# Mask R-CNN 과 zi2zi 모델을 활용하여 탐지된 객체의 스타일을 변환시키는 신경망 모델

조인수*, 최동빈*, 박용범*
*단국대학교 컴퓨터학과
**단국대학교 소프트웨어과
min428a1@gmail.com, dbchoi85@gmail.com, ybpark@dankook.ac.kr

# Neural network model for detected object style transformation using Mask R-CNN and zi2zi

In-su Jo*, Dong-Bin Choi*, Young B. Park **
* Dept. of Computer, Dankook Univerisy
** Dept. of SoftWare, Dankook Univerisy

**요 약**

스타일 변환 모델은 이미지 전체나 이미지 내에서 사용자가 지정한 영역을 대상으로 스타일을 변환시킨다. 이런 방식은 이미지 내의 다수의 객체에 대해 스타일 변환을 시행할 때 일일이 영역을 지정해 줘야 한다는 불편함과 결과물의 전체 해상도가 떨어진다는 한계를 가지고 있다. 본 논문에서는 이런 한계들을 극복하기 위해 객체탐지 모델과 스타일변환 모델을 연동한 객체스타일변환모델을 제안하고 모델 간 연동방법에 대해 자세히 서술한다. 객체탐지모델인 Mask R-CNN 을 통해 필요한 객체를 탐지하고 탐지한 객체의 특징맵들을 스타일변환 모델인 zi2zi 의 입력 값으로 전달하여 이미지 내의 필요한 객체들만 스타일변환이 이루어지도록 모델이 동작한다. 이러한 모델은 기존에 있는 두 모델을 재사용함으로써 모델을 처음부터 새로 설계할 필요가 없다는 장점이 있으며, 공개된 다양한 모델들을 서로 융합하여 사용할 수 있는 방법을 제시하는데 도움을 줄 것이다.

## 1. 서론

이미지의 특징 분포를 학습하는 모델이 등장하고 이미지 내의 스타일을 gram matrix 와 연관지어 정의하면서 이미지 스타일 변환 모델은 크게 발전하기 시작했다.[1,2,3] 스타일 변환은 이미지의 content 를 유지하면서 재질 또는 폰트와 같은 요소들은 변화시켜 새로운 이미지를 만드는 작업이다.

이미지의 확률분포를 학습하는 대표적인 모델로 Generative Adversarial Network(GAN) 이 있다. GAN 의 적대적 학습방법은 타겟 이미지의 스타일을 학습하는 동안 생성모델이 자동으로 스타일을 캐치할 수 있도록 해준다. 현재 GAN 과 Adain 기법을 활용한 스타일 변환 기법이 등장하였으며 높은 퀄리티의 결과물을 생성하는 것으로 잘 알려져 있다.[4]

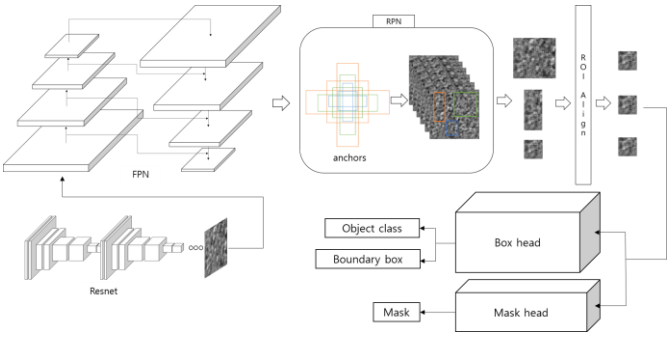
이런 스타일 변환 기법을 이용한 결과물들은 이미지 내의 재질을 바꾸거나 글자의 폰트 스타일을 바꾸는 등 다양한 분야에서 사용되고 있다. 하지만 기존에 존재하는 스타일 변환 기법들은 이미지 전체의 스타일을 바꾸는데 초점이 맞춰져 있거나, 스타일 변환을 적용하려는 객체 영역을 일일이 지정해 줘야한다는 한계점을 가지고 있다. 본 논문에서 이 한계점을 극복하기 위해서 기존에 존재하는 객체탐지모델인 Mask R-CNN 과 스타일변환모델인 zi2zi 를 연동한 객체스타일변환 모델을 제시한다.[5,6] Mask R-CNN 은 이미지 내에서 특정 객체를 탐지하여 영역을 표시해주는 기능을 한다. 이 모델은 스타일 변환이 필요한 객체를 탐지하여 zi2zi 의 입력값으로 해당 객체의 특징맵을 전달하는 역할을 담당한다. Zi2zi 는 전달받은 객체의 특징맵을 통해 스타일을 변환하는 역할을 담당하게 되고, 이렇게 변경된 객체는 이미지 내의 지정 영역에 덮어씌이게 된다. 이런 방법은 기존의 모델을 최대한 활용한다는 점에서 모델설계가 상대적으로 쉽다는 장점을 가지며, 기존의 스타일 변환 기법들의 한계들을 개선시킬수 있는 방법이 될 것이다.

본 논문에 관련된 연구는 2 장에서 서술하였고, 3 장에서는 Mask R-CNN 과 zi2zi 를 연결하는 방법에 대해 자세히 서술하였다. 4 장에서는 연동된 모델을 학습하는 방법에대해 서술하였고, 5 장에서 결론으로 마무리하였다.

## 2. 관련 연구

### 2.1 Mask R-CNN

Mask regions with convolutional neural networks(Mask R-CNN) 은 residual neural network(ResNet)에서 기본적인 특징맵을 추출하고 feature pyamid network(FPN)에서 이 특징맵을 입력값으로 받아 다양한 크기의 특징맵을 region proposal network(RPN)에 전달한다.[7,8] RPN 은 객체에 해당하는 특징맵들을 반환하며 이 특징맵들은 ROI-Align 을 통해 모두 동일한 크기로 바뀌게 된다. 이렇게 정제된 객체 특징맵들은 classification, boundary box(Bbox), 마스크를 담당하는 브런치의 입력값으로 사용된다. ROI-Align 은 Mask R-CNN 만 사용되는 기법 중 하나인데, Faster R-CNN 에서 사용한 ROI-Pooling 이 반올림하여 사이즈를 계산하는 방식과 달리 소수점자리까지 세밀하게 사이즈를 계산한다. ROI-Align 를 통해 얻은 객체들의 특징맵들은 모두 같은 크기를 가진 데이터이므로 다른 모델의 입력값으로 사용하기 용이하다. 그림 1 에서는 Mask R-CNN 의 전체 구조를 보여주며 ROI-Align 에서 모든 feature 들이 같은 크기로 전환되는 것을 확인할 수 있다.

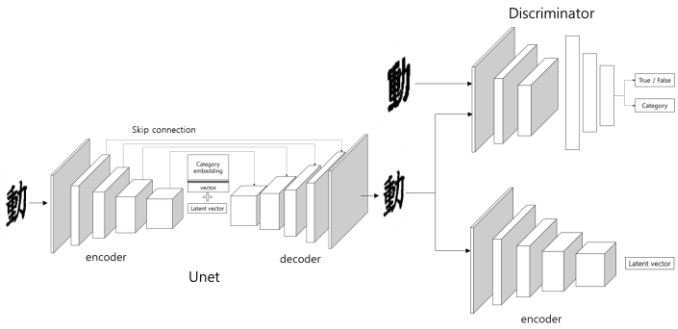



(그림 1) Mask R-CNN 모델 구조

### 2.2 GAN 을 이용한 스타일 변환모델

Generative Adversarial Network(GAN)은 형태에 따라 다양한 기능을 수행한다. 그중
이미지의 스타일을 변환하는 기능을 가진 GAN 도 존재한다. 간단한 모델부터 복잡한 모델까지 다양한데 기본적인 원리는 유사하다. 타겟 이미지의 스타일을 갖는 이미지를 생성하도록 생성자를 학습하되 소스 이미지의 content 를 유지하도록 추가 학습을 시켜준다. 그림 2 는 GAN 을 이용해서 만든 스타일변환 모델 중 하나인 zi2zi 의 모델구조를 보여준다.

GAN 은 궁극적으로 생성자가 목적에 맞는 데이터를 생성할 수 있도록 하는 것을 목표로 한다. 따라서 복잡한 형태를 가진 GAN 이라 할지라도 데이터를 생성하는데 쓰이는 실제모델은 생성자 하나이다. 따라서 모델과 모델을 융합할 때 학습이 완료된 생성자만 활용하더라도 원하는 결과물을 도출해 낼 수 있다.




(그림 2) zi2zi 모델 구조

## 3. 접근

본 논문에서는 객체탐지모델과 스타일변환모델로 Mask R-CNN 과 zi2zi 를 사용한다. Mask R-CNN 은 Bbox 뿐만 아니라 마스크로 객체의 위치를 탐지한다. 마스크로 객체를 탐지하기 위해서는 보다 정밀한 특징 분석이 요구되기 때문에 Mask R-CNN 은 ROI-Ailgn 기법을 사용하여 정밀한 특징분석을 시행한다. Zi2zi 는 중국어의 폰트를 변환하는데 쓰이는 gan 을 이용한 모델이다. 다른 스타일변환모델에 비해 비교적 단순하여 구조 파악이 쉽고, 카테고리 임베딩을 사용하여 일대 다 학습에 용이하다. 이 두 모델의 출력과 입력의 형태는 서로 다르기 때문에 원 상태 그대로 모델을 연결하기는 매우 어렵다. 따라서 Mask R-CNN 를 통해 얻은 출력값이 zi2zi 의 입력값으로 쓰이도록 하는 방안이 이 논문의 핵심이다.

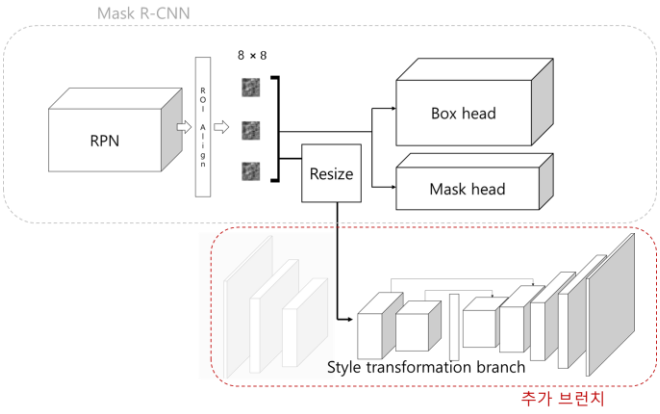
### 3.1 두 모델의 입출력

Zi2zi 는 256 × 256 크기의 한 글자가 적힌 이미지를 입력받아 해당 글자의 다른 스타일이 그려진 이미지를 출력한다. 기본적으로 입력되어야 할 데이터의 크기는 256 × 256 이며 원본 이미지 형태를 필요로 한다. Mask R-CNN 은 다수의 객체가 존재하는 큰 이미지를 입력을 받아 여러가지 종류의 형태로 결과를 출력한다. 출력 형태는 크게 객체 클래스, 객체 Bbox 좌표, 객체 마스크 좌표가 있다. 이 세 출력 모두 zi2zi 의 입력값으로 활용하기에 크기도 형태도 맞지 않는다. 하지만, 두 모델 모두 이미지의 특징이 담긴 특징맵을 입출력으로하는 레이어를 가지고 있다는 공통점을 가지고 있다. 이 특징맵을 이용하면 입출력의 크기를 똑같이 맞출 수 있을 뿐만아니라 형태 또한 비슷하게 만드는 것이 가능하다.

Zi2zi 의 생성자영역은 Unet 형태를 띄고 있다. 입력값이 생성자의 인코더 레이어를 지나면서 출력된 특징맵이 풀링기법에 의해 크기가 ½씩 줄어들게 되고, 이때 생기는 특징맵들은 학습을 통해 본래의 이미지의 특징들이 잘 드러나도록 변화된다. Mask R-CNN 도 마찬가지로 ResNet 을통해 특징맵 추출이 가능하다. 단, Mask R-CNN 은 여러 객체가 존재하는 큰 이미지를 입력 데이터로 받기 때문에 같은 특징맵이라 하더라도 크기와 함축된 특징들이 차이가 있다. 그렇기 때문에 단순히 원본 이미지에서 추출한 특징맵이 아니라 객체에 해당하는 특징맵을 찾아야 한다.

이 특징맵은 Mask R-CNN 의 box head 영역에서 찾을 수 있다. Box head 영역은 resnet 을 통해 얻은 특징맵이 region proposal network(RPN)를 지난 후에 거치는 영역으로, 객체의 Bbox 좌표와 클래스의 정보를 결과로 추출하는 영역이다. RPN 은 입력받은 특징맵 내에서 객체에 해당하는 부분만 잘라 각기 다른 크기의 특징맵을 좌표와 함께 출력한다. Convolution 과 fully connected layer 로 이루어진 box head 영역은 입력값의 크기가 고정되어야 한다는 특징을 가지고 있다. 때문에 각기 다른 크기를 가진 객체의 특징맵들은 box head 의 입력값으로 바로 사용할 수 없다. 이 특징맵들은 ROI-Align 기법에 의해 모두 같은 크기로 변환된 후 box head 의 입력값으로 사용된다.

Mask R-CNN 의 box head 영역에서 ROI-Align 기법에 의해 출력된 특징맵을 zi2zi 내 인코더의 레이어 중 크기가 맞는 레이어의 입력값으로 전달하여 하나의 작업과정으로 만들 수 있다. 이 방법으로 연결된 두 모델은 Mask R-CNN 의 box head 영역에 스타일변환 작업을 가진 브런치가 추가된 형태로 이해할 수 있다. 그림 3 에서 Mask R-CNN 의 출력단에 zi2zi 의 생성자 영역을 연결하여 style transformation branch 가 추가된 모습을 보여준다.




(그림 3) zi2zi 의 생성자 영역을 Mask R-CNN 의 출력단에 style transformation 브런치로 추가한 구조를 보여준다

### 3.2 세부 파라미터 조절

Mask R-CNN 과 zi2zi 를 연결할 위치를 알았다고해서 두 모델을 바로 연결할 수 있는 것은 아니다. Mask R-CNN 의 ROI-Align 기법의 출력값은 box head 영역에서 기본적으로 7×7 크기로 설정되어 있다. 그에 반해 zi2zi 내 인코더의 레이어가 받아들이는 입력값의 크기는 2 의 배수형태(256, 128, …, 8, 4, 2)로 이루어져 있다. 따라서 zi2zi 로 출력값을 전달하기 위해서는 Mask R-CNN 을 그대로 학습시키는 것이 아니라 ROI-Align 의 크기를 2 의 배수인 8 로 변경한 후 학습을 진행해야 한다. 만약 MS COCO 나 imagenet 같은 pre training 데이터를 불러와 사용할 경우, head 영역의 입출력 크기가 맞지않아 오류가 발생하므로 head 영역을 제외한 나머지 영역을 불러오도록 설정하여야 한다.

Mask R-CNN 은 일반 CNN 과는 달리 레이어를 지나는 텐서들의 형태가 다르다. Classification 에 쓰이는 일반 CNN 에 사용되는 텐서는 (batchsize, width, height, depth) 형태를 가지고 있으며 zi2zi 의 생성자 또한 같은 형태의 텐서를 사용한다. 하지만 Mask R-CNN 의 head 영역은 배치 하나당 여러 개의 객체 특징맵을 뽑아내기 때문에 하나의 차원을 더 가진 (batchsize, roi, width, height, depth) 형태의 텐서를 사용한다. 따라서 batchsize 와 roi 를 하나의 차원으로 만들어 (batchsize × roi, width, height, depth)형태로 resize 과정을 거친 후 zi2zi 로 전달해야 한다.

한가지 더 고려해야 할 부분은 zi2zi 의 생성자영역이 Unet 형태를 가지고 있다는 것이다. Unet 은 오토인코더와는 다르게 인코더와 디코더가 skip connection 이라는 기법에 의해 연결되어 있다. 오토인코더에 skip connection 을 사용하게 되면 인코더의 손상되기 전 정보를 디코더에 전달해 줌으로써 생성되는 결과물의 질을 높일 수 있다. 하지만 Mask R-CNN 으로부터 전달받은 특징맵은 입력되는 위치가 달라지기 때문에 레이어별 skip connection 의 사용여부도 수정하여야 한다. 8×8 크기의 입력값을 취하는 경우, 그 이하의 레이어들(4×4 와 2×2 크기의 입력데이터를 취하는 레이어들)만 skip connection 을 적용시켜야 한다. 그렇지 않으면 디코더 영역에서 존재하지 않은 입력값을 skip connection 을 통해 요구하는 문제가 생기게 된다. 그림 3 에서 사용되지 않은 인코더의 레이어들은 제외되며 해당 skip connection 또한 디코더에서 사용하지 않음을 확인할 수 있다. 표 1 에서 보여준 세부사항을 통해 인코더와 디코더의 입출력 사이즈와 skip connection 연결관계를 확인할 수 있다.

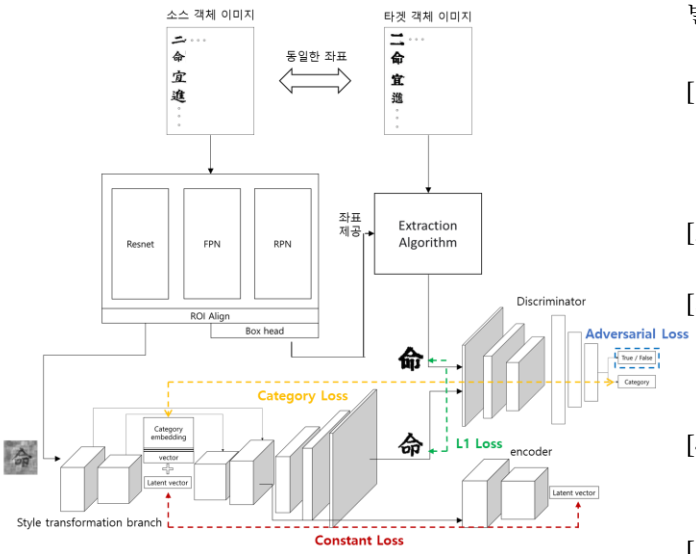
<표 1> zi2zi 의 인코더와 디코더 레이어별 세부사항

| model | Layer | Filter | Stride | Output size | Input size | Skip connection |
|---|---|---|---|---|---|---|
| Encoder | conv_layer_1 | 5×5 | 2×2 | 128×128×64 | 256×256×3 | |
| | conv_layer_2 | | | 64×64×128 | 128×128×64 | skip_output_1 |
| | conv_layer_3 | | | 32×32×256 | 64×64×128 | skip_output_2 |
| | conv_layer_4 | | | 16×16×512 | 32×32×256 | skip_output_3 |
| | conv_layer_5 | | | 8×8×512 | 16×16×512 | skip_output_4 |
| | conv_layer_6 | | | 4×4×512 | 8×8×512 | skip_output_5 |
| | conv_layer_7 | | | 2×2×512 | 4×4×512 | skip_output_6 |
| | conv_layer_8 | | | 1×1×512 | 2×2×512 | skip_output_7 |
| Decoder | deconv_layer_1 | 5×5 | 2×2 | 2×2×512 | 1×1×512 | skip_input_7 |
| | deconv_layer_2 | | | 4×4×512 | 2×2×512 | skip_input_6 |
| | deconv_layer_3 | | | 8×8×512 | 4×4×512 | skip_input_5 |
| | deconv_layer_4 | | | 16×16×512 | 8×8×512 | skip_input_4 |
| | deconv_layer_5 | | | 32×32×256 | 16×16×512 | skip_input_3 |
| | deconv_layer_6 | | | 64×64×128 | 32×32×256 | skip_input_2 |
| | deconv_layer_7 | | | 128×128×64 | 64×64×128 | skip_input_1 |
| | deconv_layer_8 | | | 256×256×3 | 128×128×64 | |

## 4. 데이터 전처리

Zi2zi 와 객체스타일변환 모델은 학습 시 입력데이터의 형식이 다르다. Zi2zi 는 스타일 학습을 위해 소스이미지 뿐만 아니라 소스이미지와 같은 종류에 해당하는 타겟이미지도 입력값으로 요구한다. 하지만 Mask R-CNN 을 거쳐 얻을 수 있는 특징맵은 소스이미

지에만 해당되며, 그 특징맵이 어떤 종류의 이미지인지 알 수 없다. 이 문제를 해결하기 위해서는 이미지 내의 소스 객체와 동일한 좌표에 해당 타겟 객체가 있는 이미지가 필요하다. 소스 객체가 있는 이미지를 입력값으로 취하고 해당 배치에대한 Bbox 좌표를 순서대로 얻는다. 이렇게 얻은 Bbox 좌표를 타겟 객체가 있는 이미지에 순서대로 적용시켜 소스이미지에 대응하는 타겟이미지를 출력해 낸다. 이 후에는 zi2zi 에서 학습하던 것처럼 소스이미지에 해당하는 특징맵과 대응하는 타겟이미지를 통해 스타일변화 브런치를 학습한다. 그림 4 에서 같은 종의 객체를 동일한 좌표로 설정한 소스 객체이미지와 타겟 객체이미지를 이용하여 모델을 학습하는 전체 프로세스를 보여준다.




(그림 4) 객체스타일변환모델의 구조와 loss function을 보여준다. 소스 객체이미지와 타겟 객체이미지를 학습데이터로 필요로 하며 타겟객체이미지는 추출알고리즘을 거쳐 discriminator 에서 사용된다.

## 5. 결론

기존의 스타일변환 모델은 이미지 전체를 대상으로 스타일 변환이 이루어지거나 스타일변환이 적용될 객체 영역을 사용자가 일일이 지정해야 한다는 한계점을 가지고 있다. 본 논문에서는 이 한계점을 극복하기 위한 방안으로 Mask R-CNN 과 zi2zi 를 합친 객체스타일변환 모델을 제안한다. 객체스타일변환 모델은 새로 모델을 설계하는 것이 아닌 이미 널리 쓰이고 있는 두 모델을 연동하여 만든 모델로서 모델 설계가 비교적 쉽다는 장점을 가진다. 두 모델 연동을 위해 모델간 연결할 위치부터 세부 파라미터 조정까지 자세히 설명하였고, 연동된 두 모델을 pre-training 을 활용하여 학습하는 방법 또한 제시하였다. 객체스타일변환 모델은 이미지 내에서 원하는 객체를 대상으로 스타일 변환이 이루어지게 하는 역할을 하며, 실사 동영상과 같은 전체 이미지 해상도 보존이 필요한 분야를 대상으로 활용할 수 있다. 이 모델은 Bbox 영역에서 스타일 변환이 이루어지기 때문에 객체를 벗어난 영역도 스타일변환이 이루어질 수 있다는 한계를 가지고 있다. 향후 한계극복을 위해 객체의 마스크영역을 대상으로 스타일변환이 이루어지도록 모델을 발전시키는 연구가 필요하다.

## 감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 육성지원사업의 연구결과로 수행되었음(IITP-2020-2017-0-01628)

이 논문은 2021 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2021-0-00177, 스마트 컨트랙트의 개발-배포-실행의 전주기적 취약점 및 신뢰성 오류 개선 기술개발)

## 참고문헌

[1] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. NIPS, 2014.

[2] Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[3] Gatys, L. A., Ecker, A. S., and Bethge, M.. Image style transfer using convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In CVPR, 2019.

[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-´ CNN. arXiv:1703.06870, 2017.

[6] Yuchen Tian. zi2zi: Master chinese calligraphy with conditional adversarial networks, https://github.com/kaonashi-tyc/zi2zi., 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, arXiv:1512.03385, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.