

GAN을 활용하여 나만의 동화책 만들기

요 약

“나 자신”이 “내가 찍은 사진”의 주인공이 되는 그림책을 만드는 웹사이트를 제작하였으며, Style Transfer로는 “AnimeGAN”을, Pose Transfer로는 “Porogressive Pose Attention Transfer for Person Image Generation” 논문을 참고하였다. 웹사이트를 통해 풍경 사진과 인물 사진, 그리고 풍경 사진에서 인물을 합성하고자 하는 위치를 선택하면 애니메이션 스타일로 변형된 합성된 이미지가 출력되는 웹페이지를 제작하였다.

1. 서 론

“내가 찍은 사진이 그림책처럼 스타일이 변한다면 어떻게 될까?”라는 궁금증에서 시작한 프로젝트로, 아이디어를 점차 발전시켜 “나 자신”이 “내가 찍은 사진”의 주인공이 되는 그림책을 만드는 웹사이트를 고안하게 되었다. 풍경 이미지를 애니메이션처럼 스타일을 변형시키는 GAN(General Adversarial Network)과 인물 이미지의 인물 포즈를 변형시키는 GAN을 사용한다.

2. 배경 지식

2.1 GAN(General Adversarial Network)

GAN[1]은 딥러닝 모델 중 이미지 생성에 사용되는 모델로 input 데이터 셋과 유사한 이미지를 만들어 내는 것을 목표로 한다. GAN은 생성자(Generator)와 판별자(Discriminator) 두 개의 모델이 동시에 학습되며, 서로 적대적인 방식으로 학습을 진행한다. 생성자는 실제 데이터의 분포를 학습하여 그럴싸한 이미지를 생성해 내며, 판별자는 그 생성된 이미지가 원래의 데이터인지 아니면 생성자로부터 만들어진 데이터인지 판별한다. 그리고 이 단계를 반복하면서 각각의 역할을 가진 두 모델을 통해 진짜 같은 가짜 데이터를 생성해낸다.

2.2 Pose Transfer

Pose Transfer란 인물의 사진과 포즈가 주어졌을 때, 인물 사진의 포즈를 주어진 포즈로 변형하는 것을 말한다. 그림 1을 보면 Condition image 속의 인물이 Target poses들의 포즈로 각각 포즈가 변환된 것을 볼 수 있다.

본 프로젝트에서는 Pose Transfer를 위해 GAN을 사용하였으며, 참고한 논문은 “Progressive Pose Attention Transfer for Person Image Generation”[2, 3]이며, 논문에서 제안하는 네트워크의 구조는 그림 2과 같다.

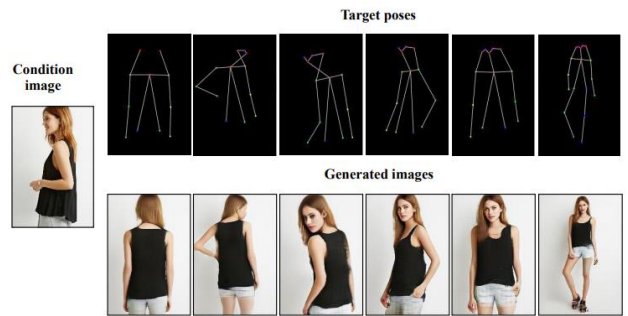


그림 1. Pose Transfer[2]

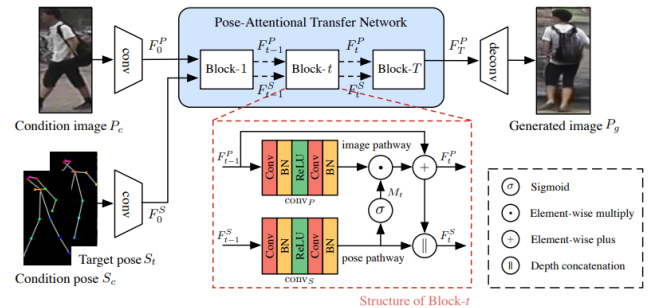


그림 2. [2]에서 제안하는 네트워크 구조

2.3 Style Transfer

Style Transfer란 두 영상이 주어졌을 때 content image의 주된 형태는 유지하면서 스타일은 style image와 유사하게 변형시키는 것을 말한다. 아래 그림을 보면 주택 사진을 content image로 설정하고, 고흐의 작품을 style image로 설정하면, 주택의 형태와 배치는 유지되면서 content image의 스타일만 바뀐 것을 사용할 수 있다[4].



그림 3. CNN을 활용한 Image Style Transfer[4]

본 프로젝트에서는 GAN을 이용하여 style transfer를 진행하였으며, 참고한 논문은 “AnimeGAN: A Novel

Lightweight GAN for Photo Animation”[5, 6]으로, 실제 풍경이나 사람 사진을 애니메이션 스타일로 변경하는 GAN이다. 논문에서 제안하는 네트워크의 구조는 아래 그림과 같다.

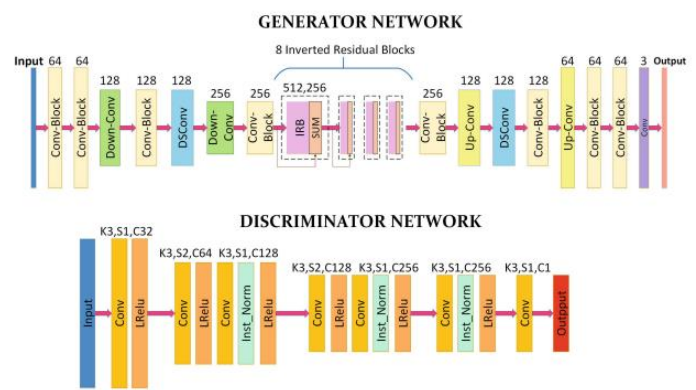


그림 4. AnimeGAN의 generator와 discriminator 구조[5]

2.4 Flask

Flask[7]란 파이썬을 활용하여 웹 어플리케이션을 만들 수 있는 프레임워크로, Jinja 템플릿 엔진과 Werkzeug WSGI 툴킷에 의존한다. 본 프로젝트에서는 대부분의 코드를 파이썬으로 작성하였으며, GAN 모델과의 원활한 연결을 위해 Flask를 활용하여 웹 어플리케이션을 제작하였다.

3. 웹사이트 제작

3.1 프로그램 구조도

웹사이트는 Flask를 이용하여 제작하였으며, 웹페이지는 HTML으로, 네트워크와 사진을 합성하는 부분은 Python으로 코드를 작성하였다. 본 프로젝트에서 제작한 프로그램의 구조도는 다음과 같다.

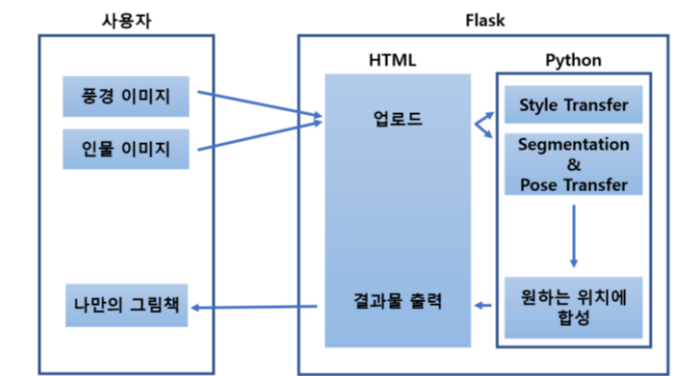


그림 5. 프로그램 구조도

먼저 사용자가 합성하고자 하는 풍경 이미지와 인물 이미지를 각각 홈페이지에 업로드 한다. 업로드가 완료되면 사용자가 원하는 포즈대로 인물 사진의 포즈를 변경한다. 그 후, 풍경 이미지와 변경된 인물 이미지는 애니메이션 스타일로 변형되며, 이후 합성을 위해 인물 이미지는 segmentation을 진행한다. 이후

사용자가 합성을 원하는 위치를 선택하면 해당 위치로 사진이 합성된다. 합성 완료 후 웹사이트를 통해 결과물을 보여주며, 사용자는 이를 확인할 수 있다.

3.2 Pose Transfer 결과

“Progressive Pose Attention Transfer for Person Image Generation” 논문[2]에서 제시한 모델을 이용하였으며, pretrained model[3]을 활용하여 원하는 포즈대로 변형할 수 있도록 코드를 작성하였다.

아래는 Pose Transfer를 실행한 결과이다. 왼쪽 사진부터 순서대로 원본 인물 사진, 원본 인물 사진에서 포즈를 추출한 사진, 원하는 포즈의 인물 사진, 원하는 포즈를 추출한 사진, 원하는 포즈로 변형된 사진이다.



그림 6. 원본 인물 사진과 변형된 인물 사진

3.3. Style Transfer 결과

“AnimeGAN: A Novel Lightweight GAN for Photo Animation” 논문[5]에서 제시한 모델을 실행한 결과이다. “미야자키 하야오” 작가의 화풍으로 실제 사진의 스타일을 변형시키는 모델의 weight 값을 불러와[6] 사용하였다.



그림 7. 원본 풍경 사진[8]



그림 8. 변형된 풍경 사진



그림 9. 인물 사진



10. 변형된 인물 사진



그림 13. 최종 합성 결과

3.4 Segmentation 결과

이후, 인물 사진과 풍경 사진을 합성하기 위해 mask 이미지를 만들어야 하므로, 인물 사진에 대해 segmentation을 진행한다. Segmentation은 이미 구현되어 있는 github 소스 코드[9]를 활용하였다. segmentation 결과 사진은 아래와 같다. Pose Transfer와 Style Transfer를 수행하였음에도 불구하고 배경과 인물이 잘 분리된 것을 확인할 수 있다.

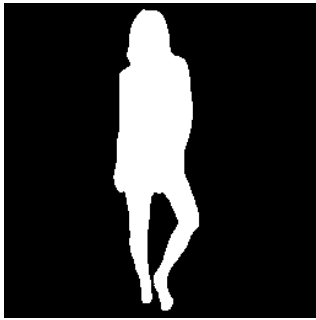


그림 11. 인물사진을 Segmentation 한 결과



그림 12. 변형된 인물 사진에서 배경을 제거한 사진

3.5 합성 결과

본 프로젝트에서는 결과적으로 배경 사진과 인물 사진을 합성한 후 웹사이트에 출력하여 사용자에게 보여준다. 위에서 살펴본 배경 사진과 인물 사진을 합성한 결과는 다음과 같다.

4. 결론

최종 합성 결과를 보면 기존의 원본 사진과 비교하여 굉장히 애니메이션 스타일로 변경된 것을 확인할 수 있다. 또한 인물과의 합성 역시 특별한 문제 없이 자연스럽게 합성이 진행된 것을 확인할 수 있다.

처음 목표했던 것은 네트워크 학습까지였지만, 개발 환경 상 시간이 너무 오래 걸려 학습이 계속 중간에 중단되었다. 따라서 어쩔 수 없이 pretrained model 을 불러와 웹페이지를 제작할 수밖에 없었다. 처음에는 조금 아쉬운 기분이 들었지만 복잡한 GAN 네트워크의 구조를 공부하고 코드를 이해하면서 본 프로젝트에 사용할 수 있도록 가공/수정하는 과정을 겪으면서 공부가 많이 되었다고 생각한다.

추후에 웹 사이트를 더 발전시킬 수 있는 기회가 된다면 우리가 원하는 애니메이션 스타일로 바꿀 수 있도록 학습을 새로 시켜보고 싶고, 웹 사이트 역시 조금 더 발전시켜보고 싶다.

참 고 문 헌

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, Y. Bengio, "Generative Adversarial Networks", In Proceedings of NIPS, pp.2672-2680, 2014
- [2] Zhen Zhu, Tengpeng Huang, Baoguang Shi, Miao Yu, Bofei Wang, Xiang Bai, "Progressive Pose Attention Transfer for Person Image Generation", CVPR, 2019
- [3] tengteng95, "Pose-Transfer", <https://github.com/tengteng95/Pose-Transfer>, 2019(last updated-master branch)
- [4] Leon A.Gatys, Alexander S.Ecker, Matthias Bethge, "Image Style Transfer Using Convolutional Neural Networks", In Computer Vision and Pattern Recognition(CVPR), pp. 2414-2424, 2016
- [5] Jie Chen, Gang Liu, Xin Chen, "AnimeGAN: A Novel Lightweight GAN for Photo Animation", In

International Symposium on Intelligence Computation and Applications, pp.242–256, 2019

[6] TachibanaYoshino, “AnimeGANv2”, <https://github.com/TachibanaYoshino/AnimeGANv2>, 2021(last updated–master branch)

[7] Flask, “Welcome to Flask – Flask Documentation(2.0.x)”, <https://flask.palletsprojects.com/en/2.0.x/>, 2021(version: 2.0.x)

[8] Katherine Lagrave, “14 Travel Tips to Know Before Heading Abroad”, <https://www.cntraveler.com/story/14-travel-tips-to-know-before-heading-abroad>, 2017

[9] Veronica1997, “Human–segmentation”, <https://github.com/Veronica1997/Human-segmentation>, 2021(last updated–master branch)