**딥러닝을 이용한 사진 변환**

강민석, 김성준

경희대학교 컴퓨터공학과

aaaa727@khu.ac.kr ayayryey@khu.ac.kr

**Photographic synthesis by using deep-learning**

Gang minseok Kim sungjun

Department of Computer Science and Engineering, KyungHee University

**요 약**

많은 어플들이 닮은 연예인 혹은 닮은 동물을 찾아주기만 한다. 닮은 것과 원본 사진을 합성시켜서 보여준다면 더욱 신뢰도를 높일 수 있고 이용자들의 흥미를 끌어낼 수 있다. 따라서 CNN(Convolutional Neural Network)을 통해 동물을 학습시키고 사진과 닮은 동물을 찾아낸다. 그리고 이를 합성시켜 출력해주는 프로그램을 설계한다.

**1. 서 론**

**1.1. 연구 배경**

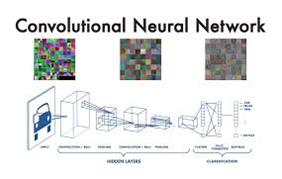
알파고의 등장으로 인공지능은 전 세계적으로 큰 인기를 얻었다. vgg [3] 모델이나 구글의 GoogLeNet을 통해 데이터를 쉽게 학습할 수 있다. 그리고 DCGAN(Deep Convolutional Generative Adversarial Nets)이나 Neural Style Transfer 등을 통해 영상 산업에도 딥러닝이 쓰이고, 이러한 기술들을 이용하여 머신들이 좀 더 쉽게 이미지를 생성할 수도 있다. 컴퓨터 산업이 발전하여 일반 가정집에서도 좋은 그래픽카드를 보유하고 있다. 많은 사람들이 쉽게 딥러닝 시도할 수 있다. 딥러닝에 관한 오픈소스도 매우 많아 필요한 건 언제든지 가져다 쓸 수 있다. 기존에 나와 있는 많은 어플리케이션 같은 경우 인간-인간으로 닮은 사람을 찾아준다. 사람을 볼 때나 본인이 생각할 때 닮은 연예인도 궁금하지만, 닮은 동물이 궁금할 때도 있다.

**1.2. 연구 목표**

머신러닝을 이용해 사람들이 정확히 짚어내지 못하는 동물과 닮은 특성을 통해 닮은 동물을 찾고 사람과 동물 사진을 합성해준다. 첫 번째 목표로 사람 사진을 넣었을 때 닮은 동물을 찾을 수 있는 모델이 필요하다. 모델의 정확도는 동물 사진을 넣었을 때의 정확도를 기준으로 향상시키도록 구성하였다. 두 번째 목표로 닮은 동물을 찾은 뒤 그 동물 사진과 사람 사진을 합성시켜주는 모델이 필요하다. 또한 매번 새로운 결과물이 나올 수 있도록 합성될 동물 사진이 랜덤하게 입력되게 구현한다.

**2. 기존 연구**

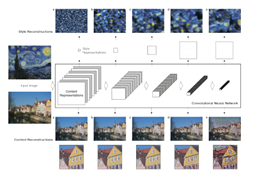
**2.1 CNN**

Fully connected Layer 만으로 구성된 인공 신경망의 입력데이터는 1차원으로 한정된다. 하지만 한 장의 컬러 사진은 3차원 데이터이고, 배치 모드에 사용되는 여러 장의 사진은 4차원 데이터이다. 따라서 Fully connected Layer 신경망을 사용할 때 3차원 사진 데이터를 1차원으로 평면화 시켜야 한다. 이 과정에서 공간 정보가 손실될 수밖에 없다. 결과적으로 이미지 공간 정보 유실로 인한 정보 부족은 인공 신경망이 특징을 추출 및 학습하는데 비효율적이고 정확도를 높이는데 한계가 있다. 이미지 공간 정보를 유지한 상태로 학습이 가능한 모델이 CNN(Convolutional Neural Network)이다.

[그림 1] CNN 모델 구조

**2.2 Neural Style Transfer**

style transfer란, 두 영상(content image &style image)이 주어졌을 때 그 이미지의 주된 형태는 content image와 유사하게 유지하면서 스타일만 우리가 원하는 style image와 유사하게 바꾸는 것을 말한다. 만약 style image가 그림이라면, 그 그림에 존재하는 고유한 특성을 추출해낸다. content image가 사진이라면 , style image에서 뽑아낸 고유한 특징들만 사진에 남기고 사라지게 한다.



[그림 2] style-transfer 원리



[그림 3] style transfer로 만들어진 이미지

**2.3 GAN(Generative Adversarial Network)**

GAN이란 두개의 네트워크로 구성된 심층 신경망 구조이다. 두 네트워크는 서로 경쟁적인 구조이기 때문에 적대적이란 의미가 포함되어 있다. generator라고 불리는 하나의 신경망이 새로운 데이터 인스턴스를 생성하고 다른 하나인 discriminator는 데이터의 진위를 평가한다. 즉 discriminator는 검토한 각각의 데이터 인스턴스가 실제 트레이닝 데이터 세트인지 아닌지를 판단하게 된다. 이를 반복하여 generator는 진짜에 가까운 가짜 이미지를 생성하게 되고 discriminator 그 이미지를 가짜로 식별하는 것을 목표로 한다.



[그림 4] GAN이 이미지를 생성하는 과정

**3.  프로젝트**

**3.1 기존 연구의 문제점**

CNN을 이용해 직접 모델을 구현할 경우 분류 정확도가 매우 낮았다. 데이터셋이 부족하다 판단하여  ImageDataGenerator를 이용해 데이터를 증식했다. 하지만 정확도가 향상하지 않았다. 데이터셋을 바꿔보고 하이퍼 파라미터를 튜닝했지만 이는 근본적인 해결책이 되지 않았다.

Style transfer를 통한 합성과정에도 문제가 있었다. 우리가 원하는 사람과 동물의 합성사진이 아닌  단순히 동물사진의 전체적인 패턴을 그대로 적용시켜 출력된 흐릿한 사진이었다.

**3.2 제안방법**

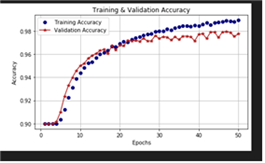
분류 정확도를 높이기 위해 이미지 분류에 유리한 사전훈련된 네트워크인 VGG16을 이용해 데이터를 분류시키고 학습시킨다.

데이터셋에 전처리 과정 거쳐 사진중 얼굴부분만 따로 추출한다.

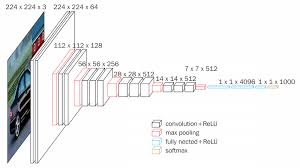
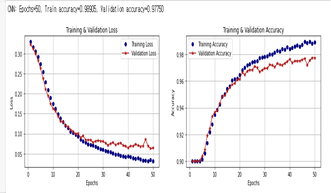
Style transfer에서 하이퍼 파라미터 튜닝을 통해 다양한 결과를 직접 확인하고 목적에 가장 적합한 값을 찾는다.

**4. 실험 결과 분석 및 토론**

**4.1 분류기 모델**

****

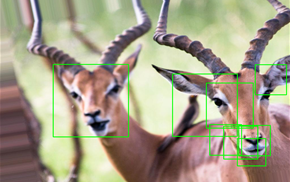
[그림 5] CNN을 이용해 직접 구현한 모델 학습 정확도

 ****

[그림 6] VGG16모델과 정확도

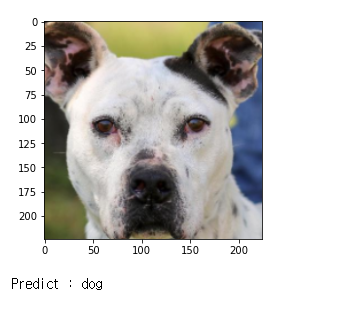
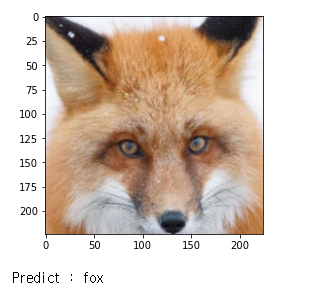
|  |  |
| --- | --- |
|  | [그림 7] VGG16모델의 Layer  위의 그림은 VGG16의 계층구조이다. 먼저 이미지는  넓이,높이,RGB의 3차원 형태이므로 파라미터가 3개가 있는 것을 볼 수 있다.  층이 깊어질수록 차원은 늘어나지만 넓이와 높이는 작아진다. 이렇게 층을 총 16개를 지나 마지막에 10종의 동물에 대한 값을 지정해 주기위해’(None,10)’이라는 결과값을 배출해 내는 것을 볼 수 있다. |

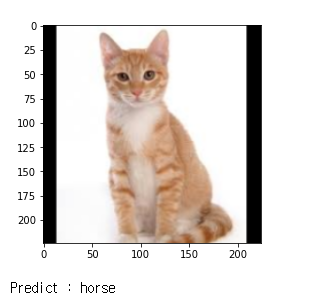
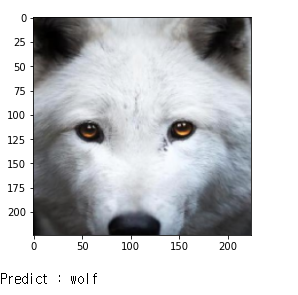
CNN을 이용해 모델을 직접 구현할 경우 정확도가 60%대로 매우 낮았다. 정확도를 높이기 위해 데이터셋을 ImageDataGenerator를 통해 증식해보고 모델의 하이퍼 파라미터를 튜닝 했지만 정확도를 높이는 데 한계가 있었다. 사전 훈련된 네트워크 VGG16과 같은 사용하기 쉽고 좋은 성능을 가진 네트워크를 이용해서 데이터 분류의 정확도를 높였다.



[그림 8] 동물의 얼굴 부분을 추출

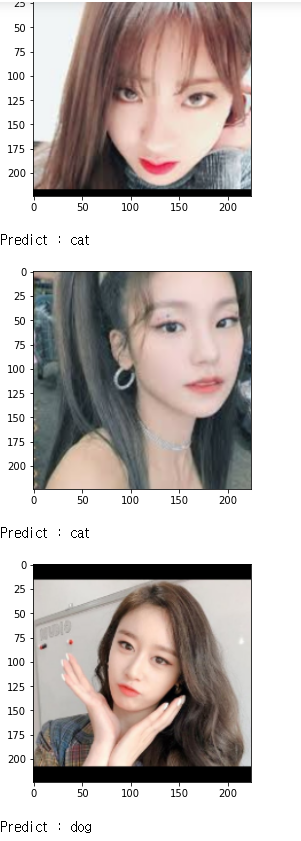
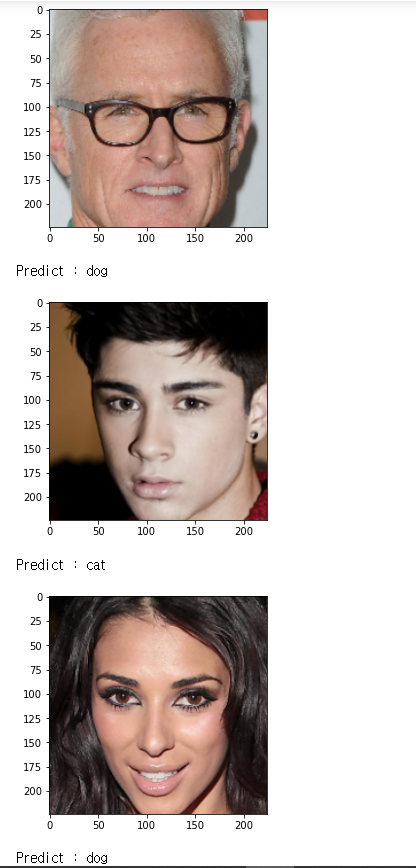
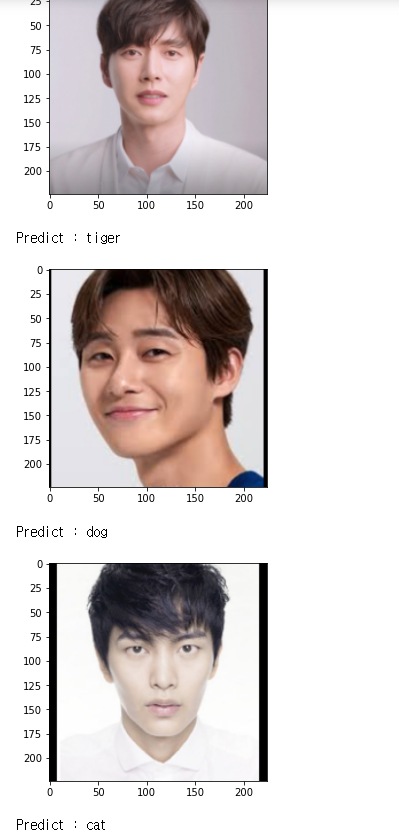
그림 8과 같이 opencv에 제공하는 기술을 통해 동물의 얼굴을 따로 추출해 학습시키려 했지만 동물의 종류가 많아 모든 동물에 기술이 적용되지 않았다. 그래서 Stargan에서 이용한 고화질 동물 얼굴 데이터셋인 AFHQ(animal face high quality)에 몇 종류 동물의 얼굴 사진만 추가해 새로운 데이터셋을 만들었다.





[그림 9] 분류 예측 결과

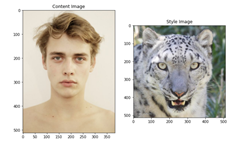
새로운 데이터셋으로 학습시킨 결과 동물의 얼굴 사진에 대한 예측이 훨씬 정확해졌다. 하지만 몸통을 포함한 사진일 경우 정확도가 떨어졌다. 우리의 목표는 사람과 동물의 얼굴을 비교해 가장 유사한 동물을 찾는 것이기 때문에 동물 얼굴의 예측 정확도가 가장 중요하다 판단했다.



[그림 10] 분류기에 사람 사진을 넣었을 때 결과

그림 10은 동물을 분류하는 분류기가 학습한 데이터로 예측 값을 출력한 모습이다. 밑의 Predict 값은 실제 머신이 예측하는 동물의 이름이다. 실제로 고양이를 닮은 연예인을 입력했을 때 예측값이 대부분 고양이로 만족할만한 결과가 나왔다. 흑인이나 백인같은 경우 결과가 대다수 개나 고양이로 나왔다. 이는 개와 고양이가 다양한 형태를 갖고 있어 나온 결과라 판단했다.

**4.2 StyleTransfer**

****

****

[그림 11] loss의 가중치에 따라 바뀐 결과

원본 사진의 손실값과 스타일 사진의 손실값의 가중치를 여러가지 시도해본 결과 원본 사진의 가중치가 너무 높을 경우 합성이 잘 이루어 지지 않았다. 반대로 스타일 사진의 가중치가 너무 높을 경우 원본 사진의 사람 형태가 잘 남아있지 않았다. 따라서 결과 사진을 토대로 가장 적합한 가중치를 설정했다.

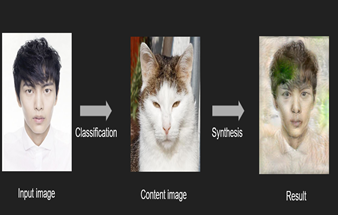
**4.3 데이터 전처리**

****

[그림 12] 얼굴 추출

입력 사진이 다수의 얼굴을 포함할 경우 사진을 잘라내기 위해 opencv face classifier를 통해 데이터 전처리 과정을 거친다.

**4.4 전체적인 시나리오**

****

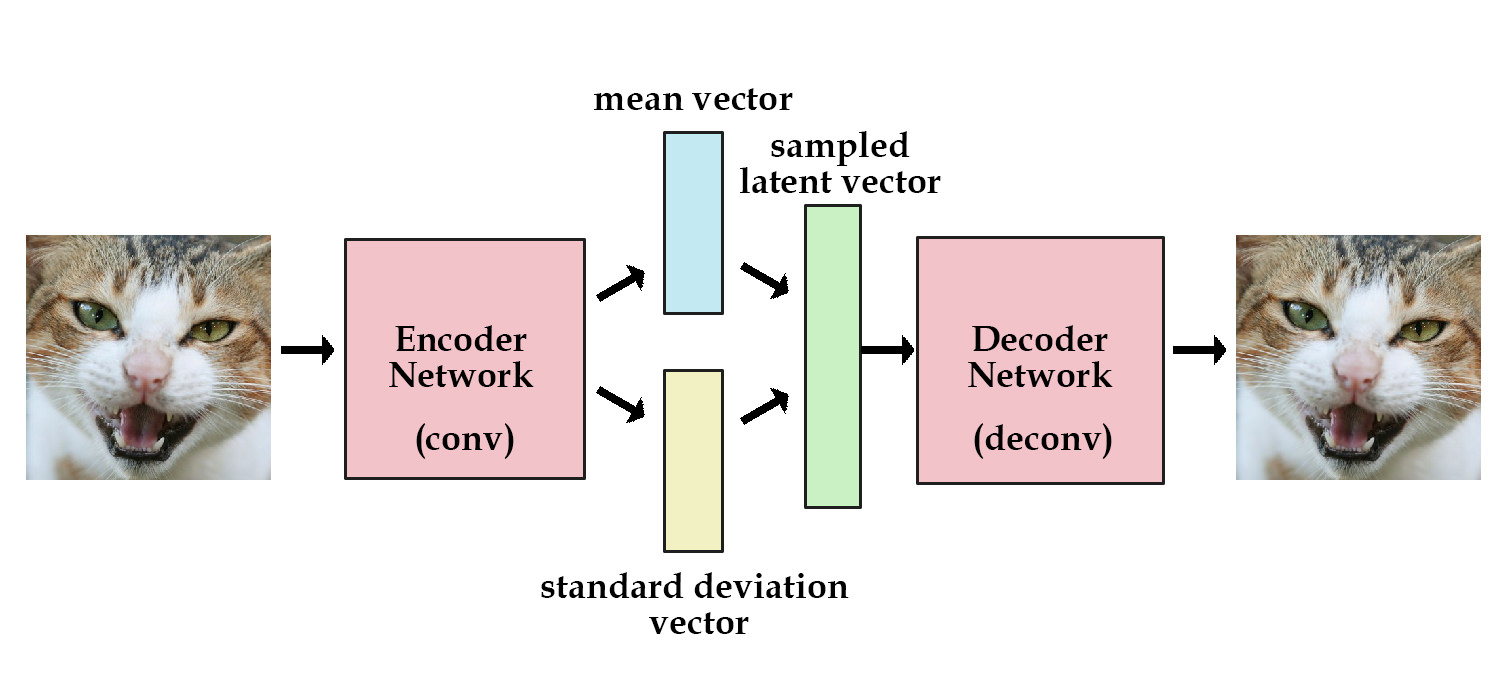
[그림 13] 전체적인 시나리오

그림 13은 시나리오의 흐름을 나타낸 것이다. 사람의 사진이 들어오면 분류기 모델(Classification)이 10가지 정도의 동물 특성을 학습한 값을 기준으로 닮은 동물을 도출해낸다. 해당 동물의 사진 중 랜덤한 사진 하나를 골라서 Style-Transfer(Synthesis) 기술을 통해 두 사진을 합성시킨다.

**5. 결론 및 향후 연구**

**5.1 향후 연구**

닮은 동물을 찾을 때 구체적으로 사람과 동물의 어떤 특징이 어느 정도로 닮았는 지 수치로 확인해보고 싶다.



[그림 14] VAE 구조

VAE에서 인코더는 고차원의 사진을 저차원으로 압축시킨 뒤 다시 디코더를 통해 이를 복원시킨다. 이때 저차원으로 압축된 latent vector를 비교한다면 수치적으로 사람과 동물의 특징을 비교하는 게  가능하다 판단한다. 또한 latent vector를 통해 단순히 사람과 동물의 패턴을 합성시키는 것이 아닌 닮은 부분만 변환시키는 것도 가능하다 생각한다.

어떤 부분이 몇 퍼센트로 유사한 지 알려주고  닮은 특징을 실제로 합성시킨다면 더 좋은 결과가 있을 거라 예상한다.    .

**5.2 결론**

본 논문은 사람과 닮은 동물을 찾아주는 것에 더해 사람과 동물을 합성한 사진을 출력해주는 것을 보여줬다. 분류기의 정확도를 높이기 위해 여러가지 실험을 해 본 결과 사전에 훈련된 네트워크를 쓰는 것이 가장 효율적인 것을 증명했다. 합성하는 과정에서 손실값의 가중치에 따라 출력 이미지도 매우 다양한 것을 확인할 수 있었다. 따라서 필요에 따라 가중치 값을 조정한다면 다양한 용도로 쓰일 수 있을 것이라 판단한다.

향후 연구로 우리는 닮은 동물을 찾을 때 사진을 압축시켜 특징만 추출해 수치화 시킨다. 그리고 이 수치화된 값들을 비교해서 가장 유사한 사진을 찾은 뒤 해당 특징을 합성시킨다면 전체적인 이미지 합성보다 사람과 동물의 가장 유사한 특징만을 이용한 이미지 합성이 가능할 것이다. 이를 통해 구체적으로 입력 사진이 동물의 어떤 특징과 유사한지 확인할 수 있을 것이라 판단한다.

**참 고 문 헌**

**[1]** Francois Chollet, 「케라스 창시자에게 배우는 딥러닝」, 박해선, 길벗(2020)

**[2]** David Foster, 「미술관에 GAN 딥러닝 실전 프로젝트」, 박해선, 한빛미디어(2019)

**[3]** Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, (2014)

**[4]** 꿈 많은 사람 이야기[웹사이트].(2020.04.20). URL:https://lsjsj92.tistory.com/355

**[5]** Google Cloud API[WebSite], URL:https://cloud.google.com/apis/docs/overview?hl=ko

**[6]** Kaggle[WebSite], URL:https://www.kaggle.com/