**Domain mixup을 이용한 unsupervised domain adaptation**

**요 약**

최근 이미지 인식(image recognition)을 포함한 다양한 딥러닝 기능의 인공지능 활용이 이루어지고 있는데 그 중 손글씨를 실제 텍스트로 바꿔주는 필기 인식의 많은 연구도 이루어지고 있다. 이 때 기존에 많이 사용되는 MNIST 데이터는 검정 배경과 하얀 글씨 또는 하얀 배경에 검정 글씨로 이루어져 있는데 실제로 사람이 쓰는 글자들의 데이터는 색과 배경이 다양하기 때문에 새로운 데이터들(실생활에서 나오는 데이터)에 대한 인식률이 낮아서 Domain Adaptation의 요구가 생기게 되었다. 우리는 이미 존재하는 MNIST, USPS, MNIST\_M 데이터를 활용하면서 여러 색이 섞여있거나 형태가 다양한 새로운 데이터이면서 라벨이 없는 데이터들을 domain mixup을 활용해 잘 인식하고는 것을 목표로 한다.

**1. 서론**

**1.1 연구배경**

현재 실제 손글씨를 문서의 텍스트로 변환시키는 필기 인식에 대한 많은 연구가 이루어지고 있다. 이 때 MNIST dataset을 주로 이용해서 학습을 시키는데, MNIST dataset이란 손으로 쓰여진 28x28픽셀 고정 크기 이미지로 된 dataset로 흰색 배경에 검은 글씨 또는 검은색 배경에 흰색 글씨로 이루어져있다. 하지만 사람이 실제로 글씨를 쓰는 환경과 도구는 다양하기 때문에 MNIST dataset 으로는 다양한 데이터들의 인식률이 높게 나올 수 없다. 예를 들면 사람이 쓰는 손글씨와 배경은 흰색과 검은색 이외에도 다양할 수 있고 줄 있는 공책에 쓰게 되면 단순한 배경에서 벗어나게 되므로 기존의 dataset으로는 예측이 힘들게 된다.

따라서 여러 환경에서 쓰여진 dataset을 이용해 새로운 모델을 만들어야 되지만 새로운 이미지들을 구하기 힘들 뿐만 아니라 구하더라도 이미지들에 라벨을 일일히 붙여야하기 때문에 시간적 비용이 많이 들어가는 문제가 있다.

이 문제를 해결하기 위해 나온 이론이 Domain Adaptation이다. Domain Adaptation이란 기존에 모델이 동작하던 영역을 소스(source) 도메인이라고 하고, 새로운 영역을 타겟(target) 도메인이라고 하는데 이 새로운 타겟 도메인에 기존의 소스 도메인의 정보를 적응(Adaptation) 시켜서 사용하고자 하는 것이다. Domain adaptation을 사용하면 기존의 dataset으로도 다양한 데이터들의 인식률을 높일 수 있을 것이다. 따라서 본 연구에서는 domain adaptation과 다양한 증강방법론 등을 통해 라벨이 없는 새로운 데이터들을 얼마나 잘 인식하는지 보이고 수치를 나타내고자 한다.

**1.2 연구목표**

이번 연구의 목표는 MNIST, USPS, MNIST\_M 데이터로 학습된 모델을 이용해 라벨이 없는 새로운 데이터들을 domain adaption과 이미지 증강을 통해 학습된 모델이 실제로 사람이 쓰는 다양한 상황과 환경에서의 손 글씨에 대한 높은 인식률을 보여주는 것이다.

**2. 관련 연구**

**2.1 전이 학습(Transfer Learning)**

일반적으로 딥러닝 모델 학습에는 많은 양의 데이터가 필요하다. 하지만 일반적으로 충분히 큰 학습 데이터를 만드는 것은 쉬운 일이 아니다. 이런 문제를 해결하기 위해 나온 것이 전이 학습이다. 전이 학습은 이미지넷 같이 큰 학습 데이터 가지고 학습시킨 모델의 가중치를 가져와 모델의 classifier 부분만 바꾸는 등 우리가 가지고 있는 데이터에 맞추어 모델을 학습시키는 방법이다. 하지만 전이 학습에는 데이터 라벨 정보도 필요하기 때문에 라벨이 없는 데이터에는 사용할 수 없다.

**2.2 Domain Adaptation**

Source 데이터와 target 데이터의 domain 차이가 많이 나면 source 데이터로 학습된 모델이 source 데이터에 대해서 예측을 잘할 수 있을 지라도 target 데이터에 대해서는 예측을 잘하지 못할 수 있다. 이를 해결하기 위해 모델을 source 데이터와 target 데이터의 분포를 유사하게 학습시키려 하고 이를 domain adaptation이라 한다.

Domain adaptation에는 여러 방법론이 있는데 대표적으로 domain classifier 부분에gradient reversed layer을 사용해 source와 target domain을 구분 못하게 하는 Unsupervised Domain Adaptation by Backpropagation (2015-Yaroslav Ganin) 방식과 source 데이터로 학습된 CNN feature extractor 부분을 활용해 target 데이터로 새로운 모델을 만드는 Adversarial Discriminative Domain Adaptation (2017- Eric Tzeng) 방식이 있다. 이 방법을 이용하면 source와 target domain 차가 큰 경우에도 잘 예측하는 모델을 만들 수 있을 것이다.

**2.3 데이터 증강(Data Augment)**

데이터 증강 기법은 데이터의 양이 부족할 때 사용되는 방법론이다. 데이터 증강을 이용해 데이터를 적절히 변형시켜 모델을 학습하면 일반화에 강한 모델이 만들어진다. 데이터 증강은 모델 훈련에 효과적이라 대다수의 모델 학습에 이 기법이 쓰인다. 이 방법을 이용해 source 데이터와 target 데이터의 domain을 비슷하게 맞춰주면 source와 target domain의 차가 크더라도 모델이 잘 예측할 것이다.

**3. 프로젝트 내용**

**3.1 학습 시나리오**

1. MNIST, USPS, MNIST-M, Custom 데이터셋을 준비한다.
2. 여러 실험을 설계한다.

Source -> Target

* 1. MNIST -> USPS
  2. MNIST -> MNIST-M
  3. MNIST -> Custom
  4. USPS -> MNIST
  5. USPS -> Custom
  6. MNIST-M -> Custom

1. Source와 target간의 도메인 차를 줄이기 위해 Domain mix up을 사용해 CNN feature extractor를 학습시킨다.
2. ADDA 기법을 사용해 모델의 정확도를 높인다.

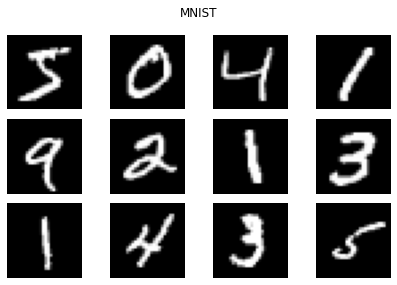
**3.2 요구사항**

**3.2.1 Domain Mix up**

서로 다른 두 도메인을 비슷하게 만들기 위해서는 각 데이터 셋에 대한 특성을 파악하고 데이터 마다 도메인 정보인 배경과 글자를 분리해야 된다. Source 데이터 셋과 target 데이터 셋에 대해 배경과 글자를 분리한 다음 source데이터셋을 target 데이터셋처럼 보이게 하기 위해 target데이터셋의 배경 부분을 source 데이터 셋에 합성해야 한다.

**3.2.1.1 데이터 셋 특징**

데이터 셋에서 글자와 배경을 분리하기 위해선 앞서 각 데이터 셋의 특징을 알아야한다.



MNIST의 경우 배경이 검은색, 글자는 하얀색으로 되어 있고 글자가 USPS 데이터셋 보다 얇은 특징을 가지고 있다.

**<그림1>**

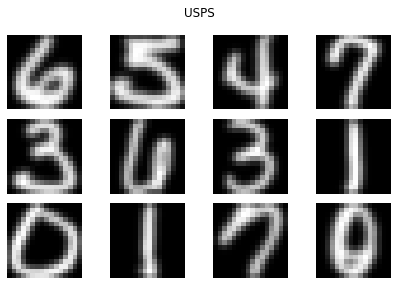


**그림**

MNIST-M의 경우 MNIST 데이터셋에 글자와 배경에 임의로 색을 입힌 특징을 가지고 있다.

**그림**

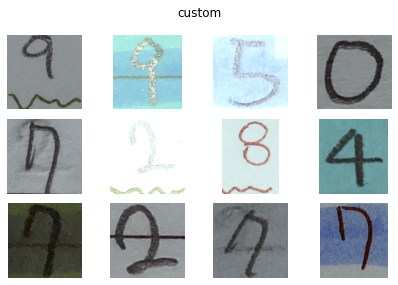
**<그림2>**



USPS의 경우 MNIST와 같이 배경이 검은색 글자는 하얀색으로 되어 있는데 MNIST에 비해 글자가 두껍고 글자와 배경 사이에 여백이 거의 없는 특징을 가지고 있다.

**그림 3**

**<그림3>**



Custom의 경우 필기구의 종류, 배경의 종류, 밝기의 종류, 글자를 쓴 형태에 따라 나뉜다.   
필기구는 검정 샤프, 파랑 볼펜으로 나뉜다.  
배경은 흰 배경, 선이 그어져 있는 배경, 노란 배경으로 나뉜다.  
밝기는 밝음, 빛 반사, 어두움으로 나뉜다.

글자를 쓴 형태는 밑줄, 글자를 X표시, 두 줄 번개 모양, 지우개, 형광펜으로 지운 형태로 나뉜다.

**그림 4**

**<그림4>**

**3.2.1.2 배경, 글자 분리**

배경과 글자를 분리하기 위해 이미지에서 글자 부분을 찾아야 된다. 처음 글자 영역을 찾기 위해

CRAFT: Character-Region Awareness For Text detection을 사용해 글자를 찾으려 했지만 전체 이미지 중 10%만 글자가 있다고 판단을 했고 10% 중 노이즈를 글자로 인식하는 경우가 대부분이라 text detect 방법으로는 배경과 글자를 분리하지 못했다.

MNIST와 USPS 데이터 셋 같은 경우 글자와 배경이 0,1로 이루어져 있어 단순 분리가 가능 했지만 MNIST-M과 Custom 데이터 셋의 경우 글자와 배경의 색이 매 이미지 마다 다르기 때문에 해당 데이터셋에서는 사용하지 못한다. 그렇기 때문에 한 이미지에서 글자와 배경을 분류해야 되기 때문에 유명한 분류 알고리즘인 k-means clustering을 사용해 이미지에서 글자와 배경을 분리한다. 색상 정보를 기준으로 k-means clustering을 사용해야 되기 때문에 색을 가장 직관적으로 표현하는 색상 모델인 HSV를 사용하였고 색상 정보만 필요하기 때문에 HUE 채널만 사용하였다. HSV로 바뀐 이미지를 1차원으로 flatten시킨 다음 글자와 배경 2개의 그룹으로 나누는 k-means 모델을 생성해 이미지에서 글자와 배경을 분리한다. 배경과 글자를 분리 후 어떤 그룹이 배경인지 글자인지 판단하기 위해 분리된 픽셀의 수가 많은 쪽을 배경으로 가정하고 배경 그룹과 글자 그룹에 라벨링을 해준다.

위 과정을 거치고 나면 <그림 5> 처럼 글자와 배경이 분리된다.   
**<그림5>**

하지만 배경에도 여러 색이 존재하기 때문에 2개의 그룹으로 나누었다 해도 항상 완벽하게 분리되진 않아 몇몇 배경에는 노이즈가 존재한다.  
글자와 배경을 분리한 후 배경을 사용하면 글자의 부분은 비어 있게 된다. 빈 공간을 채우기 위해 이미지 매 픽셀 마다 빈 공간(글자 부분)인지 확인하고 빈 공간이면 주변 빈 공간이 아닌 픽셀 값과 비슷하게 빈 공간을 채운다.

**그림 5**

**그림 6**

**<그림6>**

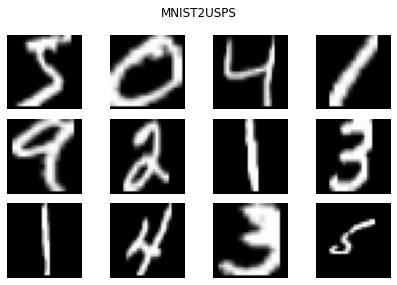
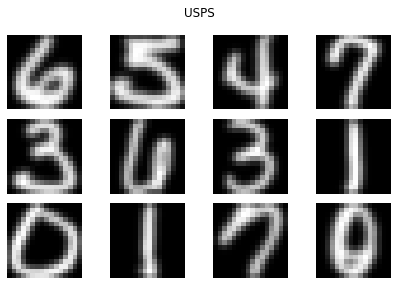
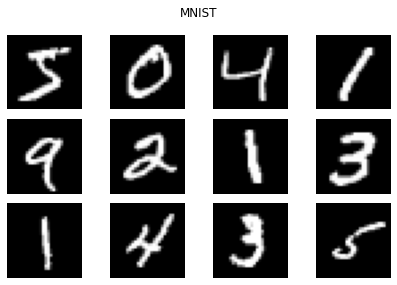
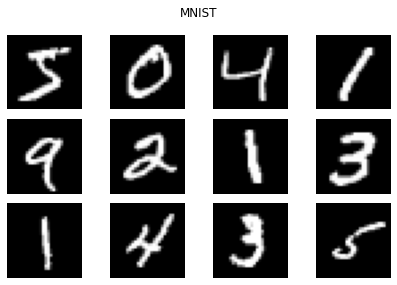
**3.2.1.3 배경, 글자 합성**

Source 데이터 셋을 target 데이터 셋처럼 보이게 하기 위해선 배경과 글자를 target 데이터 셋처럼 맞춰 주어야한다.   
앞서 구한 배경, 글자 분리 알고리즘을 이용해 source 데이터 셋의 배경 부분에 target 데이터 셋 배경을 합성한다. MNIST -> MNIST-M으로 할 때 글자 색은 임의의 색으로 바꾸었다. 합성한 결과는 <그림 7>과 같다.

MNIST와 USPS처럼 배경은 같은데 글자 부분이 다를 경우에는 배경은 그대로 두고 이미지의 글자 영역을 동일시시킨다.

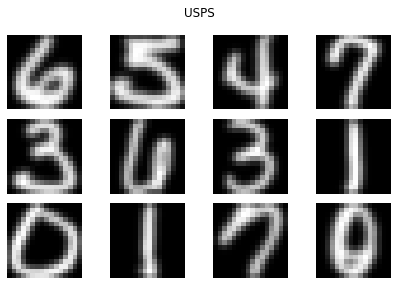
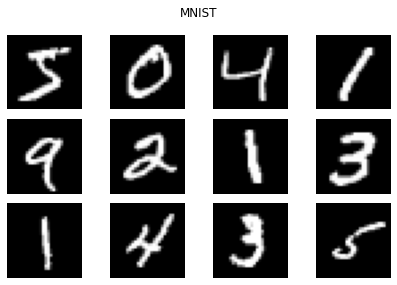
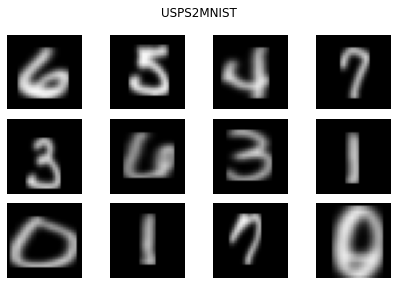
<그림 8>(MNIST -> USPS) <그림 9>(USPS -> MNIST)

**<그림7>**



**<그림8>**

MNIST, MNIST -> USPS, USPS



**그림 9**

USPS, USPS -> MNIST, MNIST

**<그림9>**

**3.2.2 학습 설계**

**3.2.1.1 학습 데이터 셋 준비**

Domain mix up 중 이미지에서 글자와 배경을 분리하기 위해서는 k-means clustering이 필요하다. 하지만 k-means clustering에는 한 이미지당 약1초씩 시간이 소요가 되었고 source 데이터 10,000장, target 데이터 50,000장에 대해 매 학습 마다 증강하는 데에는 많은 시간이 든다. 따라서 사전에 각 실험에 맞추어 학습 데이터 셋을 준비한다.

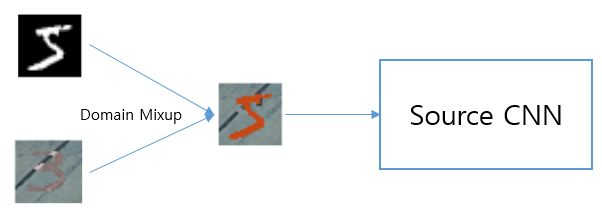
**3.2.1.2 성능 개선**

Domain mix up 같은 경우 domain adaptation 문제에서 유용하게 사용할 수 있는 증강이지 모델 학습 방법론은 아니다. 따라서 앞서 연구되었던 여러 학습 기법 중 ADDA를 사용해 모델의 성능을 더 향상시키려 한다. 실제 ADDA를 사용하니 domain mix up만을 사용했을 때 보다 점수가 오른 것을 확인할 수 있다. <표 1>

**3.2.1.3 학습**

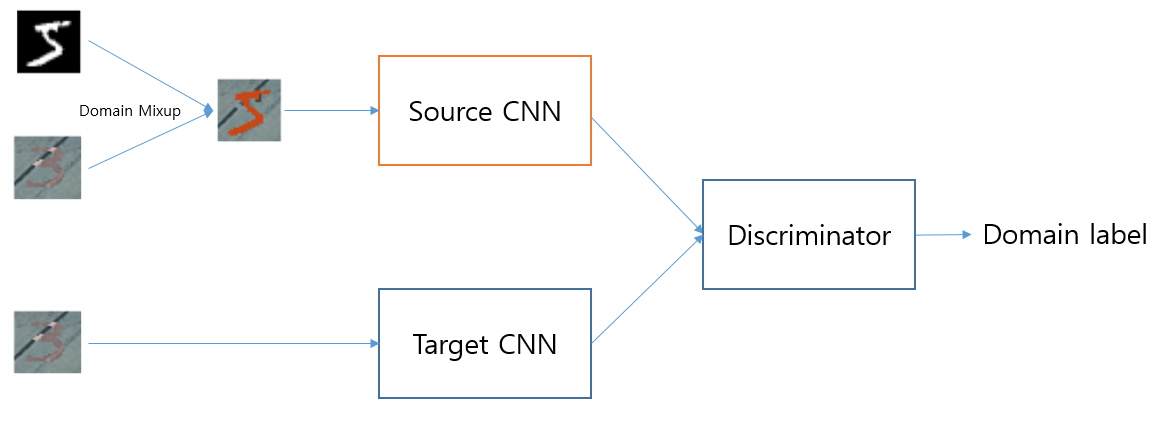
Domain mix up이 적용된 데이터 셋으로 일반적인 모델을 학습시키는 방법으로 학습을 시킨다.

Domain mix up로 학습된 모델을 가지고 ADDA 방식으로 target 데이터 셋을 위한 모델을 학습시킨다.



**<그림10>**

Domain mix up 만을 이용한 모델 학습



**<그림11>**

Domain mix up + ADDA

**3.3 실험**

어떤 데이터 셋으로 학습시켰는지에 대해선 앞서 학습 시나리오에 있는 실험들로 실험을 진행했다.

**3.3.1.1 학습 하이퍼파라미터**

학습에 사용한 모델은 CNN feature extractor와 classifier 부분으로 나누어진다. Feature extractor 부분은 Lenet을 사용했고 classfier 부분은 각 층에 500개의 hidden units을 가진 3개층 fully connected layers을 사용했고 ADDA의 Discriminator는 single fully connected layer를 사용하였다.   
 Source 데이터를 이용해 feature extractor를 학습시킬 때는 learning late: 2e-4, batch size: 128, optimizer: Adam, epoch: 20으로 고정을 하고 학습을 시켰고 ADDA에서 target 데이터를 위한 모델과 discriminator를 학습시킬 때에는 learning late: 1e-4, batch size: 128, optimizer: Adam, epoch: 50으로 고정하고 학습을 진행했다.

Source 데이터 셋은 min(10,000,source data length)장 target 데이터 셋은 min(50,000,target data length)

**3.3.2 custom dataset**

실험을 진행하기 위해 다양한 데이터 셋을 직접 만들었다. 이 때 사용한 필기구는 검정 샤프와 파란 볼펜이고, 배경을 달리하기 위해 무선 공책과 흰색, 노란색 유선 공책을 사용하였다. 필기 방식으로는 숫자만 쓴 기본 형태와 밑줄 친 형태, 형광펜을 그은 방식, 필기도구로 기존 숫자를 지우고 다시 쓴 방식 등으로 여러가지 방식을 취했다. 그리고 카테고리마다 밝은 환경, 어두운 환경, 카메라 플래시를 이용해 빛을 반사시킨 환경에서 각각 모두 찍어 총 1800장 가량의 사진을 만들고 더 다양한 데이터셋을 위해 blur, 색 온도,색상, 채도를 바꿔 약 10,000장의 데이터셋을 만들었다. 이 때 아이폰으로 찍은 사진이라 확장자가 .HEIC 이었기에 실험에 이용하기 위해서 모두 png파일로 변경해주는 작업을 진행했다.

실험을 진행하다보니 custom dataset으로 진행한 연구 결과의 성능이 매우 떨어지는 문제가 발생하였고 그 원인을 생각했을 때 기존 MNIST, USPS, MNIST-M데이터 셋에 비해 custom dataset의 배경이 차지하는 부분이 너무 크고 숫자의 크기가 너무 작았던 차이점이 존재했다. 따라서 최대한 기존 데이터 셋과 비슷하게 custom 해주어서 사진 자르기로 배경을 일부 지웠고 이로 인해 변경된 사진 크기를 256\*256 픽셀로 조정해주는 작업을 진행했다.

**4. 프로젝트 결과**

**4.1 연구 결과**

이번 연구에서 가장 중요한 점은 domain이 서로 다른 두 데이터셋 source, target 데이터셋에서 라벨이 있는 source 데이터셋을 이용해 라벨이 없는 target 데이터셋을 unsupervised learning하는 것이다. 이러한 방법을 domain adaptation이 있는데 기존에 연구되어진 여러 domain adaptation 기법 중 증강을 이용한 것이다. 사용한 증강은 domain mix up 방법으로 source 데이터셋의 domain에 target 데이터셋의 domain 정보를 집어넣는 방식이다.   
기존에 있던 domain adaptation 방법과 비교해 이번 연구의 성능을 살펴보면 다음 <표1>과 같다.

Domain mix up은 ADDA방법과는 달리 사전에 학습된 모델이 필요가 없는 것이 장점이다. 따로 source 데이터셋에 학습된 CNN모델 없이 domain mix up 증강만을 이용하면 target 데이터셋에 대한 모델을 만들 수 있고 source와 target 도메인 차가 큰 MNIST와 MNIST\_M 데이터셋에서는 ADDA 방법보다 좋은 성능을 기록했다. 또한 domain mix up 방법과 ADDA와 결합하거나 다른 기법을 이용하면 기존 모델의 성능을 더 개선할 여지 또한 남아 있다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| metric: ACC | USPS -> MNIST | MNIST -> USPS | MNIST -> MNIST\_M | MNIST\_M -> custom | MNIST -> custom | USPS -> custom |
| SOURCE ONLY | 51% | 69.90% | 44.30% | 25.10% | 13.20% | 15.80% |
| ADDA | 95.70% | 91.60% | 61.70% | 35.00% | 20.50% | 15.60% |
| ADDA + mixup | 94.80% | 88.60% | 61.60% | - | - | - |
| Mix up (ours) | 81.90% | 83.70% | 64.70% | 32.10% | 19.00% | 16.20% |
| Domain Mix up(ours) + ADDA | 95.50% | 88% | 84.30% | 38.90% | 37% | 15.60% |
| TRAIN ON TARGET | 95.60% | 96.00% | 91.80% | 87.90% | 87.90% | 87.90% |

**<표1>**

**5. 결론 및 기대효과**

**5.1 기대 효과**

Domain adaptation과 Domain mix up, ADDA 등을 이용해 모델을 학습시키고 성능을 향상시키는 연구를 진행했고 이번 연구에서 진행한 Domain mixup방법이 다른 방법론보다 좋은 결과를 낸 데이터 셋도 존재했기에 유의미한 성과를 얻을 수 있었다. 연구 초기에는 성능이 많이 좋지 않게 나왔지만 연구를 진행하면서 데이터 셋을 수정하거나 Domain mixup방법에 ADDA방법을 추가하는 등의 노력으로 성능의 개선이 있었다. 하지만 아직은 custom dataset과 같은 다양한 환경에서의 데이터를 완벽하게 인식하지는 못하는 부분이 있기에 앞으로 개선해야할 부분 또한 남아있다.

이 연구로 통해 다양한 환경에서 얻어진 data일지라도 도메인 차이를 최대한 줄여주기 때문에 일반적인 성능이 나올 것으로 기대한다. 또한 새로운 데이터 셋에 대한 새로운 모델을 만들지 않아도 되니 시간적 비용을 많이 줄일 수 있다. 그리고 앞으로 성능 개선을 위해 여러 실험을 통해 얻어진 결과를 기존 방법론들과 비교하고 더 다양한 상황에서도 좋은 성능을 갖추고 적용함으로써 필기인식을 정확히 사용할 수 있는 범위가 더 확장될 것을 기대한다.

**5.2 추후 연구 방안**

DRANet: Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation[7]을 보면 CNN을 이용해 source와 target 도메인의 feature를 뽑고 GAN을 이용해 서로의 스타일 바꾼 뒤 source 데이터셋을 target 데이터셋처럼, target 데이터셋을 source 데이터셋처럼 보이게 하는 기법이다. 이러한 기법처럼 pretrained 된 CNN model을 가지고 이미지의 feature를 알아낸 다음 해당 feature를 가지고 domain adaptation을 하면 지금의 방법보다 더 좋은 성능을 낼 수 있을 거라 기대한다.

**6. 참고문헌**

[1] <https://hongdoki.github.io/2018/05/28/skimming-a-theory-of-learning-from-different-domains.html>

[2] <https://sdc-james.gitbook.io/onebook/4.-and/5.1./5.1.3.-mnist-dataset>

[3] <http://dsba.korea.ac.kr/seminar/?mod=document&uid=1325>

[4] <https://github.com/zhaoxin94/awesome-domain-adaptation>

[5] <https://arxiv.org/pdf/1505.07818.pdf>

[6] <https://arxiv.org/pdf/1702.05464.pdf>

[7] https://arxiv.org/abs/1904.01941

[8] <https://arxiv.org/pdf/2103.13447v2.pdf>