그래프를 비교해보면 그림 10(d)의 그래프가 더욱 빠르게 편차가 적은 값으로 수렴되어 간다는 것을 확인할 수 있다. 이러한 결과를 통해 Mixup과 Random Erase가 K-MNIST의 분류 문제를 잘 해결하기 위한 학습방법이라는 것을 확인할 수 있다.

표 5에서는 본 논문에서 제안한 방법에 의한 결과와 기존의 논문에서 나타낸 결과를 비교하였다. 기존 방법에 의한 결과는 [1]번 논문의 연구팀이 추가 연구를 수행하여 웹 상에 결과를 게시한 것으로부터 가져왔다. 표 5에서 볼 수 있듯이 MNIST에 대해서는 99.75%, K-MNIST에 대해서는 99.07%, 그리고 K-49에 대해서는 97.56%의 정확도를 보임으로써 가장 높은 성능을 나타냈다.

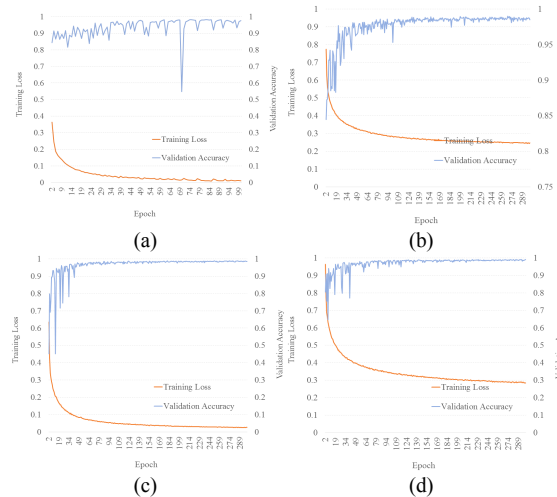


Fig. 10 Training process graph by using (a) default mode, (b) Mixup only, (c) Random Erase only, and (d) both of Mixup and Random Erase

Table. 5 Performance comparison by model and training technique [1][26]

Model	MNIST	K-MNIST	K-49
4-Nearest Neighbour Baseline[22]	97.14%	91.56%	86.01%
Tuned SVM (RBF kernel)[23]	98.57%	92.82%	
Keras Simple CNN Benchmark [24]	99.06%	95.12%	89.25%
PreActResNet-18[25]	99.56%	97.82%	96.64%
PreActResNet-18 + Input Mixup [19]	99.54%	98.41%	97.04%
PreActResNet-18 + Manifold Mixup[20]	99.54%	98.83%	97.33%
ResNet18 + VGG Ensemble	99.60%	98.90%	
Densenet-121(73) + Mixup + Random Erase (Proposed)	99.75%	99.07%	97.56%

V. 결론

본 논문에서는 Kuzushiji-MNIST와 Kuzushiji-49 데이터셋의 분류를 위한 학습 방법을 제안하였다. 실험을 통해서 CNN으로는 DenseNet이 가장 좋은 성능을 보였고, 300회 이상의 학습을 진행할 경우에 높은 정확도를 보였다. 여러 학습 기법들 중에서 Mixup과 Random Erase 조합이 좋은 학습 결과를 보인다는 것을 실험적으로 도출하여 Kuzushiji-MNIST와 Kuzushiji-49를 위한 학습 방법을 제안하였다. 이와 같은 딥러닝 기반의 기술을 통해 동아시아와 서양의 역사, 문화, 그리고 문화를 연구하는 다양한 연구자들에게 좋은 연구 기반을 제공할 것으로 사료된다. 아울러 본 연구에서와 같은 첨단 기술을 수단으로 다양한 학제간 소통과 융합에 일조하기를 기대한다.

ACKNOWLEDGEMENT

This research was supported in 2019 by The Ministry of Education, Republic of Korea, under the Program of university for innovation and Transformation.

References

- [1] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. "Deep Learning for Classical Japanese Literature," arXiv preprint arXiv:1812.01718v1, 2018.
- [2] Y. Hashimoto, Y. Iikura, Y. Hisada, S. Kang, T. Arisawa, and D. Kobayashi-Better. (2017, November). The Kuzushiji Project: Developing a Mobile Learning Application for Reading Early Modern Japanese Texts. *DHQ: Digital Humanities Quarterly* [Internet]. 11(1), pp. 1-13. Available: <http://dh2016.adho.org/static/data/254.html>.
- [3] K. Takashiro. (2013, March). Notation of the Japanese Syllabary seen in the Textbook of the Meiji first Year. *The bulletin of Jissen Women's Junior College* [Internet]. pp. 34:109 - 119. Available: <https://ci.nii.ac.jp/els/contents/110009587135.pdf?id=ART0010042265>.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097-1105, Jan. 2012.
- [5] K. Simonyan, and A. Zisserman. "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [6] M. Lin, Q. Chen, and S. Yan. "Network in network," arXiv preprint arXiv:1312.4400, 2013.
- [7] L. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation," arXiv preprint arXiv:1706.05587, 2017.
- [8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 2818-2826, 2016.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 770-778, 2016.
- [10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 2261-2269, 2017.
- [11] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. "Learning transferable architectures for scalable image recognition," arXiv preprint arXiv:1707.07012, 2017.
- [12] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 558-567, 2019.
- [13] C. for Open Data in the Humanities. Kuzushiji dataset [Internet]. Available: <http://codh.rois.ac.jp/char-shape/>.
- [14] Y. LeCun. The MNIST database of handwritten digits [Internet]. Available: <http://yann.lecun.com/exdb/mnist/>.
- [15] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.
- [16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
- [17] H.-T. Zheng, N. Ma, X. Zhang, and J. Sun. "Shufflenet v2: Practical guidelines for efficient cnn architecture design," arXiv preprint arXiv:1807.11164, 2018.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 770-778, 2016.
- [19] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. "mixup: Beyond Empirical Risk Minimization," arXiv preprint arXiv:1710.09412v2, 2018.
- [20] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. "Random Erasing Data Augmentation," arXiv preprint arXiv: 1708.04896v2, 2017.
- [21] V. Verma, A. Lamb, C. Beckham, A. Najafi, A. Courville, I. Mitliagkas, and Y. Bengio. "Manifold Mixup: Learning Better Representations by Interpolating Hidden States," arXiv preprint arXiv:1806.05236, 2018.
- [22] S. Bubeck, and U. V. Luxburg. "Nearest Neighbor Clustering: A Baseline Method for Consistent Clustering with Arbitrary Objective Functions," *Journal of Machine Learning Research*, vol. 10, pp. 657-698, Mar. 2009.
- [23] C. Chang, S. Chou. (2015, June). Tuning of the hyperparameters for L2-loss SVMs with the RBF kernel by the maximum-margin principle and the jackknife technique. *Pattern Recognition* [Internet]. 48(12), pp. 3983-3992. Available: <https://doi.org/10.1016/j.patcog.2015.06.017>.
- [24] ROIS-DS Center for Open Data in the Humanities. Keras Simple CNN Benchmark [Internet]. Available: https://github.com/rois-codh/kmnist/blob/master/benchmarks/kuzushiji_mnist_cnn.py.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision, Springer*, vol. 9, no. 4, pp. 630 - 645, 2016.
- [26] ROIS-DS Center for Open Data in the Humanities. Benchmarks & Results [Internet]. Available: <https://github.com/rois-codh/kmnist>.



박병서(Byung-Seo Park)

2019년 2월 : 광운대학교 경영학과 졸업(경영학사)
2019년 3월 ~ 현재 : 광운대학교 전자재료공학과(석박통합과정)
※ 관심분야 : 딥러닝, 2D/3D 영상 처리



이승영(Sungyoung Lee)

1996년 2월 경북대학교 일어일문학과 졸업(문학사)
1999년 2월 경북대학교 교육대학원 일어교육(교육학석사)
2005년 3월 국립춘추바대학 문예언어연구과(언어학 박사)
2007년 9월 ~ 2008년 2월 숭실대 조교수
2008년 3월 ~ 현재 : 광운대학교 교수
※ 전문영역 : 응용언어학, 일본어사



서영호(Young-Ho Seo)

1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
2005년 9월 ~ 2008년 2월 : 한성대학교 조교수
2008년 3월 ~ 현재 : 광운대학교 전자재료공학과 교수
※ 관심분야 : 실감미디어, 2D/3D 영상 신호처리, 디지털 홀로그램