도메인 어댑테이션을 이용한 폰트 변화에 강인한 한글 분류기 개발

박재우, 이은지, 조남익
서울대학교 전기정보공학부 뉴미디어통신연구소(INMC)
bjw0611@snu.ac.kr, jane0119@snu.ac.kr, *nicho@snu.ac.kr

Jaewoo Park, Eunji Lee, *Nam Ik Cho
Seoul National University Electrical Computer Engineering, INMC

요 약

본 논문에서는 도메인 어댑테이션을 이용하여 폰트 변화에 강인한 한글 분류기를 학습하는 방법을 제안한다. 제안하는 네트워크 모델은 총 7 개로 이루어져 있으며 각각 이미지로부터 폰트에 무관한 정보를 추출하는 인코더, 추출된 정보의 유효성을 판단하기 위해 이미지 재합성에 사용되는 디코더, 재합성된 이미지의 글자 분류기, 폰트 분류기, 재합성된 글자의 정교함을 판단하는 판별기(discriminator), 그리고 인코더에서 추출된 정보에 대한 글자 분류기, 폰트 분류기이다. 본 논문에서는 적대적 생성 신경망의 학습법을 따르는 도메인 어댑테이션 기법을 이용하여 인코더의 추출 정보가 폰트 정보는 속이면서 글자 분류의 정확성은 높이도록 학습하였다. 학습 결과 인코더로부터 추출되는 정보들은 폰트에 무관한 성질을 지니면서 글자 분류에 높은 정확성을 띄었으며, 추가로 디코더에서 나오는 이미지들도 원본 폰트와 같은 이미지를 생성해 낼 수 있었다.

1. 서론

글자 이미지는 많은 정보량을 지니고 있기 때문에 이미지에서 글자를 인식하려는 시도는 컴퓨터 비전 분야에서 오래전부터 지속되어 오고 있다. 기존에는 글자를 인식하기 위해 이미지의 선분이나, 기울어진 각도의 정보를 이용하여 인식하였으나[1][2][3][4], 최근 딥러닝이 다양한 분야에서 뛰어난 성능을 보여줌에 따라 글자 인식에도 딥러닝 방식들이 사용되고 있다[5][6]. 글자 이미지를 인식하는 문제는 폰트를 이용하여 생성된 디지털 이미지 인식, 사람이 쓴 손 글씨 이미지 인식, 그리고 다양한 디자인이 들어간 간판 이미지 인식으로 크게 3 가지 종류로 나누어진다. 이 중 폰트를 이용하여 생성된 디지털 문자 이미지는 이미지 형태로 되어 있는 다양한 문서로부터 정보를 추출하여 데이터화 하는 작업에 사용될 수 있어 최근 많은 관심을 받고 있다.

디지털 글자는 사용되는 폰트의 형식에 따라 글자가 다르게 생성되고, 최근 많은 폰트들이 제시되고 사용됨에 따라 Figure 1 과 같이 글자 이미지가 지닐 수 있는 형태가 다양하게 변화하고 있다. 또한, 문서에서 글자 이미지의 크기도 다양하게 사용되고 이에 의한 블러 현상이 발생하므로 이러한 글자 내용 자체와는 무관한 이미지의 변화를 걸러내고 인식할 수 있는 강인한 분류기의 필요성이 대두되고 있다. 기존에 제시된 딥러닝을 이용한 글씨 분류기는 글씨 이미지의 다양한 폰트에서 강인한 방식으로의 학습보다는 다양한 폰트의 글자 이미지를 학습 데이터셋에 구성함으로써 성능을 개선하였다. 그러나, 최근 사용되는 폰트 및 글자 크기가 다양하여 해당 방식으로는 학습셋에 포함되지 않은 글자가 들어오는 경우 낮은 성능을 보였으며 모든 폰트 문자에 대한 학습셋을 구성하는 것은 매우 비용이 커서 문제가 발생하고 있다.

위와 같은 문제를 해결하기 위해, 최근 도메인 어댑테이션 방식을 이용하여 핵심 정보를 추출하는 연구들이 제안되어 많은 관심을 받고 있다[7][8]. 도메인 어댑테이션을 이용하여 다양한 상황에 대해 강인한 분류기를 학습하는 방법 중 최근 많이 제시되는 방법은 적대적 생성 신경망[9]의 방식을 이용하여, 분류기에서 사용되는 벡터가 다양한 상황으로부터

Figure 1. Various types of font images

추출되어도 클래스가 같다면 동일하게 나오도록 학습하는 것을 목표로 한다. 이에 따라, 본 논문에서는 도메인 어댑테이션을 이용하여 폰트 및 크기 변화에 강인한 한글 분류기를 학습하는 것을 목표로 한다.

기존의 대부분의 연구들이 숫자 및 영어에 관련된 도메인 어댑테이션을 학습하여 성공을 거두었으나, 숫자는 클래스 10 개, 영어는 대 소문자를 구분하여도 클래스 52 개임에 반해, 한글은 실용적으로 사용되는 글자의 클래스만 해도 2450 개에 달해 더욱 어려운 문제로 다루어지고 있다. 본 논문에서는 이를 초성, 중성, 종성으로 나누어 인식해 정확도를 높이는 방식을 따른다. 초성은 총 19 개의 클래스, 중성은 총 21 개의 클래스, 그리고 종성은 총 28 개의 클래스를 이용한다.

실험 결과 기존의 분류기에 비해, 학습 때 사용되지 않은 폰트 및 크기의 글자 이미지에 대해 정확한 성능을 보였으며, 추가적으로 학습과정에서 원하는 폰트의 글자 이미지를 생성할 수 있는 생성모델을 얻을 수 있었다.

2. 비지도 도메인 어댑테이션

비지도 도메인 어댑테이션(Unsupervised 도메인 Adaptation)이란 레이블이 있는 소스 도메인의 데이터와 레이블이 없는 타겟 도메인의 데이터를 학습하여 소스 도메인 데이터 분류의 성능도 유지하면서, 타겟 도메인 데이타의 분류 성능을 높이는 것을 목표로 한다. 학습 전에는 타겟 도메인 데이터로부터 추출한 피쳐와 소스 도메인으로부터 추출한

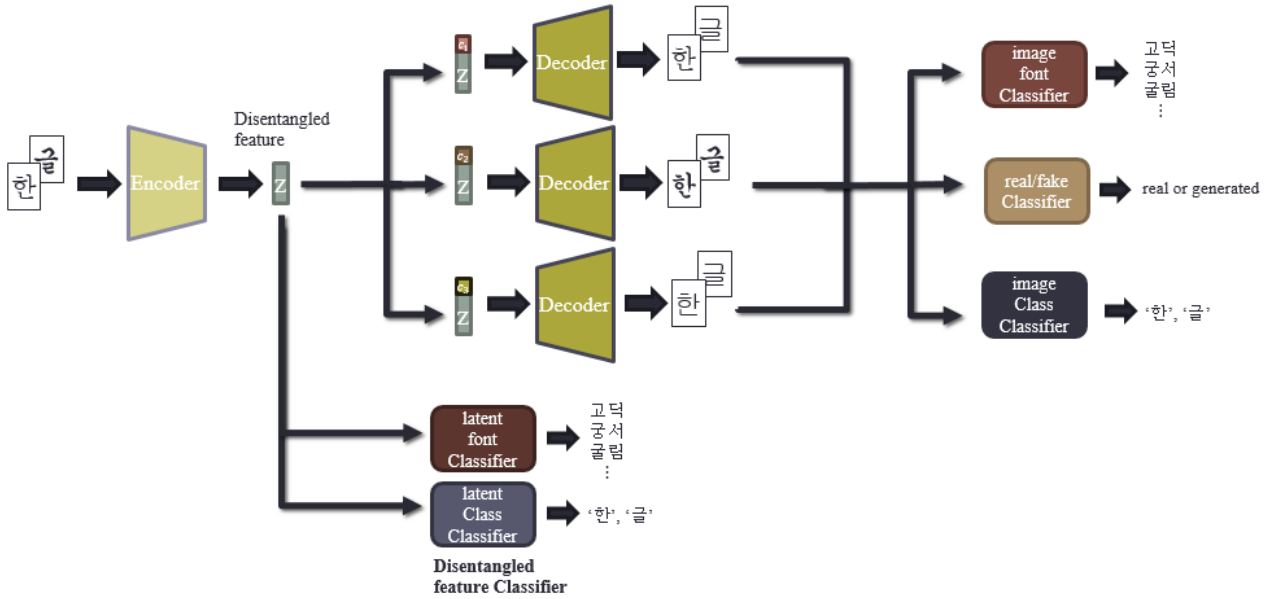

Figure 2. Proposed network model with domain adaptation

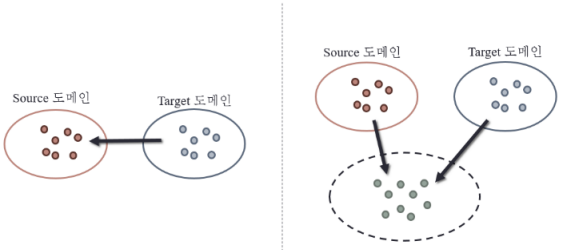

Figure 3. Domain adaptation methods

피쳐는 서로 다른 분포를 지니고 있으며 이로 인해 소스 도메인에 대해 학습한 분류기의 성능이 타겟 도메인에서 크게 떨어지게 되는데, 이를 도메인 시프트 현상이라고 한다. 따라서, 대부분의 도메인 어댑테이션은 도메인 시프트 현상을 낮추는 방향으로 학습이 진행된다.

도메인 시프트를 방지하는 방법으로는 두 가지 유형이 있는데, 타겟 도메인 데이터로부터 추출한 피쳐의 분포를 소스 도메인으로부터 추출한 피쳐 분포와 같이 만드는 방식과 소스 도메인의 특성과 타겟 도메인의 특성 중 서로 공통되는 피쳐만 추출하는 방식이 있다 (Figure 3). 앞의 방식의 경우 미리 소스 도메인에 대하여 높은 성능을 지니는 분류기를 학습해 두고, 해당 분류기에 타겟 도메인의 데이터를 넣어 타겟 도메인으로부터 나오는 피쳐의 분포를 소스 도메인에서와 동일하게 맞추는 방식으로 진행된다. 후자의 경우, 소스 도메인과 타겟 도메인의 데이터를 동시에 사용하여 학습하며, 이 때, 소스 도메인으로부터 추출된 것 인지, 타겟 도메인으로부터 추출된 것 인지 구분할 수 없는 피쳐 (disentangled feature)를 추출하는 것을 목표로 학습한다.

이를 위해, 두 분포의 거리를 줄이기 위한 방법으로 adversarial learning 방식을 이용하는 논문들[7][8], Maximum Mean Discrepancy(MMD)를 이용하는 논문들이 최근 제안되었으며[10], 좋은 성능으로 인해 많은 관심을 받고 있다. 또한, 분류 문제 외에도 이미지 segmentation [11], detection[12] 분야에서도 레이블이 없어서 학습이 힘든 경우 사용되어 큰 성과를 거두고 있다.

본 논문에서는 위의 두 가지 방식 중 disentangled 피쳐를 추출하는 방식을 따르며, 이 때, 소스 도메인은 레이블을 가지고 있는 폰트 이미지, 타겟 도메인은 레이블을 가지고 있지 않은 폰트 이미지로 설정하였다. 또한, 두 분포의 거리를 줄이는 방법으로는 adversarial learning 방식을 이용하였다.

3. 실험 방법

실험에 사용된 모델의 전체적인 구조는[7]을 기반으로 하며 이는 Figure 2 와 같다. 한글 이미지로부터 소스, 타겟 특성 없이 글자의 클래스 정보만 가지고 있는 disentangled 피쳐 $\mathbf{z}$ 를 추출하는 인코더와 해당 피쳐로부터 글자 클래스를 분류하는 글자 분류기(latent 클래스 classifier), 폰트를 구분하는 폰트 분류기(latent font classifier)가 있다.

이 후, $\mathbf{z}$가 실제로 글자의 핵심 정보를 폰트를 제외하고 추출했는지 여부를 파악하기 위해 $\mathbf{z}$를 이용하여 글자 이미지를 재생성하는 디코더가 사용되는데, 이 때, $\mathbf{z}$와 함께 폰트 생성 조건 $\mathbf{c}_i$를 입력하여 디코더가 $\mathbf{z}$ 로부터 어느 폰트에 해당하는 이미지를 생성할 지 구분할 수 있도록 한다. 이는 폰트 정보와 무관한 정보만 $\mathbf{z}$에 포함되도록 하기 위함이며, 위의 그림에서 디코더는 모두 동일한 파라미터를 공유하는 모델이다. 이 후, 디코더가 생성한 이미지의 유효성을 판단하기 위해, 재생성한 이미지가 폰트 정보를 잘 담고 있는지 파악하는 이미지 폰트 분류기 (image font classifier), 재생성한 이미지가 클래스 정보를 제대로 담고 있는지 파악하는(image 클래스 classifier), 그리고 재생성된 이미지의 정교함을 판단하는 이미지 discriminator 로 구성되어 있다. 이 때, 사용되는 인코더와 디코더는 Variational Auto 인코더(VAE)[13] 방식을 사용하였다.

학습에 사용되는 loss 함수 및 학습 방법은 아래와 같다. 먼저 인코더와 디코더는 VAE 에 속하는 모듈로, 해당 loss 를 사용하여, 원본 이미지$\mathbf{x}$와 동일한 폰트로 재생성된 이미지$\hat{\mathbf{x}}$간의 거리를 L_2 거리로 사용하며, Kullback-Leibler divergence 를 통해$\mathbf{x}$로부터 추출한 $\mathbf{z}$가 항상 Gaussian 분포 p(z)를 따르도록

한다.

$$L_{vae} = \|x - \hat{x}\|_2^2 + KL(q(z|x)||p(z))$$

식 1. VAE 목적함수

z 폰트 분류기는 인코더가 폰트에 무관한 정보를 추출하도록 하기 위해 사용되므로, **z** 폰트 분류기와 인코더는 adversarial learning 의 관계를 지니고 학습된다. 이 때, adversarial learning 에서의 real/fake 의 구분 대신 폰트 정보를 구분하도록 학습한다.

$$L_{z\ font}^{adv} = E[\log(D_z(\mathbf{z_{source}}))] + E\left[\log\left(1 - D_z(\mathbf{z_{target}})\right)\right]$$

식 2. **z** 폰트 분류기와 인코더의 adversarial learning loss

또한, 인코더에서 글자의 클래스 정보는 유지되며 **z** 가 추출되어야 하므로 **z** 클래스 분류기와 인코더는 cross-entropy loss 를 이용하여 latent 분류기의 정확도가 높아지는 방향으로 학습된다.

$$L_{z\ 클래스} = E[\log(P_{z\ classifier}(\mathbf{z}))]$$

식 3. **z** 클래스 분류기 loss 함수

재생성된 이미지 $\hat{x}$의 유효성을 판별하기 위한 모듈 중 재생성된 이미지 폰트 분류기 및 재생성된 이미지 클래스 분류기는 위의 **z** 에 사용되는 방식과 동일하며, 이때는 인코더 대신 디코더가 adversarial learning 의 generator 역할을 하게 된다. Loss 함수는 아래와 같다.

$$L_{\hat{x}\ font}^{adv} = E[\log(D_{\hat{x}}(\hat{\mathbf{x}}_{source}))] + E\left[\log\left(1 - D_{\hat{x}}(\hat{\mathbf{x}}_{\mathbf{target}})\right)\right]$$

식 4. $\hat{\mathbf{x}}$ 폰트 분류기와 디코더의 adversarial learning loss

$$L_{\hat{x}\ class} = E[\log(P_{\hat{x}\ classifier}(\hat{\mathbf{x}}))]$$

식 5. $\hat{x}$ 클래스 분류기 loss 함수

마지막으로, 재생성된 이미지의 정교함을 추가하기 위해, VAE(인코더, 디코더)모델 전체를 generator 로 보고 입력이 학습하는 원본 데이터 이미지인지, VAE 로부터 생성된 이미지인지 학습하는 adversarial learning loss 가 사용된다.

$$L_{\hat{x}}^{adv} = E[\log(D_x(\mathbf{x}))] + E[\log(1 - D_{\hat{x}}(\hat{\mathbf{x}}))]$$

식 6. $\hat{\mathbf{x}}$의 adversarial learning loss

4. 실험 결과 및 분석

학습에 사용한 데이터는 소스 도메인으로 굴림체, 바탕체를 사용하였으며, 타겟 도메인으로 돋움체를 사용하였다. 각각의 폰트별로 글자를 합성하여 사용하였으며, 소스 도메인의 데이터는 총 4900 자, 타겟 도메인의 데이터는 총 2450 자의 글씨를 데이터로 사용하여 학습하였다.

도메인 어댑테이션의 효과를 확인하기 위하여, Figure 2 에서의 인코더와 latent 클래스 classifier 에 사용되는 cross-entropy (식 3)를 제외한 loss 는 모두 무시하는 기존 분류기를 학습하여 도메인 어댑테이션을 따랐을 시 성능의 변화를 측정하였다.

정확도	기존 분류기	도메인 어댑테이션
소스 도메인 (굴림체, 바탕체)	99.21%	**99.97%**
타겟 도메인 (돋움체)	85.15%	**98.21%**

Table 1. Accuracy comparison of domain adaptation

실험결과 도메인 어댑테이션을 사용한 경우와 기존 분류기의 성능이 레이블을 가지고 있는 소스 도메인의 데이터에서는 0.75%의 차이만 나타났으나, 타겟 도메인의 데이터에서는 13.06%의 큰 차이를 나타내는 것을 확인 할 수 있었다. 이를 통해, 도메인 어댑테이션 방식의 학습이 도메인 시프트를 줄이는 역할을 하고 있음을 확인할 수 있었다. 또한, 도메인 어댑테이션을 사용하는 경우 소스 도메인에서도 기존 분류기보다 성능이 높음을 확인할 수 있었는데, 이는, 도메인 어댑테이션 방식의 학습이 글자에서 선택적으로 중요한 정보만을 추출하는데 도움이 되기 때문이라고 할 수 있다.

또한, 위의 도메인 어댑테이션을 사용하여 학습하는 과정에서 이미지 재생성 과정이 포함되는데, 재생성된 이미지의 결과는 Figure 4 와 같다.

Figure 4. source image, target image, reconstruction image and domain translated image

Figure 4 에서 점선의 왼쪽 이미지는 가장 왼쪽부터 소스 이미지, 소스 → 소스 재생성 이미지, 소스 → 타겟 재생성 이미지이며, 점선의 오른쪽의 가장 왼쪽부터 타겟 이미지, 타겟 → 타겟 재생성 이미지, 타겟 → 소스 재생성 이미지이다. 실험결과 글자의 클래스는 유지하면서 폰트만 변경하는 성공적인 도메인 translation 이 이루어지고 있음을 확인 할 수 있었다.

5. 결론

본 논문에서는 도메인 어댑테이션 방식을 이용하여 레이블이 없는 폰트의 한글 글자의 분류기를 학습하였다. 학습결과 기존 분류기 방식에 비해 유의미한 성능의 향상을 확인할 수 있었으며, 재생성된 이미지의 정교함을 통해, 제안한 모델이 글자 이미지로부터 폰트 정보를 제외한 유의미한 정보들을 추출할 수 있음을 보였다. 본 논문의 실험에서는 합성 글자 이미지에 대해 폰트 정보에 따라 소스와 타겟 이미지를 구분하였으나, 향후 scan 된 글씨 이미지를 타겟 이미지로

사용하고 합성 글씨를 소스 도메인으로 사용하여 취득이 힘든 scan 글씨 이미지에 대해 성능을 높이는 방식의 연구를 진행하면, 레이블 취득 비용을 절감할 수 있을 것으로 보인다.

감사의 글

본 연구는 ㈜한글과컴퓨터의 지원 및 경찰청과 치안과학기술연구개발사업단의 치안과학기술연구개발사업(PA-C000001)의 지원을 받아 이루어진 것입니다.

참고 문헌

[1] 최낙승, “Hough 변환을 이용한 필기체 한글의 인식에 관한 연구”, 건국대학교 석사학위논문, 1989

[2] 정민철, “오프라인 필기체 한글 인식을 위한 자소 내 자획의 분리”, 한국산학기술학회논문지 Vol.7, No.3, pp. 385-392, 2006.

[3] 김태균, 이병희, “한국어 정보처리 : 한글 문자 인식에서의 오인식 문자 교정을 위한 단어 학습과 오류 형태에 관한 연구”, 정보처리학회논문지, Vol. 3, No. 5, 1273-1280, 1996.

[4] Natarajan et al. “Multilingual Machine Printed OCR,” International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 01, pp. 43-63, 2001

[5] Kartik Dutta, Praveen Krishnan, Minesh Mathew, C.V. Jawahar. “Improving CNN-RNN Hybrid Networks for Handwriting Recognition,” ICFHR, 2018.

[6] Thomas M. Breuel, “High Performance Text Recognition Using a Hybrid Convolutional-LSTM Implementation”, IAPR, 2017.

[7] Liu et al, “A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation” NIPS, 2018.

[8] Pedro O. Pinheiro, “Unsupervised Domain Adaptation with Similarity Learning”, CVPR, 2018.

[9] Goodfellow et al. “GenerativeAdversarial Nets,” NIPS, 2014.

[10] M Long et al, “Learning Transferable Features with Deep Adaptation”, ICML, 2015.

[11] Z Murez et al, “Image to Image Translation for Domain Adaptation”, CVPR, 2018.

[12] Q cai et al, “Exploring Object Relation in Mean Teacher for Cross-Domain Detection,” CVPR, 2019.

[13] Diederik P Kingma, Max Welling, “Auto-Encoding Variational Bayes,” ICLR, 2014.