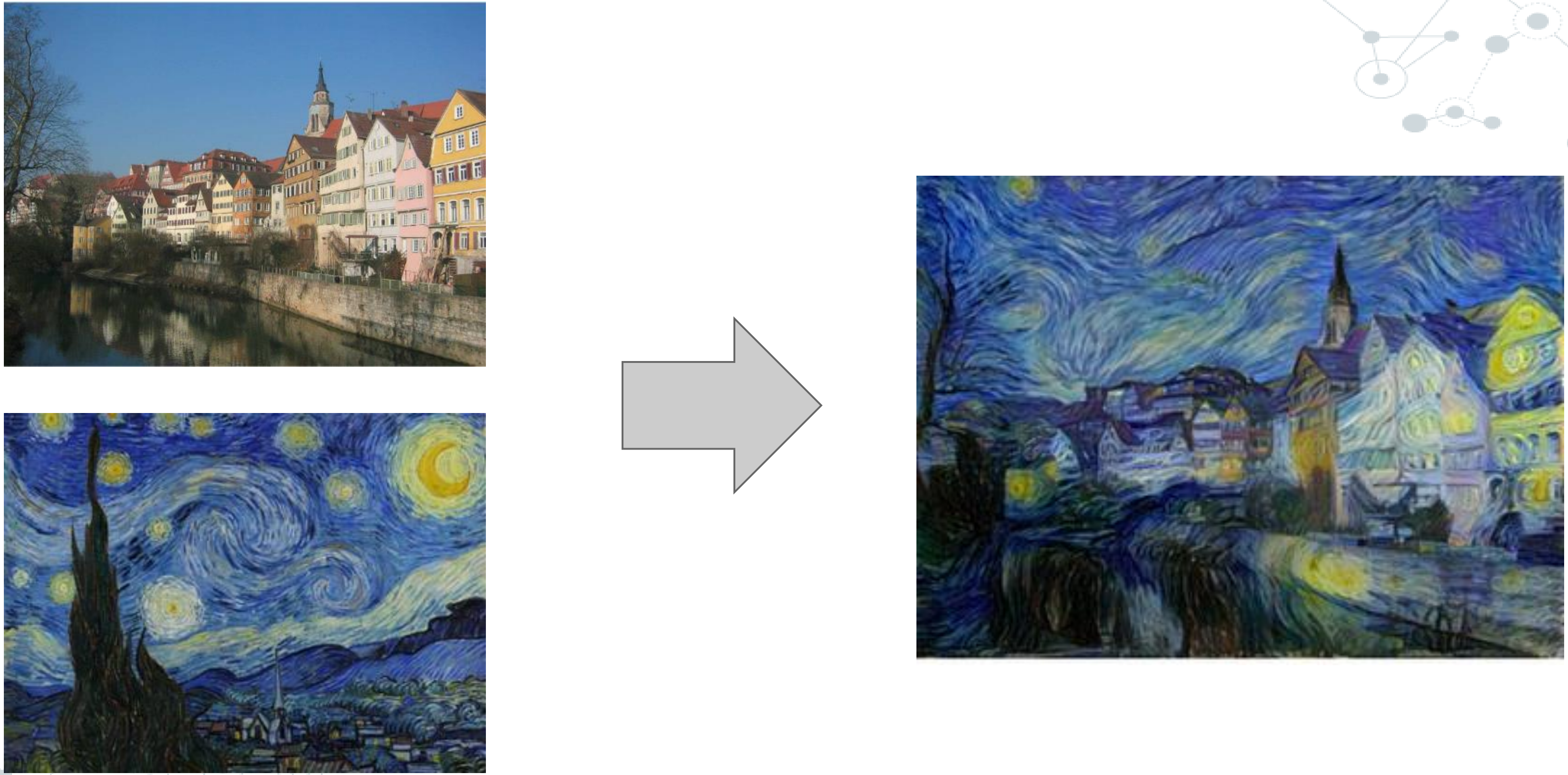
Neural Style

정승환

2018.07.07

Style Transfer이란..



https://jvns.ca/blog/2017/02/12/neural-style/

Style Transfer이란..

이미지 형태는 유지

그림의 Style만을 Copy

https://jvns.ca/blog/2017/02/12/neural-style/

A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3]* Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

[1]Werner Reichardt Centre for Integrative Neuroscience
and Institute of Theoretical Physics, University of Tübingen, Germany
[2]Bernstein Center for Computational Neuroscience, Tübingen, Germany
[3]Graduate School for Neural Information Processing, Tübingen, Germany
[4]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[5]Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA
*To whom correspondence should be addressed; E-mail: leon.gatys@bethgelab.org

In fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. However, in other key areas of visual perception such as object and face recognition near-human performance was recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks.[1,2] Here we introduce an artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality. The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images. Moreover, in light of the striking similarities between performance-optimised artificial neural networks and biological vision,[3–7] our work offers a path forward to an algorithmic understanding of how humans create and perceive artistic imagery.

Image Style Transfer Using Convolutional Neural Networks

Leon A. Gatys
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Graduate School of Neural Information Processing, University of Tübingen, Germany
leon.gatys@bethgelab.org

Alexander S. Ecker
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Baylor College of Medicine, Houston, TX, USA

Matthias Bethge
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract

Rendering the semantic content of an image in different styles is a difficult image processing task. Arguably, a major limiting factor for previous approaches has been the lack of image representations that explicitly represent semantic information and, thus, allow to separate image content from style. Here we use image representations derived from Convolutional Neural Networks optimised for object recognition, which make high level image information explicit. We introduce A Neural Algorithm of Artistic Style *that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of numerous well-known artworks. Our results provide new insights into the deep image representations learned by Convolutional Neural Networks and demonstrate their potential for high level image synthesis and manipulation.*

1. Introduction

Transferring the style from one image onto another can be considered a problem of texture transfer. In texture transfer the goal is to synthesise a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. For texture synthesis there exist a large range of powerful non-parametric algorithms that can synthesise photorealistic natural textures by resampling the pixels of a given source texture [7, 30, 8, 20]. Most previous texture transfer algorithms rely on these non-parametric methods for texture synthesis while using different ways to preserve the structure of the target image. For instance, Efros and Freeman introduce a correspondence map that includes features of the target image such as image intensity to constrain the texture synthesis procedure [8]. Hertzman et al. use image analogies to transfer the texture from an already stylised image onto a target image[13]. Ashikhmin focuses on transferring the high-frequency texture information while preserving the coarse scale of the target image [1]. Lee et al. improve this algorithm by additionally informing the texture transfer with edge orientation information [22].

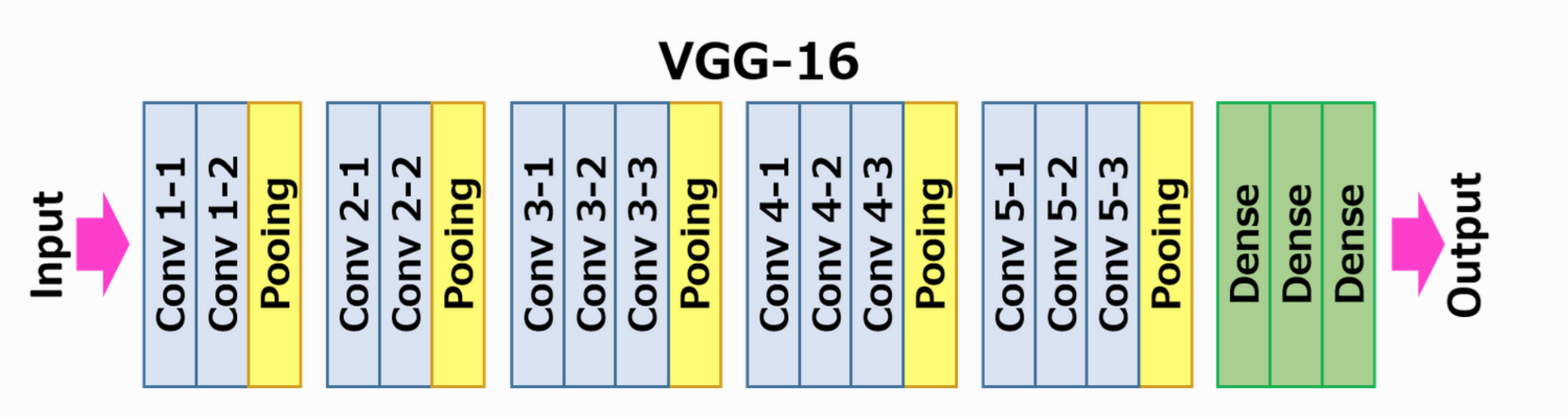
Although these algorithms achieve remarkable results, they all suffer from the same fundamental limitation: they use only low-level image features of the target image to inform the texture transfer. Ideally, however, a style transfer algorithm should be able to extract the semantic image content from the target image (e.g. the objects and the general scenery) and then inform a texture transfer procedure to render the semantic content of the target image in the style of the source image. Therefore, a fundamental prerequisite is to find image representations that independently model variations in the semantic image content and the style in which

Style Transfer이란..

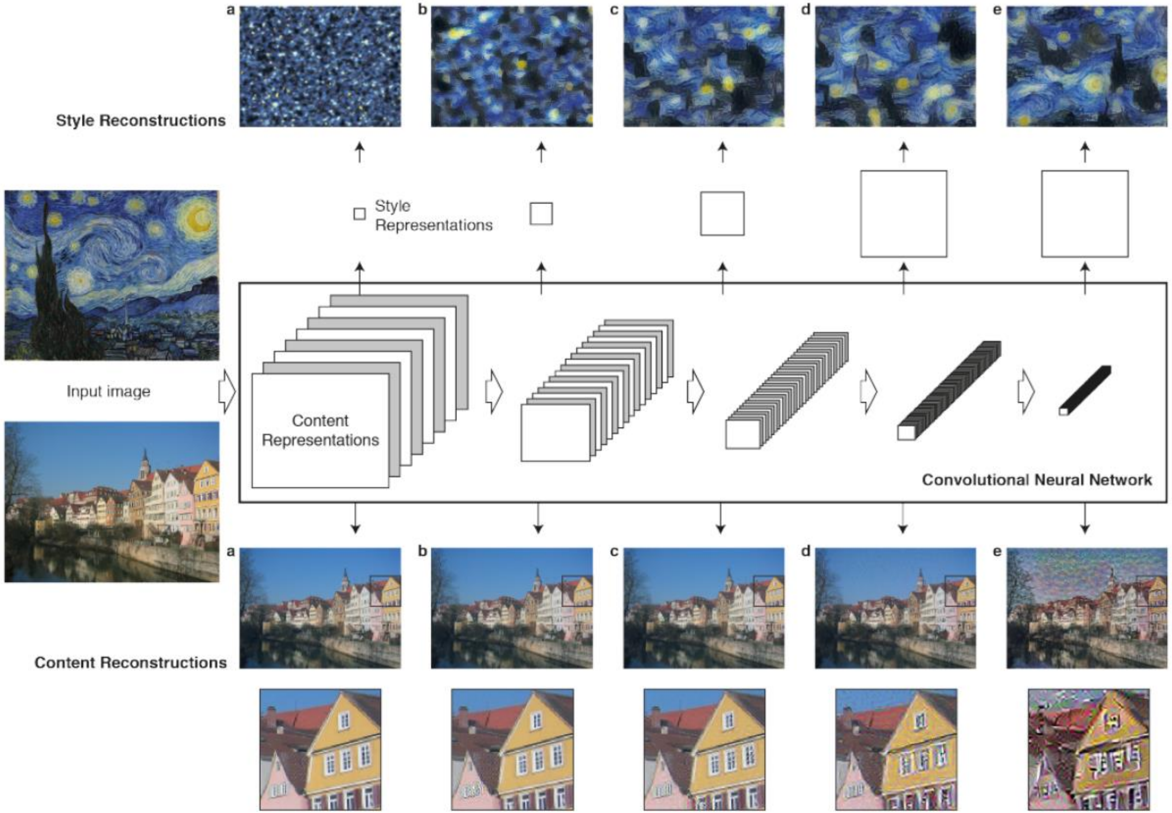
- Style Transfer 에서 중요한 점은 원본의 이미지 형태는 유지(Content loss)하면서 우리가 원하는 이미지의 Style만을 복사(Style loss)할 수 있는 방안이다.

- 본 논문에서는 VGG network(VGG16)를 활용해서 Style/Content loss를 정의함.

VGG 16


VGG-16
Input
Conv 1-1
Conv 1-2
Pooing
Conv 2-1
Conv 2-2
Pooing
Conv 3-1
Conv 3-2
Conv 3-3
Pooing
Conv 4-1
Conv 4-2
Conv 4-3
Pooing
Conv 5-1
Conv 5-2
Conv 5-3
Pooing
Dense
Dense
Dense
Output

CNN에서의 이미지 표현



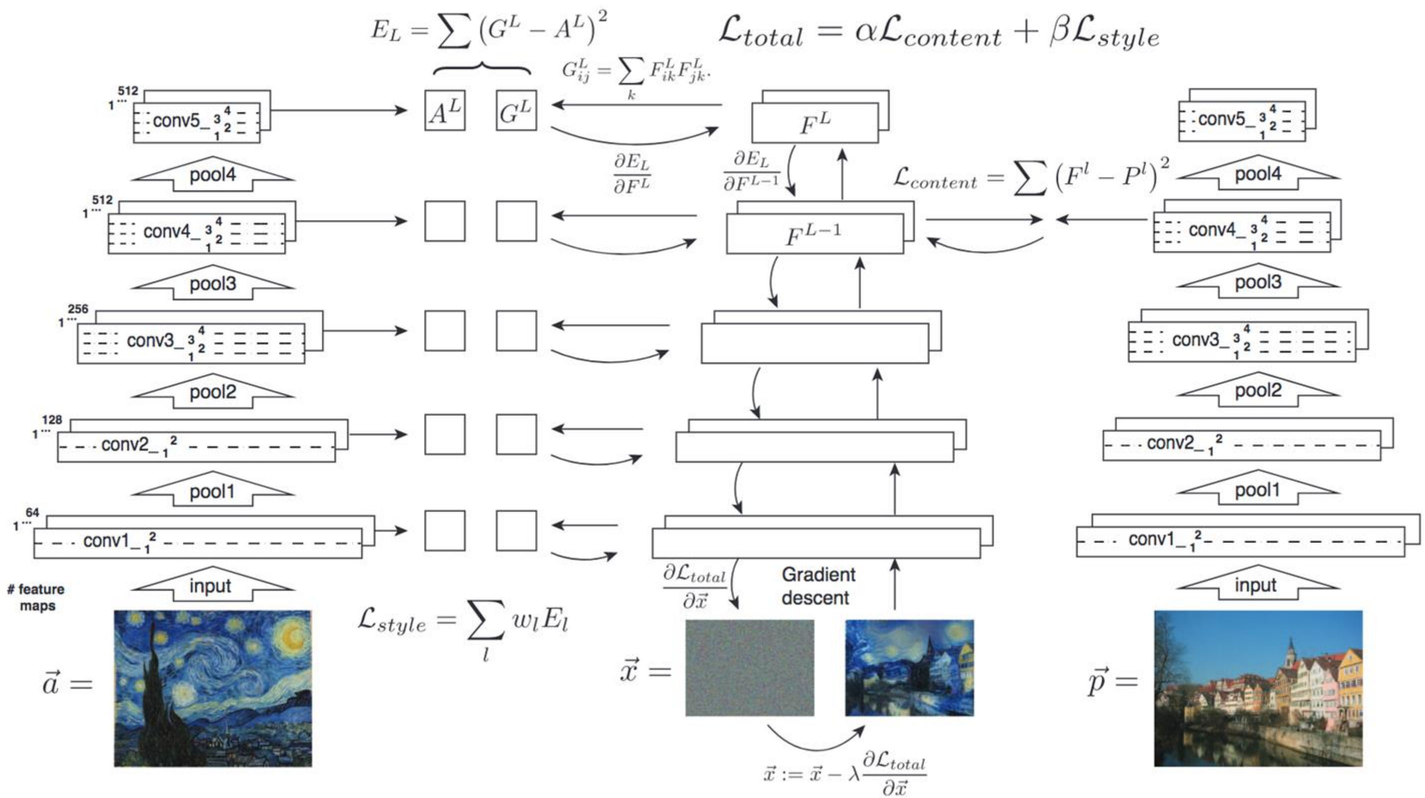
Style Reconstruction

a) Conv1_1
b) a) + Conv2_1
c) b) + Conv3_1
d) c) + Conv4_1
e) e) + Conv5_1

Content Reconstruction

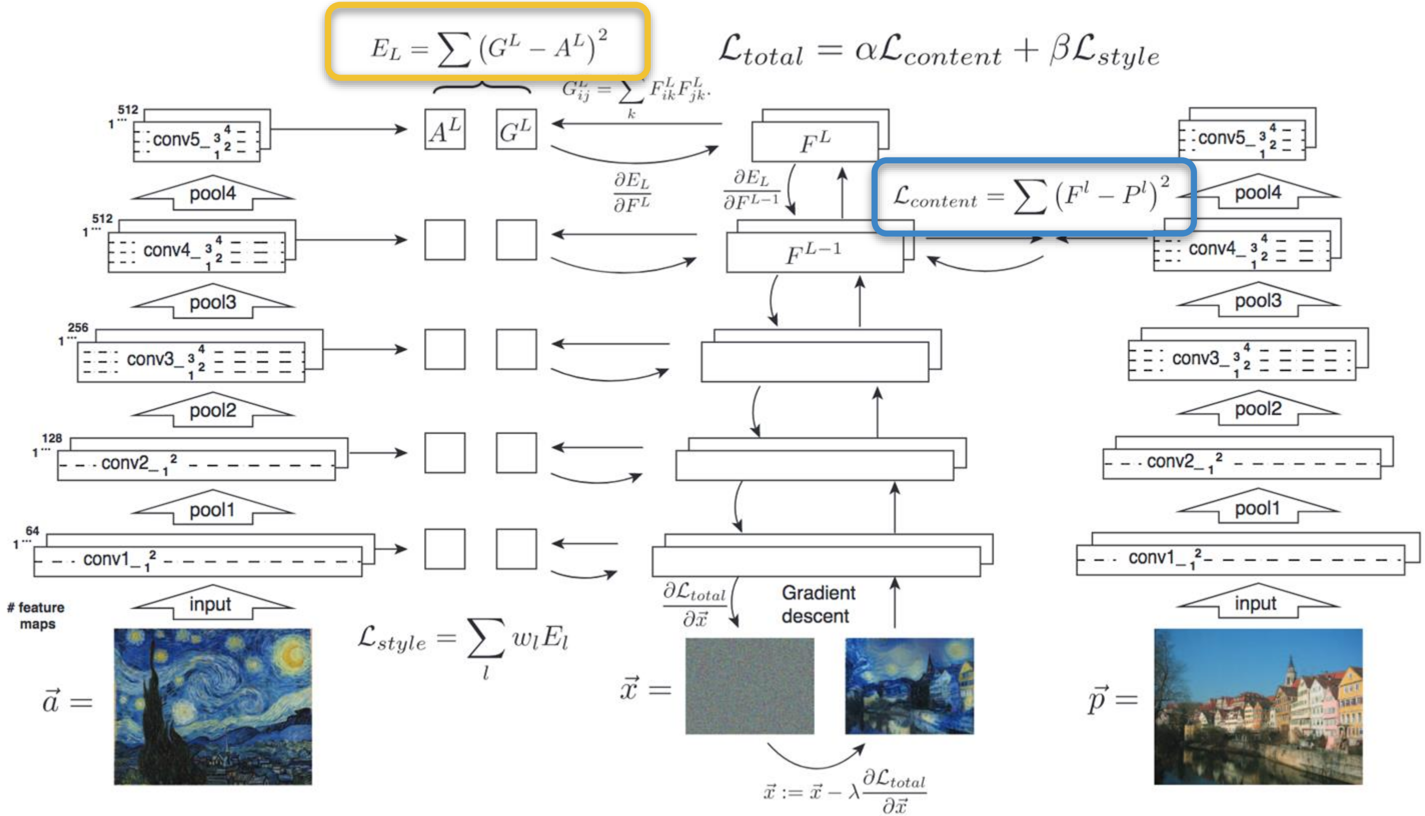
a) Conv1_1
b) Conv2_1
c) Conv3_1
d) Conv4_1
e) Conv5_1

Loss 정의



$E_L = \sum (G^L - A^L)^2$
$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$
$G^L_{ij} = \sum_k F^L_{ik}F^L_{jk}.$
A^L
G^L
F^L
F^{L-1}
$\frac{\partial E_L}{\partial F^L}$
$\frac{\partial E_L}{\partial F^{L-1}}$
$\mathcal{L}_{content} = \sum (F^l - P^l)^2$
512
conv5_
512
conv4_
256
conv3_
128
conv2_
64
conv1_
pool4
pool3
pool2
pool1
input
feature maps
$\vec{a} =$
$\mathcal{L}_{style} = \sum_l w_l E_l$
$\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
Gradient descent
$\vec{x} =$
$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
$\vec{p} =$

Loss 정의



$E_L = \sum (G^L - A^L)^2$
$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$
$G^L_{ij} = \sum_k F^L_{ik} F^L_{jk}.$
$\mathcal{L}_{content} = \sum (F^l - P^l)^2$
$\frac{\partial E_L}{\partial F^L}$
$\frac{\partial E_L}{\partial F^{L-1}}$
A^L
G^L
F^L
F^{L-1}
conv5_1 conv5_2 conv5_3 conv5_4
pool4
conv4_1 conv4_2 conv4_3 conv4_4
pool3
conv3_1 conv3_2 conv3_3 conv3_4
pool2
conv2_1 conv2_2
pool1
conv1_1 conv1_2
input
512
256
128
64
feature maps
$\mathcal{L}_{style} = \sum_l w_l E_l$
$\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
Gradient descent
$\vec{a} =$
$\vec{x} =$
$\vec{p} =$
$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$

Loss 정의

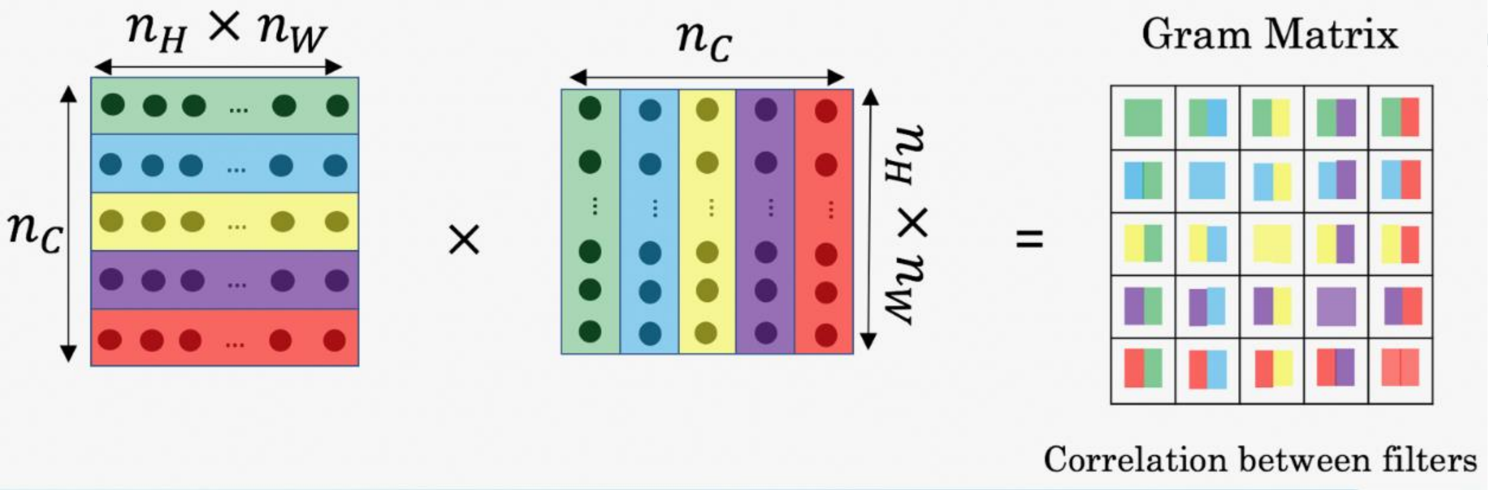
Content loss

- MES (Mean Square Error)을 통해 loss를 산출
- Conv5_1을 통과한 content 이미지(P^l)와 Style 학습을 수행하는 이미지(F^l)간 비교 수행

Style loss

- Gram Matrix를 통해 loss를 산출
- Conv5_1까지 누적으로 통과한 content 이미지(P^l)와 Style 학습을 수행하는 이미지(F^l)간 비교 수행

Gram Matrix




```python
def gram_matrix(y):
    (b, ch, h, w) = y.shape
    features = y.reshape((b, ch, w * h))
    #features_t = F.SwapAxis(features,1, 2)
    gram = F.batch_dot(features, features, transpose_b=True) / (ch * h * w)
    return gram
```

Thanks!

Any questions?