**Autoencoder, CNN기반의 화가에 따른 style(명화) 분류 가능한 딥러닝 모델**

**2017103972 김성연**

**개 요**

딥러닝을 이용한 masterpiece를 입력으로 해당 작품이 어떤 화가의 작품일지를 분류하는 모델을 제안한다. 언어는 pytorch를 가지고 만든다.

1. **서론** 
   1. **연구 배경**

이번 프로젝트를 시작하게 된 계기는 “Neural Style Transfer”를 통해서였다. Style transfer는 content image와 style image를 이용해 해당 style을 갖는 새로운 이미지를 생성하는 방식이다. 그런데 pre-trained 모델에 style image 입력으로 고흐의 그림이 아닌 이미지 사진을 넣게 되면 제대로 된 이미지를 생성하지 못했다. 이를 통해 content image와 style image는 서로 다른 특징을 가지는 것은 아닌지 궁금했고 딥러닝 학습에 따라 그 결과가 달라질 수도 있겠다는 생각을 했다.

기존에는 Resnet, Vggnet 등의 여러 학습 모델이 개미, 벌과 같이 무언가 형상을 띠는 image를 분류하는 연구가 많이 나왔다. 하지만 style, texture를 분류해주는 딥러닝 모델은 존재하지 않는다. style, texture를 분류한다는 말이 이상할 수 있지만 우리는 앤디워홀의 작품과 반 고흐의 작품이 서로 다르다고 분류할 수 있다. 물론 개미, 벌과 같이 형상으로 정의할 수 없지만 그 둘 작품의 색감이나 스타일이 다르다는 것은 느낄 수 있다.



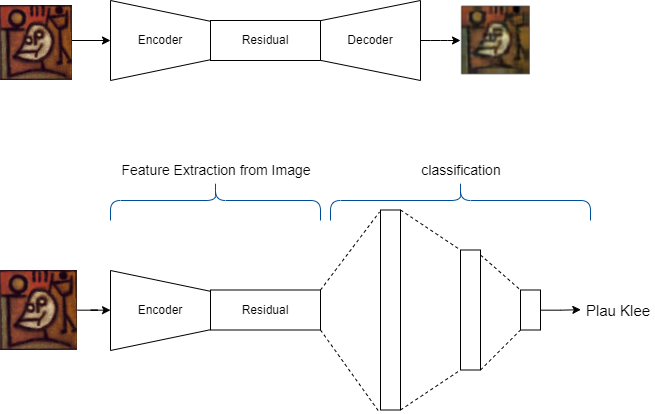
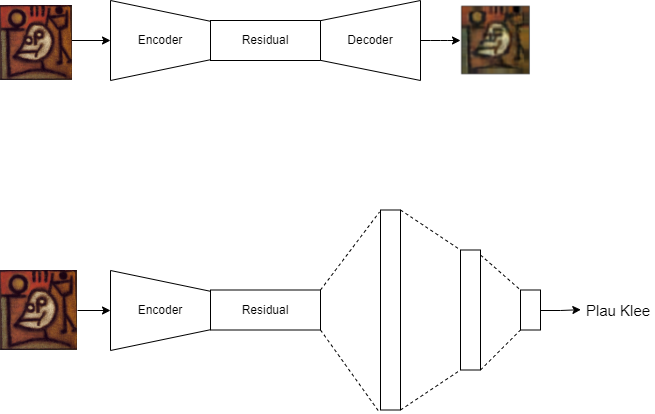
**그림 1. <Andy\_Warhol>**  **<Vincent\_van\_Gogh>**

그래서 과연 Convolution network가 이 둘의 색감이나 스타일의 특징을 잘 추출하여 서로 다른 화가가 그린 그림이라고 분류할 수 있는지가 궁금했다.

1. **기존 연구**
   1. **Nerual style transfer**

Neural style transfer를 관련 연구로 둔 것은 그림 2의 transform net을 이번 프로젝트의 autoencoder model로 두었기 때문이다. 본 neural style transfer는 미리 pretrained된 network에 content image와 style image를 쌍으로 넣어줘서 학습을 통해 style transfer를 하는 방식이다. 해당 그림의 경우는 미리 학습된 style image에 content image간의 loss 값으로 image를 변형시키는 모델로서 작동한다.

1. **프로젝트** 
   1. **Masterpiece classification using residual**



1. **진행 일정**

|  |  |
| --- | --- |
| 진행일자 | 내용 |
| ~11 | 프로젝트 진행에 필요한 논문 수집, 데이터 확보 |
| 11 ~ 14 | 해당 프로젝트에 적합한 모델 생성 |
| 11 ~ 16 | 프로젝트 진행에 있어 논문 코드 분석 |
| 16 ~ 20 | 최종 모델 선정(또는 생성) 및 데이터 학습 |
| 20 ~ | 결론 도출 |

1. **결론**

최종적으로 해당 모델의 학습을 통해 input으로 들어간 masterpiece가 어느 화가의 작품인지를 분류하도록 하였다.

.

**참고 문헌**

**[1]** He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. pages 770-778, 2016.

**[2]** Lianfa Li, Ying Fang, Jun Wu, Jinfeng Wang, et al. "Autoencoder Based Residual Deep Networks for Robust Regression Prediction and Spatiotemporal Estimation" arXiv preprint arXiv:1812.11262v1 (2018).