Universal Style Transfer via Feature Transforms

- Universal Style Transfer via Feature Transforms.
- NIPS 2017.
- https://github.com/Yijunmaverick/UniversalStyleTransfer

Abstract

- Universal style transfer 가 무엇인가?
  - Universal style transfer의 목표는 무작위의 시각적 스타일을 컨텐츠 영상으로 전달하는 것을 목표로 함.
- 기존 방식의 문제점은 무엇인가?
  - 기존 Universal style transfer은 알고리즘의 효율성(메모리, 처리속도, …)에만 집중을 해왔을 뿐 관측하지 않은 스타일 혹은 절충된 시각적 질(compromised visual quality)에 일반화 시키는 능력이 부족하였음.
- 논문의 저자들은 어떻게 문제를 해결했는가?
  - 논문의 저자들은 어떠한 사전에 정의된 스타일에 대한 학습을 하지 않는 점에 대한 (without training on any pre-defined styles) 한계점을 비판하고 있음.

Abstract

- 논문의 주된 시작점(아이디어)는 무엇인가?
  - 논문에서 제안된 방식의 핵심 요소는 a pair of feature transform임.
  - A pair of feature transform은 whitening 과 coloring으로 이를 영상 복원 네트워크에 embedding 시킬 수 있음.
  - Whitening과 coloring 변환은 주어진 스타일 영상에 컨텐츠 영상의 feature의 공분산을 직접 일치시키는 것을 말하며, neural style transfer에서 Gram matrix를 손실함수로 두고 feature의 분포를 일치시키려고 하는 것과 비슷한 맥락임.
    - 그렇다면, 왜 gram matrix를 사용하지 않고 whitening 과 coloring 기법을 활용 하였는지 궁금.
- 논문의 실험 결과와 그 의미는 무엇인가?
  - 고해상도 스타일 영상을 생성해 내는데 효과적임을 입증하였음.
  - 뿐만 아니라, feature coloring 기법을 통해서 texture를 합성하고 whitened된 feature를 시각화 하여 저자들이 제안한 방식을 분석하였음.

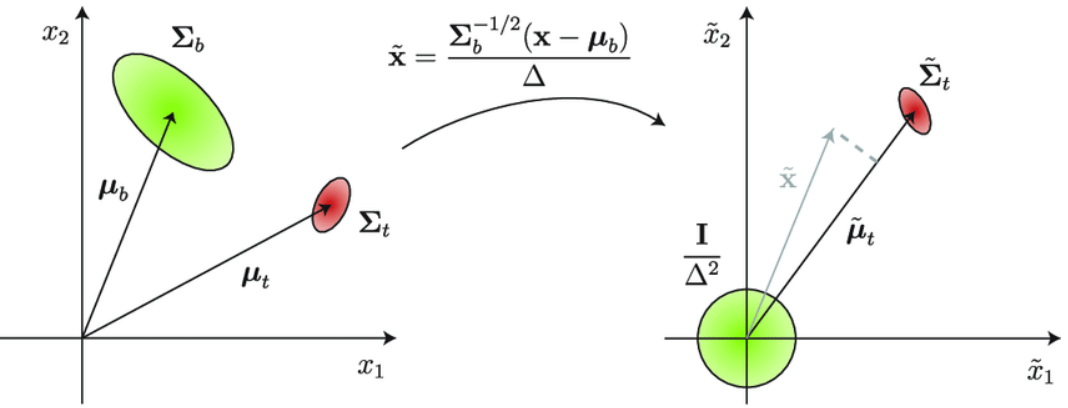
Embedding

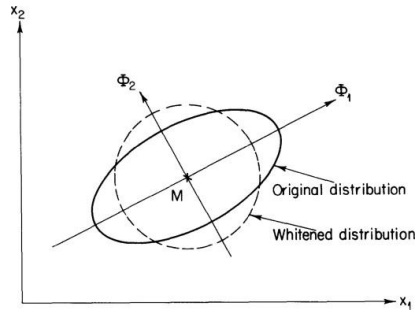
- Embedding이란 일종의 픽셀 값의 집합이라고 볼 수 있는 영상에(세상은 수로 이루어져 있다는 유클리드의 관점에서 보면) 내제된 벡터를(즉, 영상을 표현할 수 있는 벡터) 어떠한 공간(latent space, 특징들이 놓여진 공간, latent space ppt 자료 참고)놓을 수 있고 안에 내제된 특징을 표현하는 벡터를 암호화 시키는 것을 embedding이라고 이해. R차원의 공간으로 사상 시켜 주는 방식을 의미한다고 봄.
- the context of machine learning, an embedding is a low-dimensional, learned continuous vector representation of discrete variables into which you can translate high-dimensional vectors.
- Generally, embeddings make ML models more efficient and easier to work with, and can be used with other models as well.

https://datascience.stackexchange.com/questions/53995/what-does-embedding-mean-in-machine-learning

Whitening from. wiki

- 그렇다면, whitening이란 대체 무엇이길래 논문에서 자주 등장할까?
  - Whitening transform(미백 변환) 혹은 sphering transformation(구형 변환)은 선형변환의 일종으로 알고있는 공분산 행렬을 가지고 있는 랜덤 변수 벡터를 공분산이 항등 행렬인 새로운 변수 집합으로 변환하는 선형 변환으로, 상관관계가 없고 각 분산이 1임을 의미.
  - 위와 같은 변환이 whitening이라고 불리는 이유는 입력 벡터를 백색 잡음 벡터로 변환해 주기 때문.
    - 신호처리에서 백색 잡음의 의미는 다음에 더 자세히 알아보겠음.





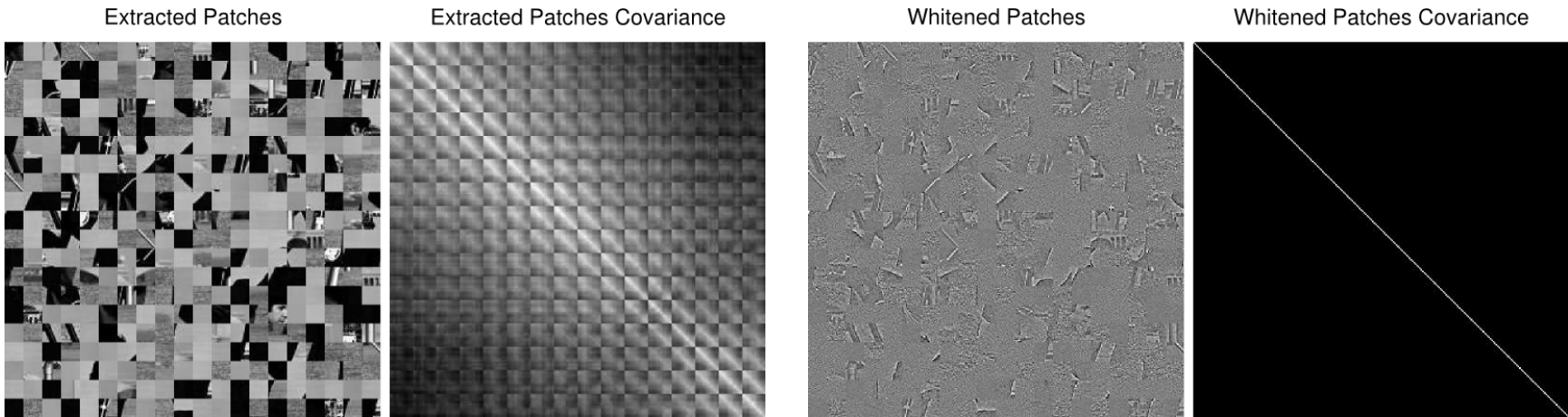
Whitening - 과정

- Whitening transform은 입력 feature를 uncorrelated 하게 만들어 주고, 각 분산을(variance)1로 만들어 줌.
- 과정.(다양한 whitening 기법에서 대표적인 중 하나 인 듯)
  1. 각 sample(feature)에 대한 평균을 0으로 맞춰주어 samples이 중심에 위치하도록 함.
  2. Covariance의 eigenvector matrix를 곱하여 samples을 분산이 가장 큰 축과 일직선에 정렬(align) 되도록(samples을 분산이 가장 큰 축과 일직선에 정렬 시킨다는 의미는 correlation을 0으로 만들어 주는 같음) 회전을 시키면 samples을 decorrelate 시킴.
     - Why? Covariance의 eigenvector matrix를 곱한다는 말이 PCA 과정을 거친다는 말과 같은 과정으로 이해하여 이를 통하여 samples을 분산이 가장 큰 축과 일직선에 정렬(align) 시킬 수 있다는 것을 이해함.
     - Decorrelate 시켜 주면 최적화 과정에서 빠르게 수렴 할 수 있음.
  3. 분산을 1로 만들어 주기 위하여 1보다 큰 분산을 가지는 차원에 대하여 늘리고(stretch) 1보다 작은 분산을 가지는 차원에 대하여 줄여주어(squeeze) scaling을 함.
     - 분산을 1로 만들어 주면 feature가 같은 중요도를 가지도록 한다는 말임.

Whitening - 의미

- 데이터 처리에서 whitening 은 입력 값을 더 쓸모 있게 만드는 작업임.
- 데이터를 neural network 에 친화적으로 만든다고 하기도 함.
  - Why?
- 영상처리에서는 어떠한 의미를 가질까?

- 대표적인 whitening으로는 batch 정규화가 있음.



https://dive-into-ds.tistory.com/19

Introduction

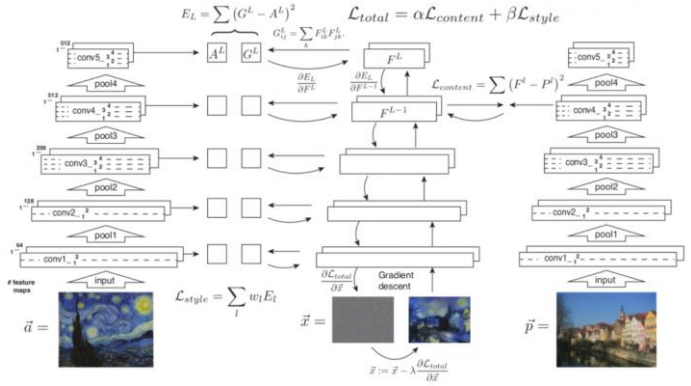
- Style transfer 란 무엇일까?
  - 새로운 예술적 작품(creation of new artistic works)을 컴퓨터가 만들어 내는 image editing task의 일종임.
  - Content와 style 영상이 주어졌을 때, style의 특징을 주입하면서 content의 개념(notion)을 유지하여 영상을 합성하는 것이 목표임.
- Style transfer 의 주된 도전 과제는 무엇일까?
  - 어떻게 style 의 표현(representations)을 효과적으로 추출해 내어 이를 content 영상에 입히는(matching)가에 달려 있음.

Introduction

- 이전 연구로는 어떤 것이 있을까?
  - 딥러닝 기반의 style transfer 기법을 초기에 제시하였던 논문인 Gatys et al.은 깊은 신경망으로 학습된 features 사이에 상관계수를(이때, 상관계수와 공분산의 차이는 별로 없었으며 공분산을 계산하는 것이 더 효율적이기 때문에 공분산을 사용) 활용 하는 것이 영상에서 시각적인 스타일을 다타내는데 효과적임을 입증.
    - 위 논문이 발표된 이후로, feature 사이에 상관계수/공분산을 줄이는 것에 집중하는 연구들이 많이 제안되었음.
- 이전 연구의 문제점은 무엇일까?
  - 영상의 질과(quality) 알고리즘의 효율성(efficiency), 일반화(generalization) 사이에는 trade-off 관계가 존재한다는 문제.
    - 예를 들어서, 무작위의 style 영상에 시각적인 만족을 시키는 최적화 기반 접근 방식을(optimization-based methods) 제안한 연구들은 높은 계산 량이 필요하였음.
      - 영상 자체를 학습시키는 방식을 말하는듯.
    - 반대로, 적은 계산 량 으로 추출 할 수 있는 단 방향 기반 접근 방식을(feed-forward methods) 제안한 연구들은 짧은 실행 시간을 요구하지만 고정된 개수의 style 영상에만 적용이 가능하고 시각적인 만족을 비교적 어렵다는 단점이 있음.
      - 영상 변환 네트워크를 학습시키는 방식을 말하는듯.
  - 지금까지는, 영상의 질과(quality) 알고리즘의 효율성(efficiency), 일반화(generalization)을 동시에 만족시키는 알고리즘이 없었다는 것을 문제점으로 지적.

Style transfer의 관점
pre-trained 된 모델을 기반으로 content와 style을 입력으로 이용해 영상을 학습시키는 방법.

- Content 영상, style 영상에 대하여 합성할 영상을 noise 로 초기화 시킴.
- 각 영상 content, style, 합성할 영상은 신경망을 통하여 feed-forward 함.
- Content 과 합성할 영상의 content loss 계산.
- Style 과 합성할 영상의 style loss 계산.
- Content와 style을 합하여 total loss 계산.
- Total loss의 back-propagation을 통하여 합성할 영상을 업데이트함.
  단, 네트워크 자체는 업데이트되지 않으며, 생성하려는 영상만 업데이트 됨.



https://www.popit.kr/neural-style-transfer-%EB%94%B0%EB%9D%BC%ED%95%98%EA%B8%B0/

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

Style transfer의 관점
pre-trained 된 모델을 기반으로 content와 style을 입력으로 이용해 영상 변환 네트워크를 학습하는 방법.

- 기존의 영상을 학습시키는 방법의 문제점.
  - Pre-trained network에 content와 style 영상을 pair로 넣어주어 새로운 영상을 생성하기 위해서는 매번 학습을 시켜야 한다는 단점이 있음.
  - Content가 바뀔 때마다 학습을 하여야 하기 때문에 시간이 오래 걸리며 연산 량이 많다는 단점이 있음.
- 이를 해결하기 위하여, 영상이 아닌 네트워크를 학습하는 방식을 제안.
  - Network에 style 영상 1장을 학습시키고 그 network를 사용하면 여러 장의 content 영상을 style transfer(inference) 할 때 매번 재 학습 시키진 않고 단순히 inference 함.

https://hoya012.github.io/blog/Fast-Style-Transfer-Tutorial/

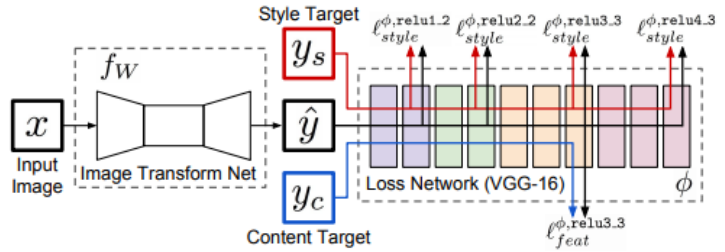
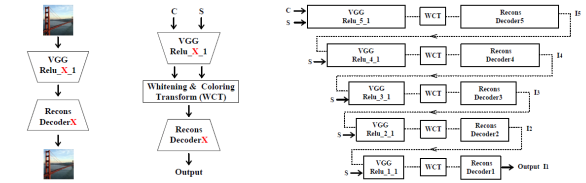


Fig. 2. System overview. We train an *image transformation network* to transform input images into output images. We use a *loss network* pretrained for image classification to define *perceptual loss functions* that measure perceptual differences in content and style between images. The loss network remains fixed during the training process.

Introduction

- 논문의 시작점 즉, 논문에서 주목한 관점은 무엇일까?
  - 논문은 transfer task를 영상을 복원(image reconstruction)하는 프로세스로 공식화 하여 표현 할 수 있다는 점에 주목하였음.
    - Why? 영상을 복원하는 네트워크에서 content 영상을 통과시켜 나온 중간 layer의 features를 Style 영상에서 추출된 features의 통계적인 정보에 맞게 transform(논문에서 transform 이라는 단어에 중점을 두고 있음) style transfer가 가능함.
    - 즉, 중간 layer에서 추출된 content features 가 같은 layer에서 추출된 style features 와 같은 통계적인 특성(여기서 말하는 통계적인 특성을 맞춰 준다는 것이 covariance matrix 를 비슷하게 해 준다는 의미인가?)을 가지도록 변형(transform) 시켜 주는 것이 목표임.
    - 이때, 말하는 transform을 어떠한 방식으로 할 것인가?
      - 저자들은 전통적인 변환 방식인 whitening 과 coloring transform (WCTs) 을 도구로 문제를 풀고 있음(왜 학습하는 방식이 아닌 전통적인 선형 변환 기법을 사용하였는지 궁금).
        - Why? Effortless manner로 style feature의 통계적인 특성을 가지도록 변형 가능하기 때문이라고 언급하고 있음. 즉, 학습하지 않고 빠르게 feed-forward pass로 통계적인 특성을 맞출 수 있는 방식은 hand-crafted 방식이기 때문이라고 이해.

Introduction



(a) Reconstruction (b) Single-level stylization (c) Multi-level stylization

Figure 1: Universal style transfer pipeline. (a) We first pre-train five decoder networks DecoderX (X=1,2,...,5) through image reconstruction to invert different levels of VGG features. (b) With both VGG and DecoderX *fixed*, and given the content image C and style image S, our method performs the style transfer through whitening and coloring transforms. (c) We extend single-level to multi-level stylization in order to match the statistics of the style at all levels. The result obtained by matching higher level statistics of the style is treated as the new content to continue to match lower-level information of the style.

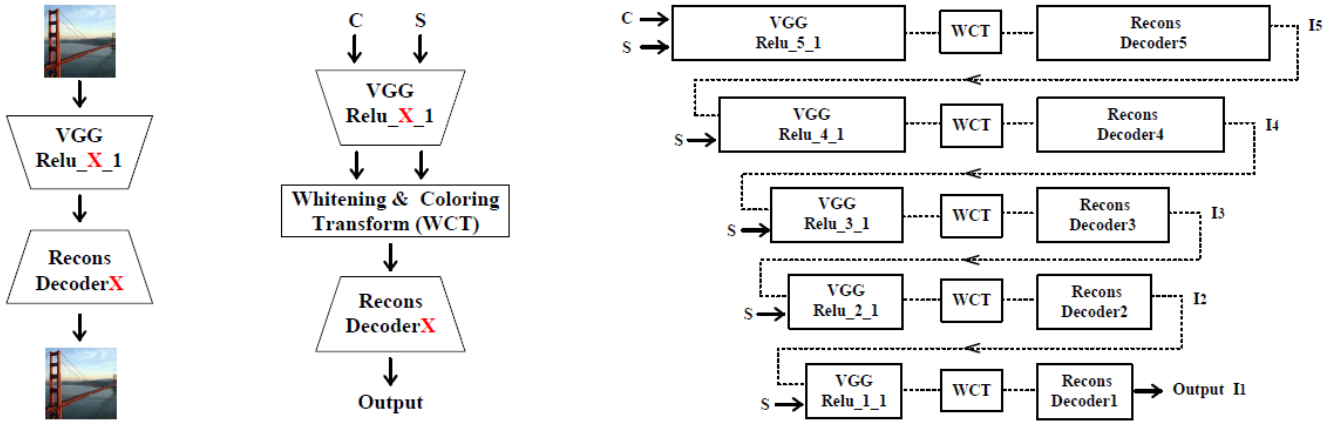
- 논문에서는 목표를 달성 하기 위하여 어떤 방식을 사용하였는가?
  1. VGG-19 network(여기서, pre-trained 된 걸 사용한다는 것인지 학습시킨다는 것인지 잘 모르겠음)을 encode 로 삼아 features를 추출하고 이렇게 인코딩 된 VGG-19 features를 원본 영상으로 변환시켜 주는 decoder를 학습시킴.
     - Encoder – decoder 구조는 영상 복원에 필수적인 구조라고 언급(하지만, SR같이 영상을 복원하는 네트워크 구조에서 반드시 encoder – decoder를 쓰는 것은 아니라고 알고 있는데 영상 복원에서 encoder – decoder가 정확히 어떤 의미를 가지는지 궁금).
     - 일단 학습이 어느정도 되면 , 이렇게 학습된 encoder와 decoder를 고정시켜 놓고 나머지 과정을 수행. Style transfer를 위한 사전 학습 과정이라고 보면 될 것 같음.
  2. 앞에서 훈련시킨 encoder – decoder 를 고정 시킨 채로, Style transfer를 위하여 whitening 과 encoder 에서 추출된 content features의 하나의 layer에 coloring transform (WCTs)을 진행. 이를 통하여 style features의 통계적인 분포인 covariance matrix를 일치 시켜 줄 수 있음.
  3. 이렇게 변형된 features를 가지고 stylized 된 영상을 출력하기 위하여 decode에 feed-forward 시켜 줌.
  4. 2. 과정에서 하나의 layer에 대해서만 WCTs를 거친다고 하였음. 이를 multi-level 에서 확장 시켜 나가기 위하여 multiple feature layers에 WCTs를 연속적으로 적용함.
     - 이와 같은 multi-level 알고리즘으로 인하여 훨씬 적은 계산 량과 동시에 질 좋은 시각적 영상 출력 가능.
     - 뿐만 아니라, style이 transfer되는 정도에 대한 제어를 하기 위하여 style과 content의 균형을 맞추어 주는 control parameter를 제안하였음.

왜 high-level 에서 부터 low-level 로 가는 거지?
Scale 을 갈수록 늘려 주기 위함 인가..?

Introduction

- 논문에서는 목표를 달성 하기 위하여 어떤 방식을 사용하였는가?

5개의 layer에 대하여 일단 모두 학습.



(a) Reconstruction (b) Single-level stylization (c) Multi-level stylization

Figure 1: Universal style transfer pipeline. (a) We first pre-train five decoder networks DecoderX (X=1,2,...,5) through image reconstruction to invert different levels of VGG features. (b) With both VGG and DecoderX *fixed*, and given the content image C and style image S, our method performs the style transfer through whitening and coloring transforms. (c) We extend single-level to multi-level stylization in order to match the statistics of the style at all levels. The result obtained by matching higher level statistics of the style is treated as the new content to continue to match lower-level information of the style.

앞에서 5개의 layer 에 대하여 모두 학습 하였기 때문에 학습된 네트워크를 가지고 WCTs 과정만 진행함.

Introduction

- 논문에서 제안한 방식은 어떤 장점이 있는가?
  - 어떠한 style 영상이 포함되지 않고 (with no style images) 오직 영상을 복원하는 decoder 구조만 학습시켜도 style transfer가 가능하다는 것을 보여주고 있음.
  - 즉, 새로운 스타일의 영상이 주어졌을 때, 해야 할 일은 단순히 feature의 covariance 행렬을 추출하고 contents feature에 whitening 과 coloring transform (WCTs) 을 적용하는 일만 하면 되기 때문에 간단함.
  - 사전에 정의된 style 을 가지고 새로운 style에 대하여 fine-tuning하는 기존 feed-forward 기반 방식과는 학습에서 자유로운 방식임.
  - 즉, 이를 통해서 새로운 style에 tuning하지 않아도 되기 때문에 universal 한 목표를 이루었다고 말 할 수 있음.

Feed-forward

- 논문에서 feed-forward라는 말이 많이 등장하는데 네트워크가 결과를 출력할 뿐 weight를 update하는 학습을 하지 않는다고 이해하였음(일종의 inference 개념).

https://seungheondoh.netlify.app/blog/mlp

Proposed Algorithm

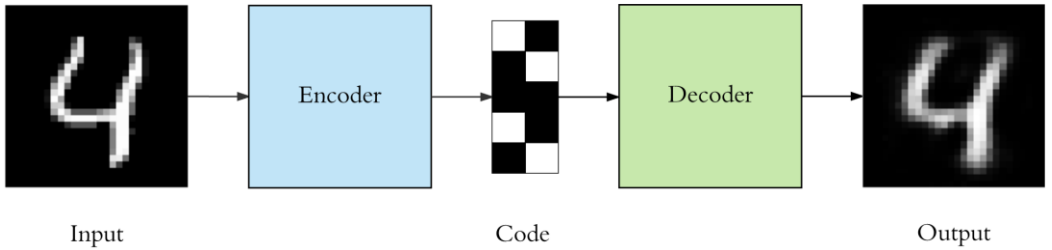
- 논문의 저자들은 style transfer가 whitening 과 coloring 을 거친 즉, feature 의 변형 과정을 거친 image reconstruction 과정의 일부라는 점에 주목하였음.
  - Why? style transfer 와 image reconstruction 는 공통적으로 RGB 영상을 features 로 되돌리는 encoding 과정이 필요하고 다시 이 정보를 가지고 RGB 영상으로 decoding 시키기 때문. 이때 , image reconstruction 와 다른 부분은 encoding 된 content features 를 style features 의 통계적인 분포와 유사해지도록 하는 것이 필요함.
  - 이것의 근본적인 근거는 일반적인 영상 복원을 위한 auto-encoder 구조를 구축하였다는 점에서 시작 되었음.

Proposed Algorithm

- 논문의 저자들은 style transfer가 whitening 과 coloring 을 거친 즉, feature 의 변형 과정을 거친 image reconstruction 과정의 일부라는 점에 주목하였음.
  - Why? style transfer 와 image reconstruction 는 공통적으로 RGB 영상을 features 로 되돌리는 encoding 과정이 필요하고 다시 이 정보를 가지고 RGB 영상으로 decoding 시키기 때문. 이때 , image reconstruction 와 다른 부분은 encoding 된 content features 를 style features 의 통계적인 분포와 유사해지도록 하는 것이 필요함.
  - 이것의 근본적인 근거는 일반적인 영상 복원을 위한 auto-encoder 구조를 구축하였다는 점에서 시작 되었음.

Autoencoder from. Wiki and 외국블로그

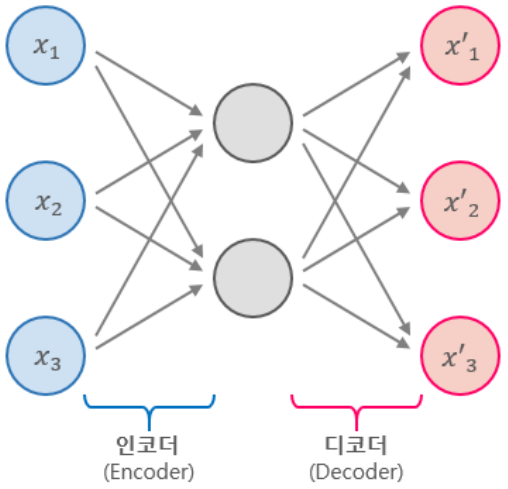
- 오토인코더(Autoencoder)는 unsupervised 방식으로 효율적인 데이터 코딩을 학습하는 데 사용되는 인공 신경망의 일종임.
  - 보통, 오토인코더를 unsupervised learning 기법 이라고 하지만 정확하게 말하면 명확하게 말하면 self-supervised 기법의 일종임. 왜냐하면 학습 데이터로부터 자신의 label를 만들기 때문임.
- 오토인코더(Autoencoder)의 목표는 신호의 잡음(noise)을 무시하도록 네트워크를 훈련시킴으로써 일반적인 차원 축소를 위한 일련의 데이터에 대한 표현을 배우는 것임.
- 입력과 출력이 같은 feedforward neural network의 일종으로 더 낮은 차원으로(lower-dimensional code)로 압축하는 것을 representation으로 compression 하는 과정과 representation으로 부터 출력물을 reconstruction 하는 과정으로 이루어져 있음.
- 이렇게 입력 으로 부터 compression 된 representation 을 code, latent-space representation 이라고 부름.



https://en.wikipedia.org/wiki/Autoencoder#cite_note-:3-32
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Autoencoder – 개념

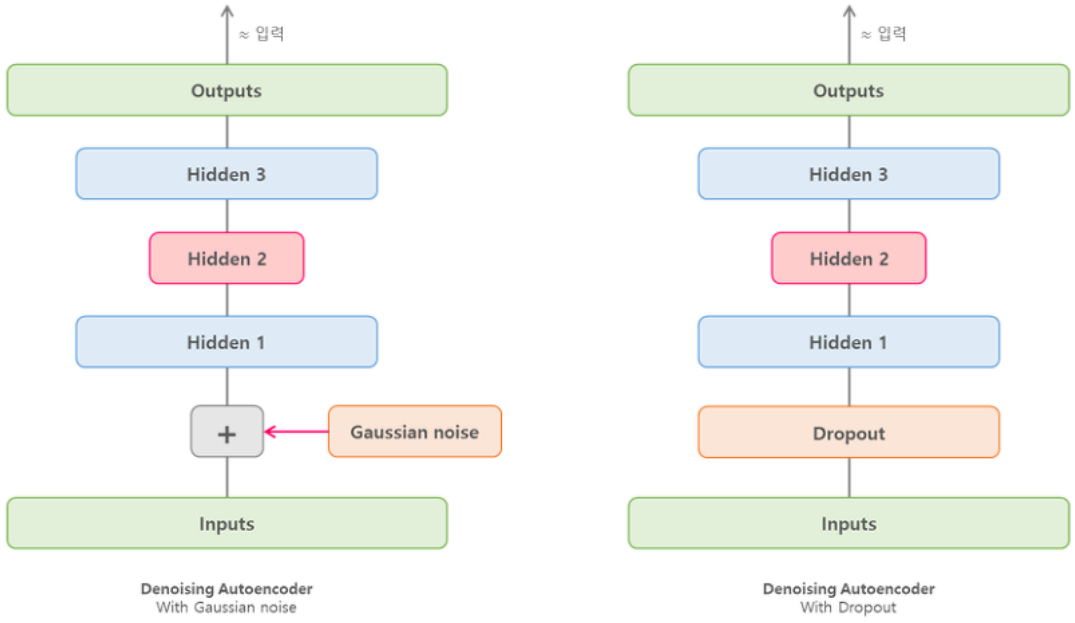
- 오토인코더(Autoencoder) 란 무엇인가?
  - 데이터를 효율적으로 coding(여기서, coding 이란 데이터를 압축하는 것을 의미함) 하기 위한 딥러닝 구조의 일종임.
  - 즉, 데이터를 효율적으로 나타내기 위하여 고차원을 저차원으로 차원 축소하는 방법.
  - 입력과 출력이 같은 신경망으로 간단해 보이지만 여러가지 방식으로 제약을 걸어줌으로써 이 제약들은 단순히 입력을 바로 출력하지 못하도록 함으로써 데이터를 효율적으로 표현(representation) 하는 방법을 학습하도록 함.



https://excelsior-cjh.tistory.com/187
블로그에 잘 설명되어 있으니까 많이 참고 하면 좋을 듯.

Autoencoder – denoising Autoencoder

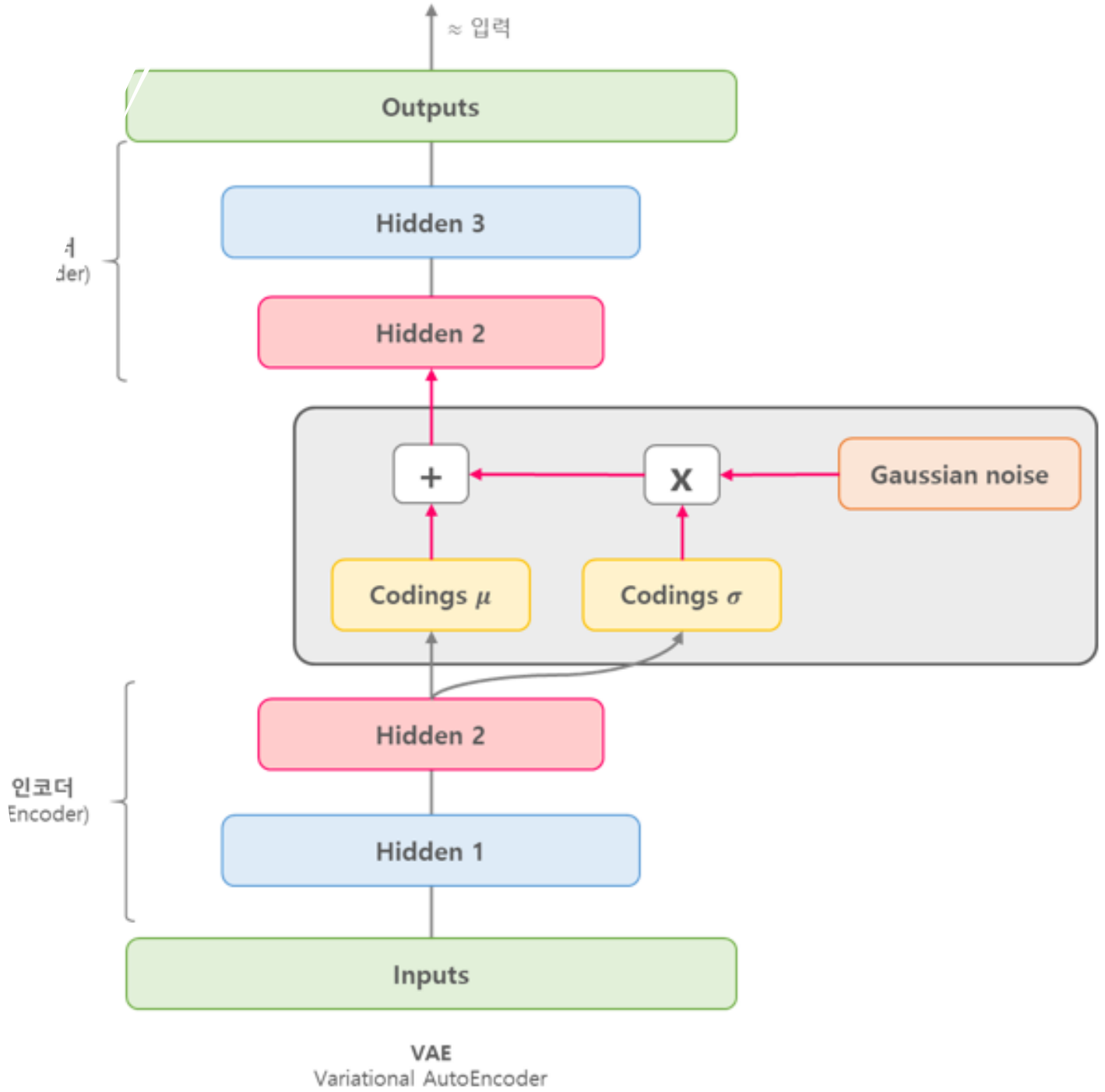
- Autoencoder 가 의미 있는 특징(feature)를 학습하도록 제약을 주는 방법 중 하나로 입력에 noise(잡음)를 추가하는 방식이 있음.
- Noise(잡음)가 추가된 입력에서 잡음이 없는 원본 입력을 재구성하도록 학습.
- 잡음을 발생시키는 방식은 그림과 같이 원본 입력에 가우시안(Gaussian) 노이즈를 추가하거나 혹은 드롭아웃(dropout) 처럼 랜덤하게 입력 유닛(node)를 꺼서 발생 시킬 수 있음.
  - Why? 드롭아웃 시키는 것이 노이즈를 발생 시킬 수 있을까?
    - https://www.researchgate.net/post/What-is-the-difference-between-dropout-method-and-adding-noise-in-case-of-autoencoder
- 이것의 효과는 무엇일까?



https://excelsior-cjh.tistory.com/187
블로그에 잘 설명되어 있으니까 많이 참고 하면 좋을 듯.

Autoencoder – Variational Autoencoder

- 논문에서 제안한 오토인코더(VAE)는 기존 오토인코더와 다음과 같은 차이점이 있음.
  - VAE는 확률적 오토인코더임.
    - 학습이 종료된 이후에도 출력이 확률적으로 결정됨.
  - VAE는 생성 오토인코더임.
    - 학습 데이터셋에서 샘플링 된 것과 같이 새로운 샘플을 생성할 수 있음.
  - 좀 더 깊은 이해가 필요할 듯.



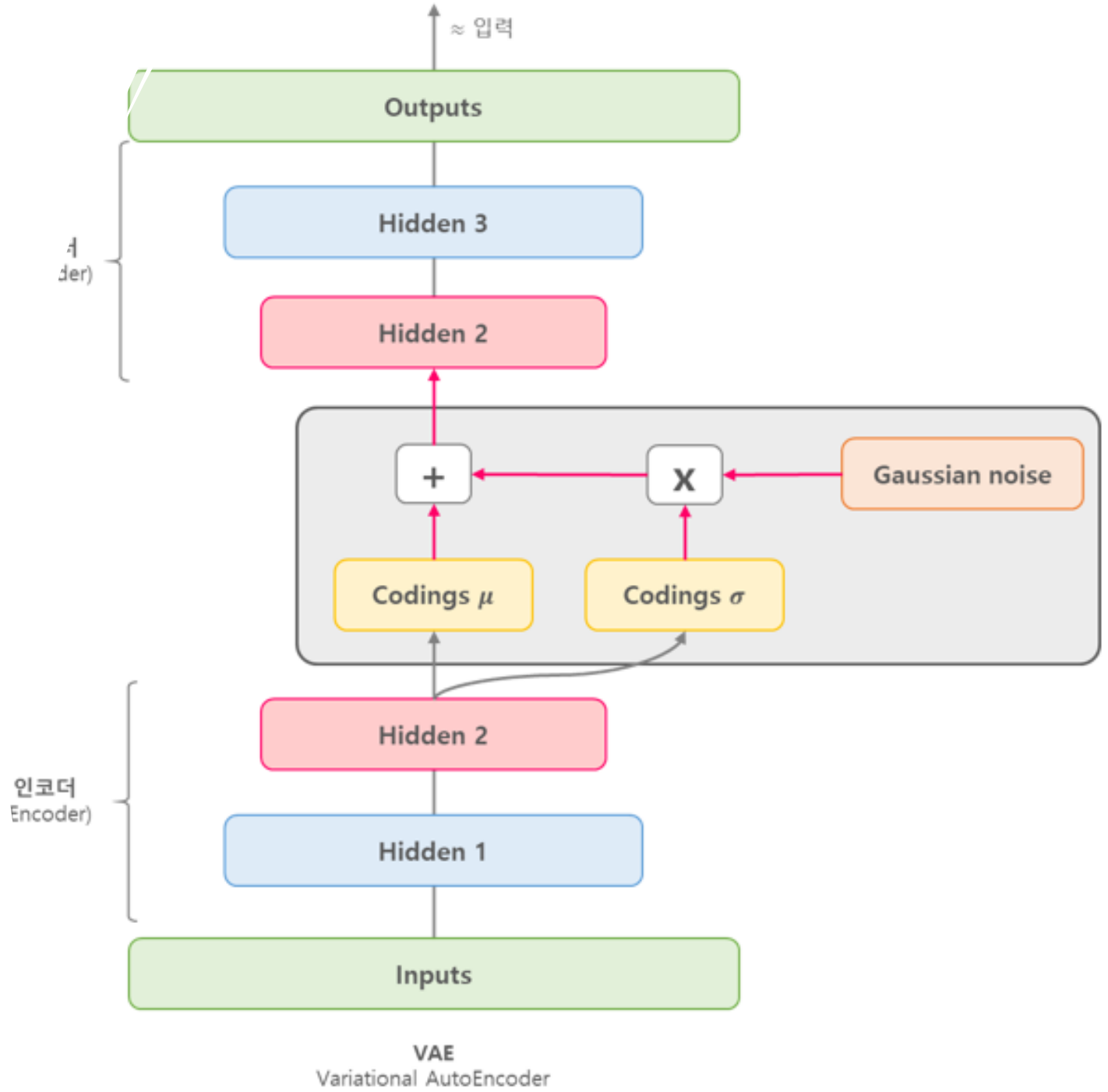
VAE
Variational AutoEncoder

https://arxiv.org/pdf/1312.6114v10.pdf

https://excelsior-cjh.tistory.com/187

Autoencoder – Variational Autoencoder

- Why?
  - 주어진 입력에 대하여 바로 coding을 생성하는 것이 아니라 encoder는 평균 coding, μ과 표준편차 coding, σ 을 생성함.
    - 실제로, 구현할 때는 평균이 μ이고 표준편차가 σ인 가우시안(gaussian) 분포에서 랜덤하게 샘플링 되며, 이렇게 랜덤하게 샘플링된 coding을 디코더(decoder)가 원본 입력으로 재 구성하게 만듦.
    - 이렇게 하는 이유는 마치 가우시안 분포에서 샘플링된 것처럼 coding을 만들어서, 학습하는 동안 손실 함수가 coding을 가우시안 샘플들의 집합처럼 보이는 형태를 가지는 latent space로 이동시키기 때문.
    - 즉, 이러한 이유로 VAE는 학습이 종료된 이후에도 새로운 샘플을 가우시안 분포로 부터 랜덤한 coding을 샘플링 할 수 있게 되고 이렇게 랜덤한 coding으로 부터 디코딩 하여 새로운 영상을 만들어 낼 수 있게 됨.



VAE
Variational AutoEncoder

https://arxiv.org/pdf/1312.6114v10.pdf

https://excelsior-cjh.tistory.com/187

Proposed Algorithm – reconstruction

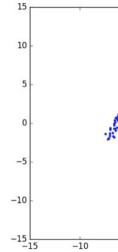
- 엔코더
  - 이를 위하여 사전에 학습된 VGG-19 구조를 엔코더로 사용하였고 이에 대한 학습은 하지 않고 파라미터를 고정.
- 디코더
  - 원본 영상으로 VGG features을 원본 영상으로 다시 복원을 해 주기 위하여 디코더를 학습.
  - 구조는 VGG-19 네트워크와 대칭적인 구조로 구성.
    - 단, feature maps을 업 샘플링 할 때 nearest-neighborhood 방식 사용.
- 손실 함수
  - Euclidean distance.
    - $\|I_0 - I_i\|^2$.
  - Perceptual distance.
    - $\|\varphi(I_0) - \varphi(I_i)\|^2$.
    - φ는 사전에 학습된 VGG 엔코더에서 추출된 feature map을 의미.
- 이렇게 학습을 진행 한 후에 style transfer 과정에서는 디코더는 고정된 채 feature를 invert시키는 용도로 사용.

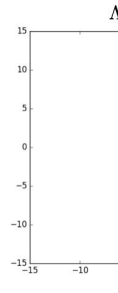
Proposed Algorithm – WCTs

- Whitening 과 coloring 변환(WCTs)를 하기 전에 content, f_c 와 style, f_s 영상에서 VGG encoder 특정 layer에서 추출된 feature map을 준비함.
  - 이때, f_c가 transform을 거치지 않는 다면 원본 입력 영상과 같은 content 영상이 출력될 것임.
- WCTs의 목표는 f_c가 f_s의 통계적인 분포(covariance matrix)를 따르도록 직접 변환해 주는 것임.

Proposed Algorithm – Whitening

- f_c를 uncorrelated 하고 분산이 1이 되도록 whitening 하는 것이 목표.
- 앞에서 언급된 과정과 같이 공분산의 eigenvector matrix를 곱하여 줌.
- 새롭게 얻어진 원본 f_c를 $\hat{f}_c = E_C D_C^{1/2} E_C^T f_c$ 을 PCA whitening 을 거침.
  - D_C 는 입력 feature, f_c는 공분산 행렬인 $f_c f_c^T$ 의 eigenvalues로 구성된 diagonal matrix 임.
  - E_C는 eigenvector의 orthogonal matrix 임.
  - 이때, $f_c f_c^T = E_C D_C E_C^T$를 만족함.
  - $\hat{f}_c \hat{f_c^T}$는 identity matrix로 decorrelated 됨.
  - 궁금증.
    - $\hat{f}_c = E_C D_C^{1/2} E_C^T f_c$ 과정이 SVD(singular value decomposition) 과정과 비슷한데 무슨 관련이 있는 것일까?
    - $f_c f_c^T = E_C D_C E_C^T$는 어떻게 성립하는 것일까?
      - $\hat{f_c^T} \hat{f}_c = f_c^T E_C D_C^{1/2} E_C^T E_C D_C^{1/2} E_C^T f_c = f_c^T E_C D_C E_C^T f_c = \mathrm{I}$
      - $E_C D_C E_C^T = f_c f_c^T$





Proposed Algorithm – Whitening

- 분석.
  - 영상의 contents의 global 한 구조를 유지한 채로 styles과 관계된 정보를 제거해 주는데 매우 도움이 됨.
  - 일종의 새 옷을 입히기 위하여 옷을 벗긴 것으로 비유.

Figure 2: Inverting whitened features. We invert the whitened VGG Relu_4_1 feature as an example. Left: original images, Right: inverted results (pixel intensities are rescaled for better visualization). The whitened features still maintain global content structures.

Proposed Algorithm – Whitening

- f_c를 uncorrelated 하고 분산이 1이 되도록 whitening 하는 것이 목표.
- 앞에서 언급된 과정과 같이 공분산의 eigenvector matrix를 곱하여 줌.
- 새롭게 얻어진 원본 f_c를 $\hat{f}_c = E_C D_C^{-1/2} E_C^T f_c$ 을 PCA whitening 을 거침.
  - D_C 는 입력 feature(f_c)의 공분산 행렬인 $f_c f_c^T$ 의 eigenvalues로 구성된 diagonal matrix 임.
  - E_C는 eigenvector의 orthogonal matrix 임.
  - 이때, $f_c f_c^T = E_C D_C E_C^T$를 만족함.
  - $\hat{f}_c \hat{f}_c^T$는 identity matrix로 decorrelated 됨.
  - 궁금증.
    - $\hat{f}_c = E_C D_C^{-1/2} E_C^T f_c$ 과정이 SVD(singular value decomposition) 과정과 비슷한데 무슨 관련이 있는 것일까?
    - $f_c f_c^T = E_C D_C E_C^T$는 어떻게 성립하는 것일까?
      - $\hat{f}_c^T \hat{f}_c = f_c^T E_C D_C^{1/2} E_C^T E_C D_C^{1/2} E_C^T f_c = f_c^T E_C D_C E_C^T f_c = \mathrm{I}$
      - $E_C D_C E_C^T = f_c f_c^T$

즉, 공분산 행렬의 eigenvector 의 방향으로 내적을 수행하는 PCA 과정임.

https://withkairos.wordpress.com/2015/06/13/ufldl-tutorial-8-whitening/
https://stats.stackexchange.com/questions/117427/what-is-the-difference-between-zca-whitening-and-pca-whitening

Content image whitening code

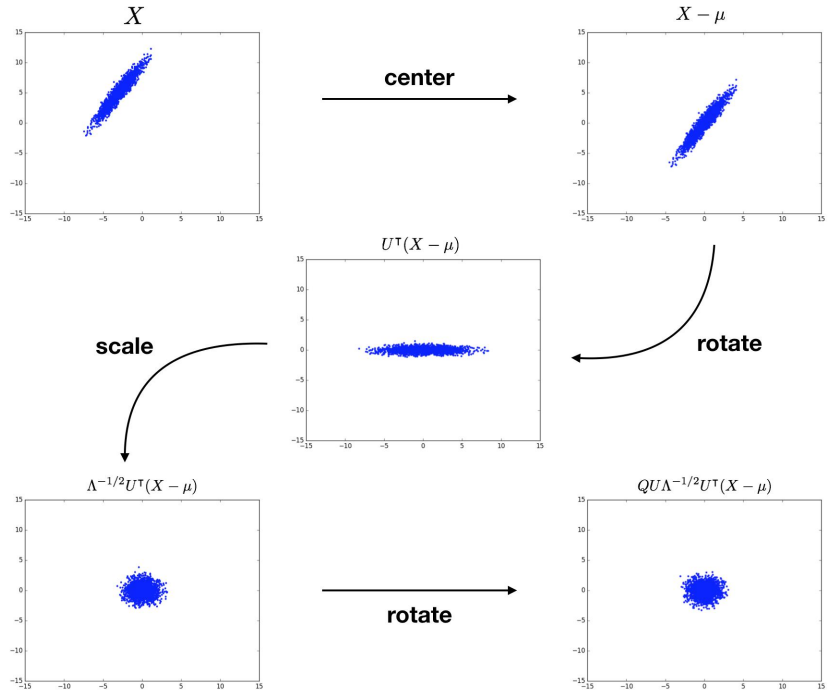
```
def whiten_and_color(self,cF,sF):
    cFSize = cF.size()
    c_mean = torch.mean(cF,1) # c x (h x w)
    c_mean = c_mean.unsqueeze(1).expand_as(cF)
    cF = cF - c_mean

    c_u,c_e,c_v = torch.svd(contentConv,some=False)
```

Whitening 과 SVD 의 관계

- Whitening 은 일종의 PCA whitening 과정임.
- 고유 벡터는 변환의 주축이라고 할 수 있음.
- 새로운 축을 찾기 위해는 고유 분해(EVD, SVD) 를 수행해야 함.

Proposed Algorithm – Whitening

- f_c를 uncorrelated 하고 분산이 1이 되도록 whitening 하는 것이 목표.
- 앞에서 언급된 과정과 같이 공분산의 eigenvector matrix를 곱하여 줌.
- 새롭게 얻어진 원본 f_c를 $\hat{f}_c = E_C D_C^{-1/2} E_C^T f_c$ 을 PCA whitening 을 거침.
  - D_C 는 입력 feature(f_c)의 공분산 행렬인 $f_c f_c^T$ 의 eigenvalues로 구성된 diagonal matrix 임.
  - E_C는 eigenvector의 orthogonal matrix 임.
  - 이때, $f_c f_c^T = E_C D_C E_C^T$를 만족함.
  - $\hat{f}_c \hat{f_c^T}$는 identity matrix로 decorrelated 됨.



https://withkairos.wordpress.com/2015/06/13/ufldl-tutorial-8-whitening/
https://stats.stackexchange.com/questions/117427/what-is-the-difference-between-zca-whitening-and-pca-whitening

Eigenvector Animation

- https://m.blog.naver.com/PostView.nhn?blogId=angryking&logNo=221206754322&proxyReferer=https:%2F%2Fwww.google.com%2F

Proposed Algorithm – Coloring

f_s 는 style 영상에서 VGG encoder 특정 layer에서 추출된 feature map 임.

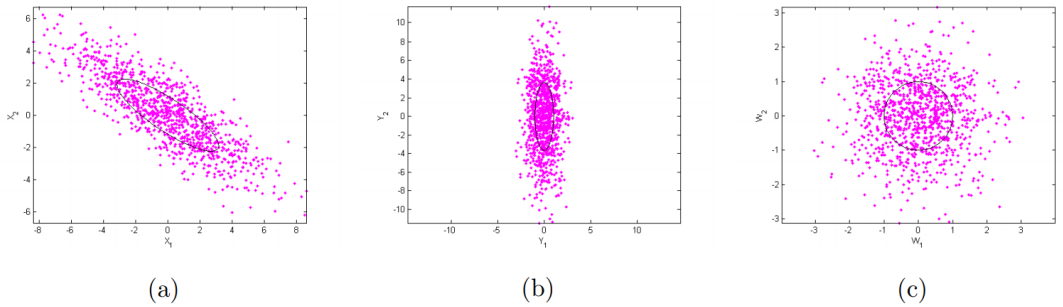
- 이제, content 정보가 제거된 $\widehat{f_c}$ 에 새로운 style 정보를 입혀 줌.
  - f_s의 통계적인 분포(covariance matrix) 를 $\widehat{f_c}$ 에 맞춰 주기 위하여 새롭게 style 정보를 입혀준 $\widehat{f_{cs}}$ 가 $\widehat{f_{cs}}\widehat{f_{cs}}^T = f_s f_s^T$ 를 만족하도록 해야 함.
- How?
  - Whitening 변형 과정을 반대로 하면 됨.
  - $\widehat{f_{cs}}\widehat{f_{cs}}^T = f_s f_s^T = E_s D_s E_s^T$ (뒤에 식은 만족하는지 잘 모르겠음).
  - $\widehat{f_{cs}} = E_s D_s^{1/2} E_s^T \widehat{f_c}$
    - D_s 는 입력 feature, f_c의 공분산 행렬인 $f_c f_c^T$ 의 eigenvalues로 구성된 diagonal matrix 임.
    - E_C는 eigenvector의 orthogonal matrix 임.
    - 이때, $f_c f_c^T = E_C D_C E_C^T$를 만족함.

Proposed Algorithm – Coloring

- 이제, content 정보가 제거된 $\hat{f_c}$ 에 새로운 style 정보를 입혀 줌.
  - f_s의 통계적인 분포(covariance matrix) 를 $\hat{f_c}$ 에 맞춰 주기 위하여 새롭게 style 정보를 입혀준 $\widehat{f_{cs}}$ 가 $\widehat{f_{cs}}\widehat{f_{cs}}^T = f_s f_s^T$ 를 만족하도록 해야 함.
- How?
  - Whitening 변형 과정을 반대로 하면 됨.
  - $\widehat{f_{cs}}\widehat{f_{cs}}^T = f_s f_s^T = E_s D_s E_s^T$ (뒤에 식은 만족하는지 잘 모르겠음).
  - $\widehat{f_{cs}} = E_s D_s^{1/2} E_s^T \hat{f_c}$
    - D_s 는 입력 feature, f_c의 공분산 행렬인 $f_c f_c^T$ 의 eigenvalues로 구성된 diagonal matrix 임.
    - E_C는 eigenvector의 orthogonal matrix 임.
    - 이때, $f_c f_c^T = E_C D_C E_C^T$를 만족함.
  - 최종적인 출력 $\widehat{f_{cs}}$ 은 style features의 평균인 m_s 로 중심이 맞춰 질 것임.

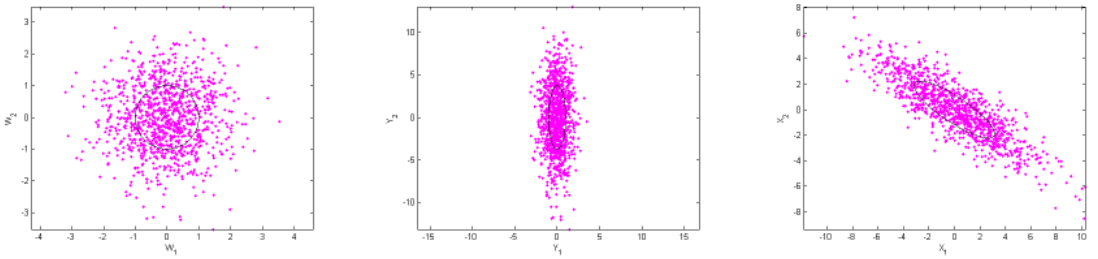
Proposed Algorithm – Coloring

- Whitening 변형 과정을 반대로 하면 coloring 변환 과정이 되는 이유.
  - Whitening 을 feature 가 고르게 분포하도록 만드는 과정이라고 이해하였음.
    - Whitening 을 통하여 feature 를 고르게 분포 시키는 이유는 style 이 통계적인 정보를 담고 있고 이러한 통계적인 정보를 제거하여 content 정보만 남기는 것이 목표이기 때문.
  - 즉, whitening 과정을 반대로 거치면 통계적인 정보를 입혀주는 것이(coloring 변환을 통해서 다른 옷을 입혀 준다고 비유) 가능하기 때문에 coloring을 whitening의 반대 과정이 된다고 이해.



(a) (b) (c)

Figure 1: Scatter plots and Contours. All contours are at Mahalanobis distance 1. (a) Original colored density of $\mathbf{X}$, (b) decorrelated density of $\mathbf{Y}$, (c) whitenend density of $\mathbf{W}$.

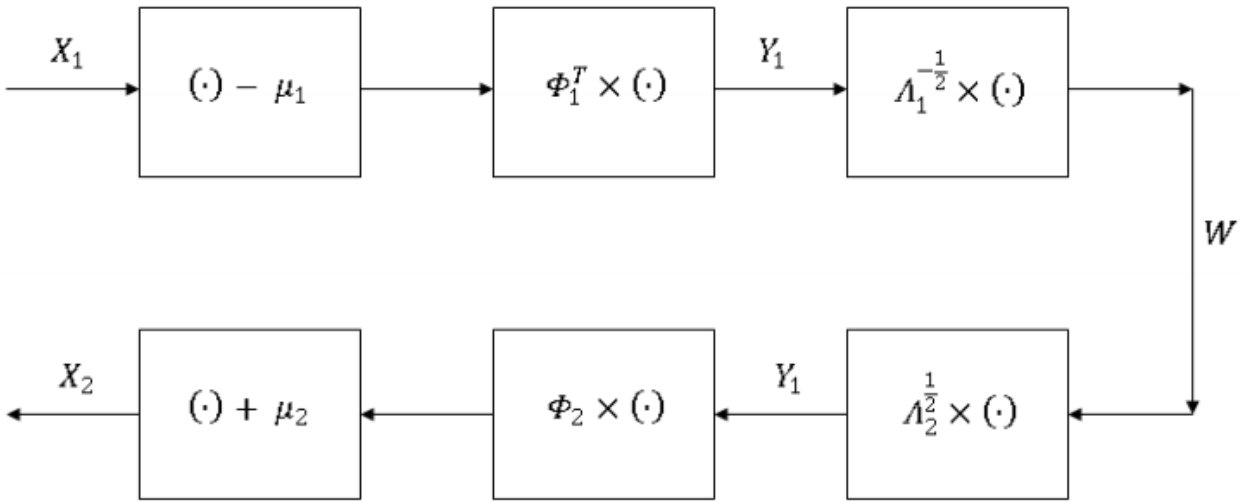


(a) (b) (c)

Figure 2: Scatter plots and Contours. All contours are at Mahalanobis distance 1. (a) Original white density of $\mathbf{W}$, (b) scaled density of $\mathbf{Y}$, (c) colored density of $\mathbf{X}$.

M. Hossain. Whitening and coloring transforms for multivariate gaussian random variables. Project Rhea, 2016.
https://www.projectrhea.org/rhea/images/1/15/Slecture_ECE662_Whitening_and_Coloring_Transforms_S14_MH.pdf

Proposed Algorithm – WCTs



$$X_1 \sim N(\mu_1, \Sigma_1) \qquad \Sigma_1 = \Phi_1 \Lambda_1$$

$$X_2 \sim N(\mu_2, \Sigma_2) \qquad \Sigma_2 = \Phi_2 \Lambda_2$$

$$W \sim N(\mathbf{0}, \mathbf{I})$$

Figure 3: Summary diagram of whitening and coloring process

Proposed Algorithm – WCTs

- WCTs의 효과를 확인하기 위하여 RGB 레벨 영상 차원에서의 처리 기법인 histogram matching(HM) 방식과 비교하였음.
  - 논문에서 비교한 channel-wise histogram matching 방식은 content 를 style 이 가진 누적 히스토그램(cumulative histogram) 으로 맞춰주는 방식임.
  - Style 영상의 전반적인 색깔을 잘 일치시켜 주지만 미세한 시각적 패턴과 부분적인 구조를 잘 표현하지 못한다는 단점이 있는 HM 기법에 비하여 WCTs 방식은 style 영상을 더 잘 표현하고 있음.
    - Why? Style feature의 channel 간의 correlation을 고려하지 않고 잇기 때문으로 영상의 style을 추출하기 위해서는 feature의 channel 간의 correlation 을 고려가 중요하다는 것을 방증하고 있음.

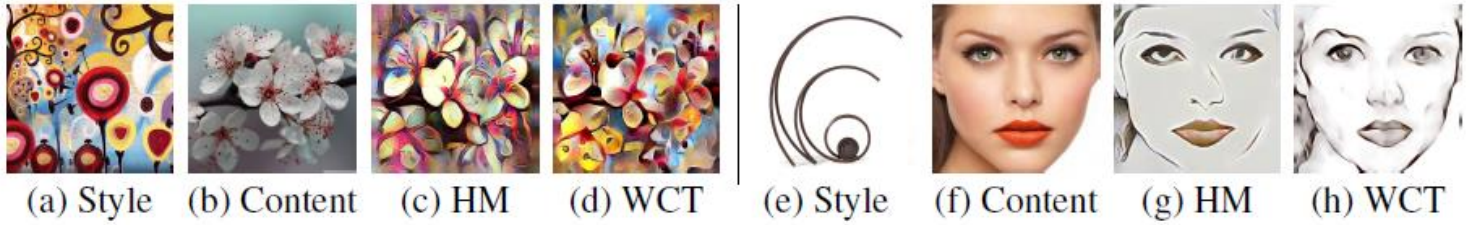


Figure 3: Comparisons between different feature transform strategies. Results are obtained by our multi-level stylization framework in order to match all levels of information of the style.

Proposed Algorithm – WCTs

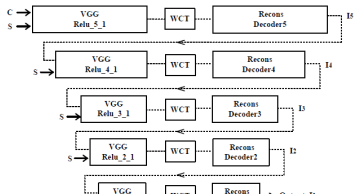
- WCTs 기법을 활용하여 style 과 content의 비율을 직접 맞춰 줄 수도 있음.
  - How?
    - α라는 parameter 를 통해서 style과 content term 간에 weight 의 합이 1이 되도록 함.
      - $\widehat{f_{cs}}=\alpha\ \widehat{f_{cs}}+(1-\alpha)\ \hat{f}_c$
        - 여기서, $\widehat{f_{cs}}$ 는 디코더에 입력으로 α를 통해서 style 을 얼만큼 나타낼지 조절하는 것이 가능함.

Proposed Algorithm – multi-level coarse to fine stylization

- 이전까지 한 style transfer 과정은 single-layer 에 대한 stylization framework 였음.
- 하지만, VGG-19 네트워크의 서로 다른 layers 에서 기능하는 고유의 역할이 있기 때문에 더욱 더 다양한 layer 를 활용하는 것이 효과적임.
  - 더 높은 layer 에서 출력된 feature 는 더 복잡한 부분적인 구조를(complicated local structures) 표현 하는데 특화되어 있고 더 낮은 layer 에서 출력된 feature 는 low-level 특징들을 표현 하는데 특화되어 있음.
- 이를 해결하기 위하여, 논문에서는 점진적인 stylization 방식인 multi-scale pipeline 을 제안 하였음.

Proposed Algorithm – multi-level coarse to fine stylization

- How?
  1. High-level 정보인 Relu_5_1 layer 에서 추출된 feature 에 WCTs 를 적용시킴.
  2. 대략적인 stylized 결과를 얻어 낼 수 있고 이것이 후속될 low-level 정보와 결합할 때 새로운 content 영상의 역할을 하며 입력됨.
  - 논문에서는 이 과정을 coarse to fine 하는 과정이라고 언급하고 있음.
- 왜 논문의 저자들은 high-level features 를 coarse 정보로 보았고 low-level features 를 fine 한 정보로 보았을까?
  - 왜냐하면, low-level features은 edge, color 와 같은 디테일 한 정보에 집중하여 학습하는 반면, high-level features 은 의미론 적인 전체적인 정보에 (salient patterns of the style) 집중하여 보는 경향이 있기 때문임.



(c) Multi-level stylization

진행하고 있는 연구에 대한 고찰과 영감

- 왜냐하면, low-level features은 edge 와 같은 디테일 한 정보에 집중하여 학습하는 반면, high-level features 은 의미론 적인 전체적인 정보에 (salient patterns of the style) 집중하여 보는 경향이 있기 때문임.
  - 이 구절을 보고 지금 연구하고 있는 망고넷에 대한 의구심이 들었음. 망고넷에서는 low-level features 을 네트워크의 전반 단계에서 뽑고 (비록, decoder 에서는 나중에 넣어주고 있지만) high-level features 를 네트워크의 후반 단계 에서 뽑는데 (비록, decoder 에서는 일찍 합쳐주고 있지만) 이것에 대한 고찰도 필요하다고 느꼈음.
    - 영상 복원 네트워크 에서 엔코더의 초반 단계와 후반 단계가 어떠한 차이점이 있는지 (무엇을 중점적으로 보는지) 디코더의 초반 단계와 후반 단계가 어떠한 차이점이 있는지 (무엇을 중점적으로 보는지) 를 리서치 필요.

1. 점진적으로 합쳐 주는 방식을 취하면 어떨까?
   - Why?
2. Stack 하는 방식을 취하면 어떨까?
   - Why?

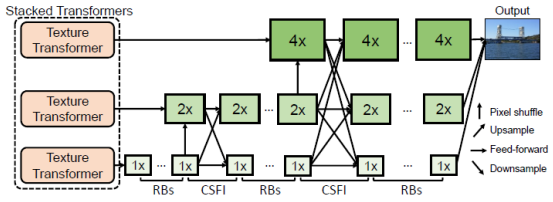
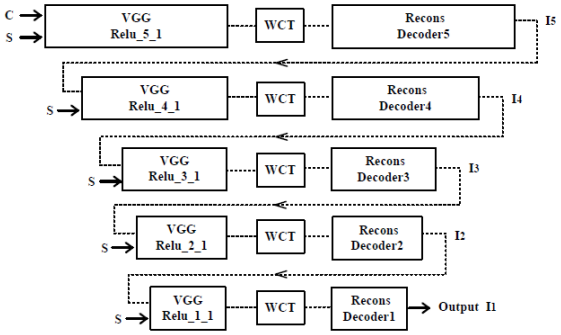


Figure 3. Architecture of stacking multiple texture transformers in a cross-scale way with the proposed cross-scale feature integration module (CSFI). RBs indicates a group of residual blocks.



(c) Multi-level stylization

Proposed Algorithm – multi-level coarse to fine stylization

- 효과.
  - high-level 특징들을 먼저 보고 나서 low-level 특징들을 나중에 보는 것이 효과적임을 알 수 있었음.
    - Style transfer 는 영상의 edge와 color 같은 low-level 정보를 살려 내는 것이 목표이기 때문에 low-level 정보에 집중하는 features 를 네트워크의 후반부에 보는 것이 효과적 이라고 이해. 초반에는 salient patterns of style 과 같은 high-level 정보를 잡아내고 후반부로 갈수록 디테일한 정보를 살리면서 low-level 정보인 디테일한 정보를 살려 내고 있는 것을 실험에서 알 수 있었음 (아래 그림 참고).

(a) I_5 (b) I_4 (c) I_1 (d) Fine-to-coarse

Figure 5: (a)-(c) Intermediate results of our coarse-to-fine multi-level stylization framework in Figure 1(c). The style and content images are from Figure 4. I_1 is the final output of our multi-level pipeline. (d) Reversed fine-to-coarse multi-level pipeline.

논문 요약 및 고찰

- 논문의 핵심과 이런 제안을 할 수 있었던 배경

- 논문의 새로운 점
  - 기존의 content loss 와 style loss 를 사용하여 style transfer 네트워크를 학습시키는 방식이 아닌 image reconstruction auto-encoder 기법으로 영상을 잘 representation 하는 coding 을 추출하고 나서 WCTs 라는 hand-crafted 방식으로 feed-forward 로 style transfer 문제를 해결하였다는 점임.
- 논문의 문제점

- 논문의 개선 방향
  - 후속 논문인 Wang et al. “Collaborative Distillation for Ultra-Resolution Universal Style Transfer”, CVPR2020 에서는 논문에서 제안된 방식이 style transfer 과정에서 decoder가 직접적으로 참여하지 않기 때문에 encoder와 decode의 collaborative 관계를 활용하지 못하였고 이러한 collaborative 관계를 가지는 특징을 고찰하여 효율적인 style transfer 방식을 제안 할 수 있었음.
    - 이러한 통찰력은 어디서 온 것일까?