Style Transfer

TNT 24-1 CV
Min Seo Lee
2024/05/07

TRAIN AND TEST

Content

1. **Michigan – Deep Learning for Computer Vision**

   : Visualizing and Understanding

2. **Image Style Transfer Using Convolution Neural Network**

TRAIN AND TEST

Lecture Review

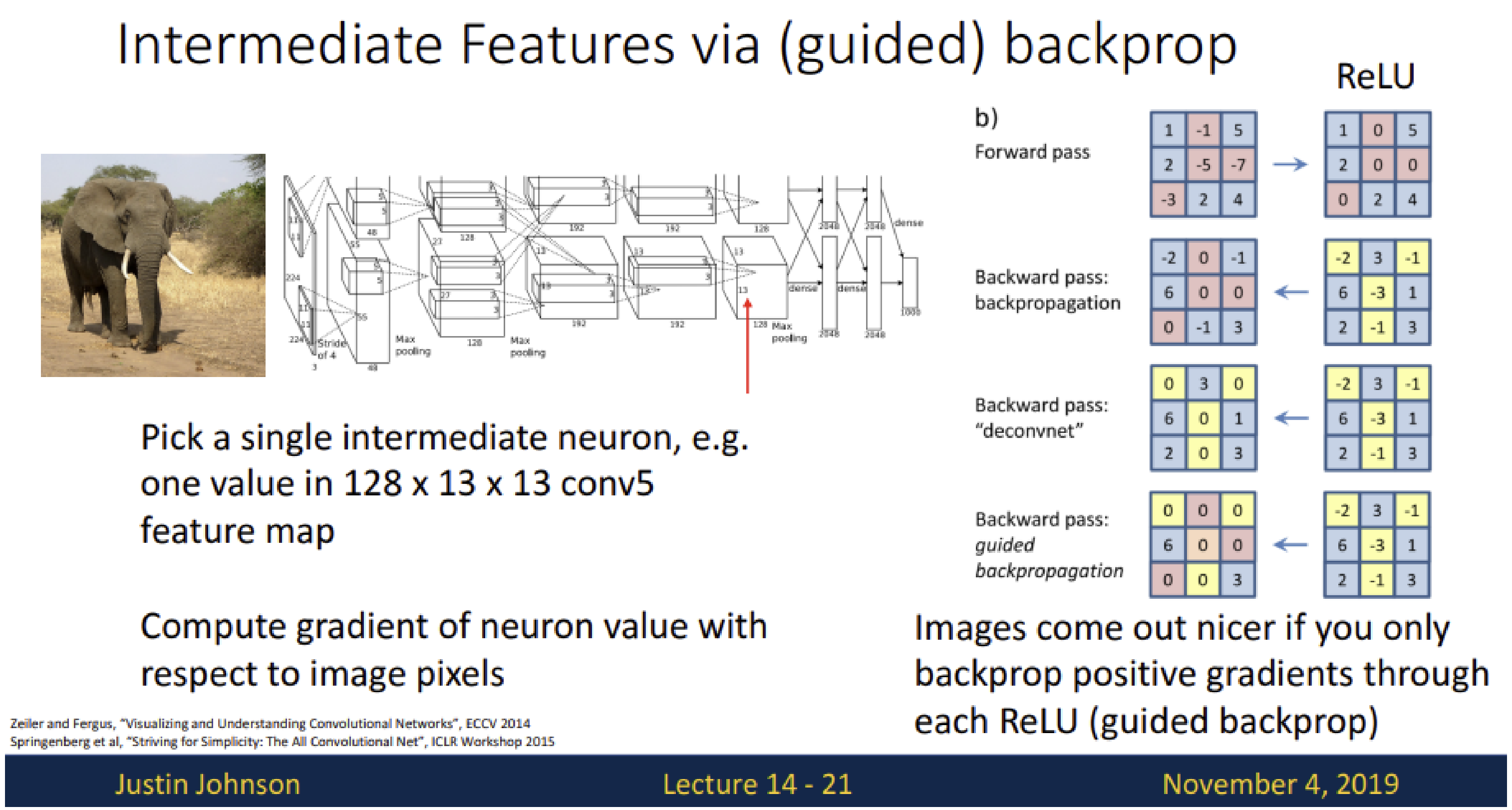
Michigan – Deep Learning for Computer Vision

Lecture 14: Visualizing and Understanding

TRAIN AND TEST

Lecture

Guided Backprop



- Input image를 intermediate neuron에 통과-> 어떤 픽셀이 영향을 많이 미치는지를 계산
- guided backprop →negative인 경우 전부 0으로 (ReLU 이용)
- 이유는없고 그냥 이미지가 잘 나와서 이렇게 함

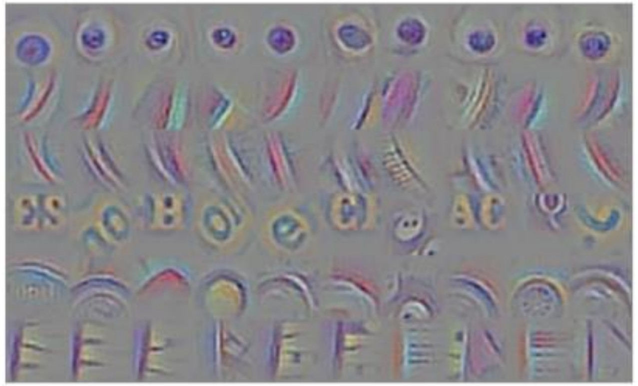
TRAIN AND TEST

Lecture

Guided Backprop

Intermediate Features via (guided) backprop

Maximally activating patches
(Each row is a different neuron)



Guided Backprop

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Justin Johnson | Lecture 14 - 22 | November 4, 2019

- 어떤 픽셀이 뉴런의 value에 영향을 미치는지 visualize
- test image에 국한하지 말고 어떤 image가 classification score를 maximize 하는지를 확인해보자→ gradient ascent

TRAIN AND TEST

Lecture

Gradient Ascent

Visualizing CNN Features: Gradient Ascent

(Guided) backprop:
Find the part of an image that a neuron responds to

Gradient ascent:
Generate a synthetic image that maximally activates a neuron

$$I^* = \arg\max_I \boxed{f(I)} + \boxed{R(I)}$$

Neuron value

Natural image regularizer

- Image를 initialize
- synthetic image를 만들 때까지 이미지를 gradient를 ascent하기

Lecture

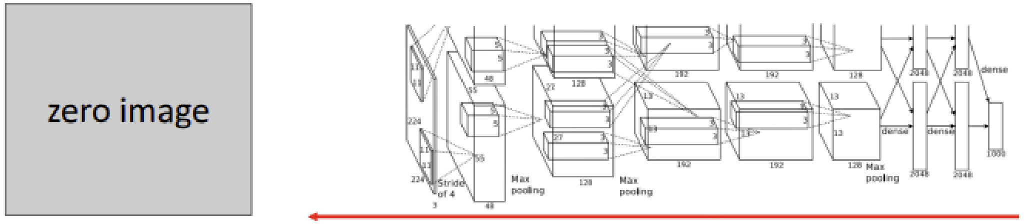
Gradient Ascent

Visualizing CNN Features: Gradient Ascent

$$\boxed{\arg\max_{I} \boxed{S_c(I)} - \lambda \|I\|_2^2}$$

score for class c (before Softmax)

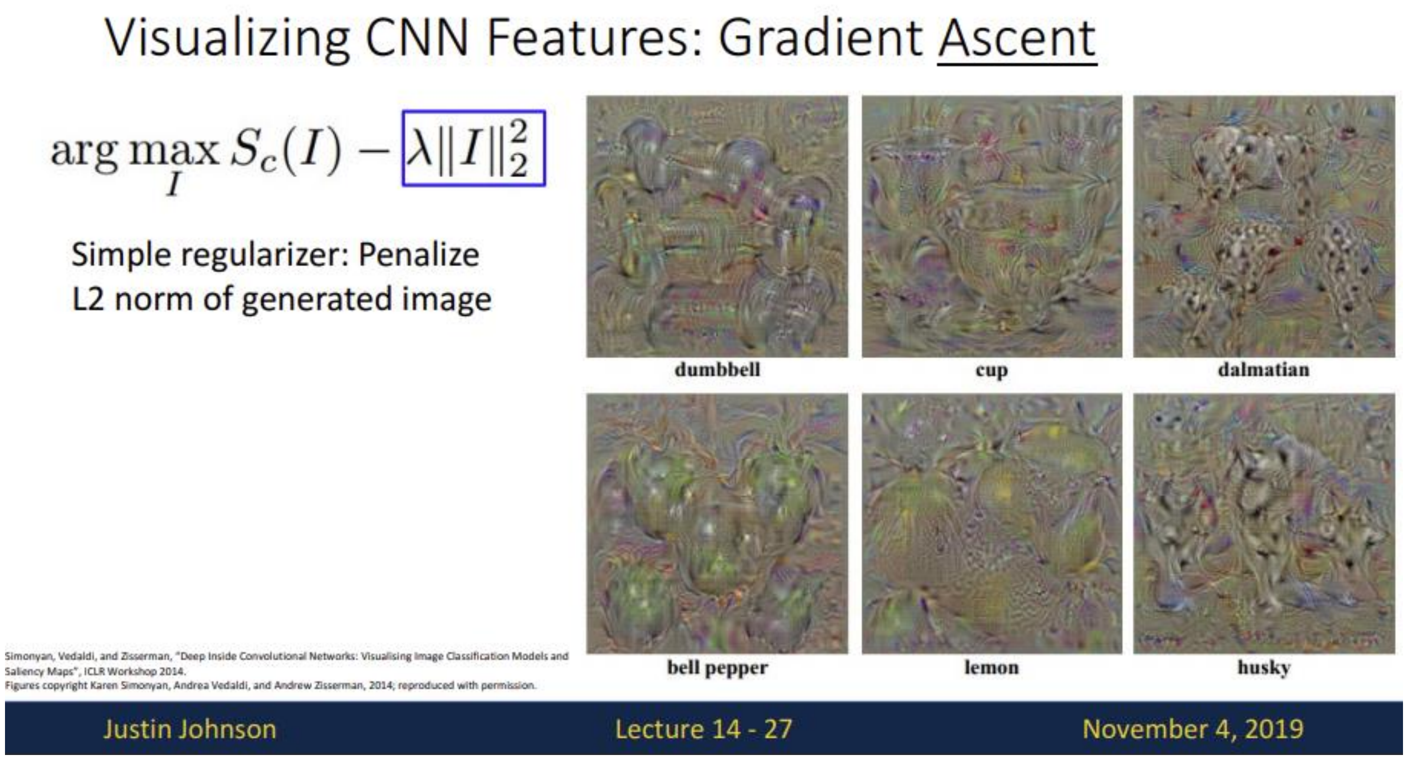
1. Initialize image to zeros



Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Lecture

Gradient Ascent



- 이미지를 learn하도록 하는 방식
- weight는 fix, image에 대해서 gradient ascent를 진행
- 해당 이미지가 realistic하지 않기에 다른 regularizer를 사용하도록 함 (blurring/clipping을 사용해서 좀더 realistic하게 만들 수 있음)

Lecture

Feature Inversion

Feature Inversion

Given a CNN feature vector for an image, find a new image that:

- Matches the given feature vector
- "looks natural" (image prior regularization)

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^{H\times W\times C}}{\operatorname{argmin}} \ \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda\mathcal{R}(\mathbf{x})$$

Given feature vector

Features of new image

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j}\left((x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2\right)^{\frac{\beta}{2}}$$

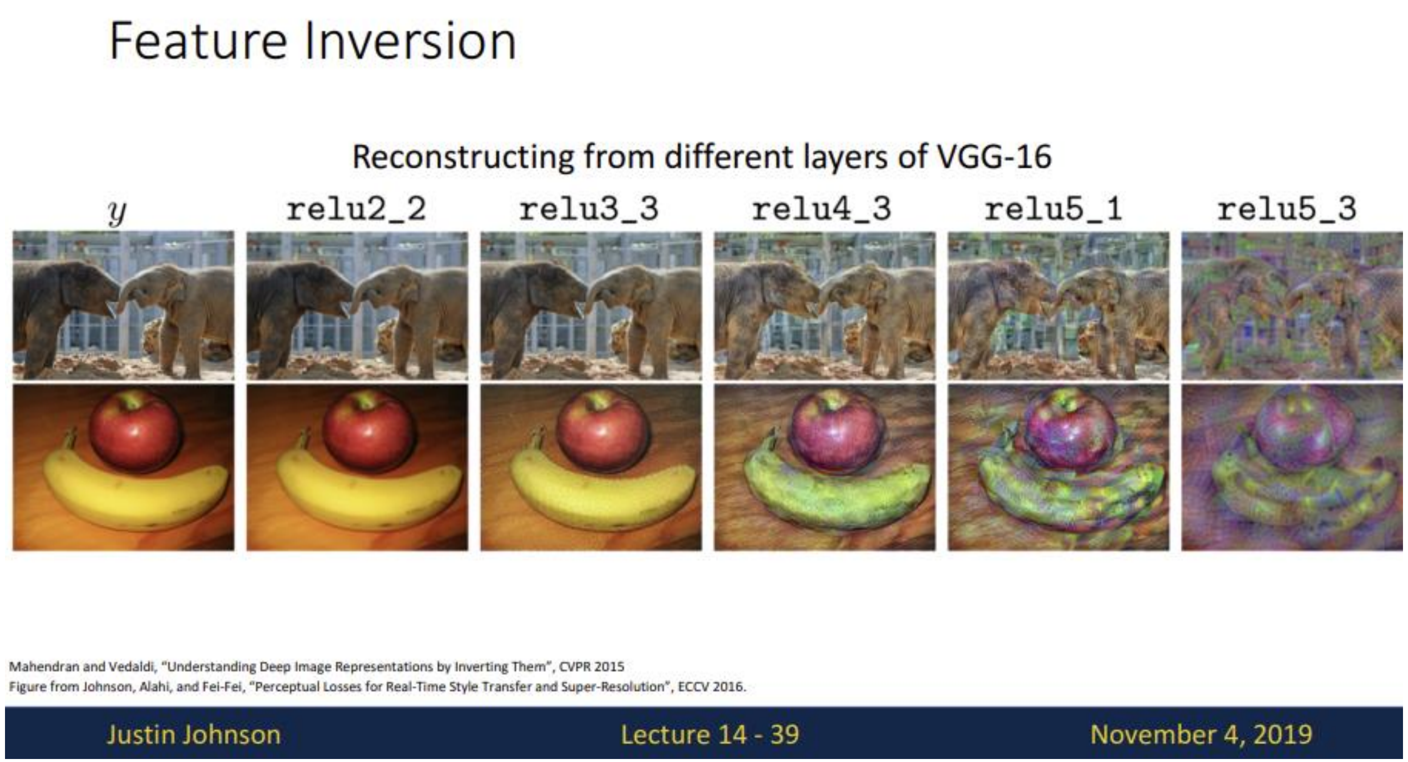
Total Variation regularizer (encourages spatial smoothness)

Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015

- image→ feature representation
- Feature representation을 가지고 새로운 이미지를 만들어 보자

Lecture

Feature Inversion



- input image→ extract feature representation→ new image
- relu4/5 손실 발생

  (row level feature 의 손실)
- relu5: color, local information 손실, raw information은 보존

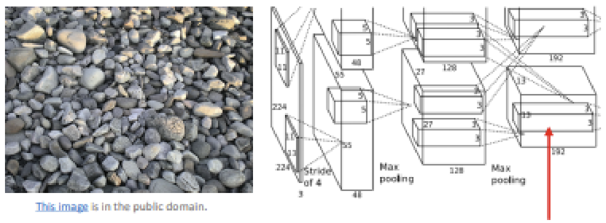
TRAIN AND TEST

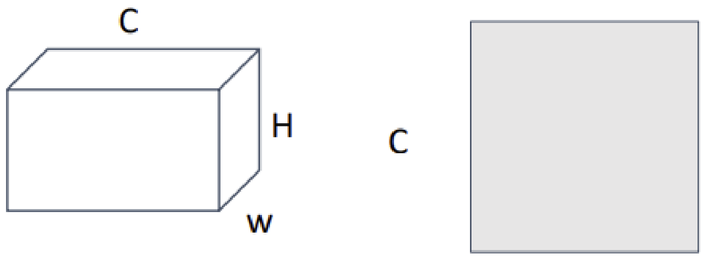
Lecture

Gram Matrix

input이미지에서 어떤 feature들이 연관이 있는지/없는지를 확인

Texture Synthesis with Neural Networks: Gram Matrix


This image is in the public domain.



Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

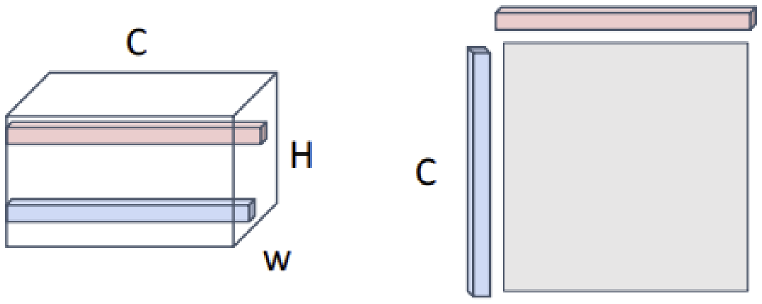
Outer product of two C-dimensional vectors gives C x C matrix of elementwise products

Average over all HW pairs gives **Gram Matrix** of shape C x C giving unnormalized covariance

Efficient to compute; reshape features from

C x H x W to F = C x HW

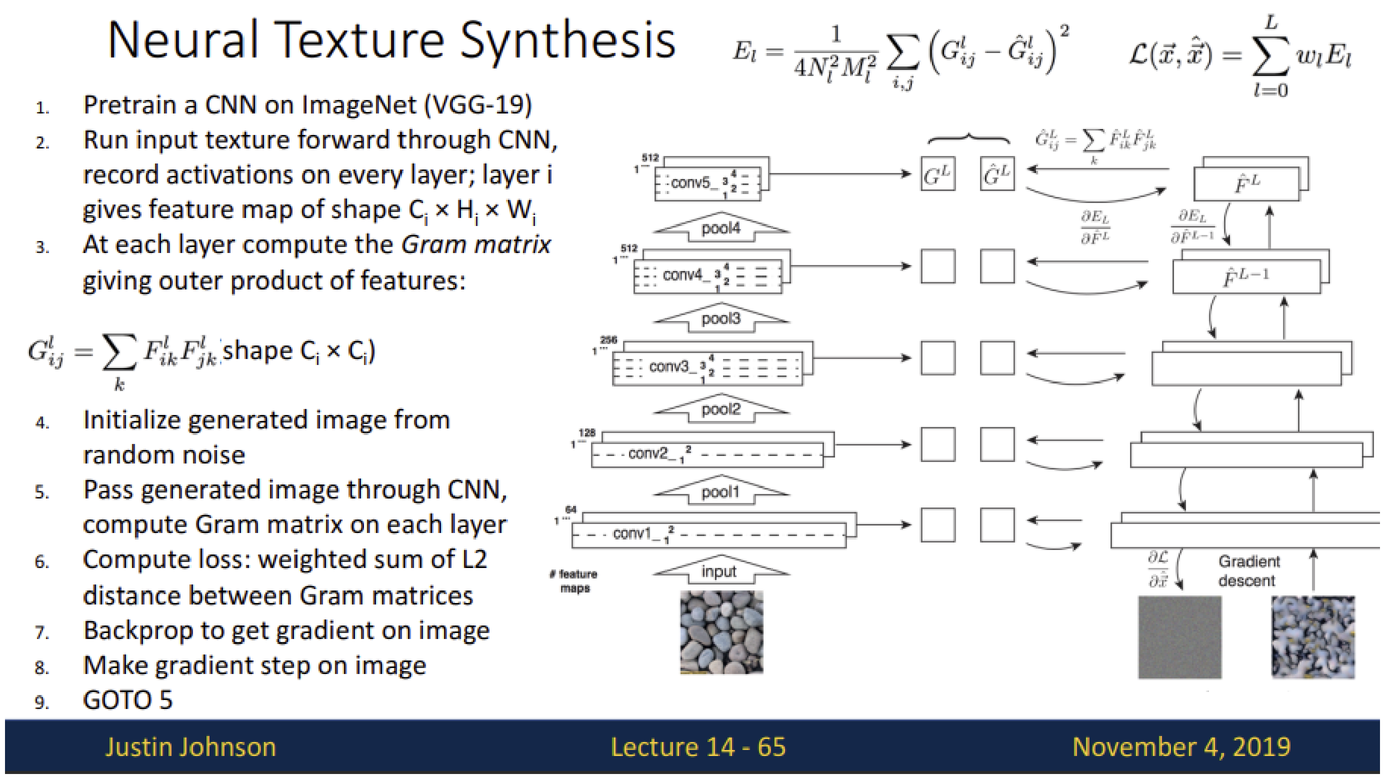
then compute $G = FF^T$



- 공간 정보는 버리고 texture만
- CNN -> 두개의 vector뽑아서 element wise product를 진행 →average를 구하기
- Output= (CxC: GRAM Matrix)

Lecture

Gram Matrix



- input image→NN 통과→gram matrix
- generated image를 initialize→ 각 레이어 별로 gram matrix 계산
- real gram matrix와 유클리드 거리→픽셀별로 gradient descent→image에 대해서 gradient step→generate image(input과 같은 gram matrix를 가지는)

TRAIN AND TEST

Lecture

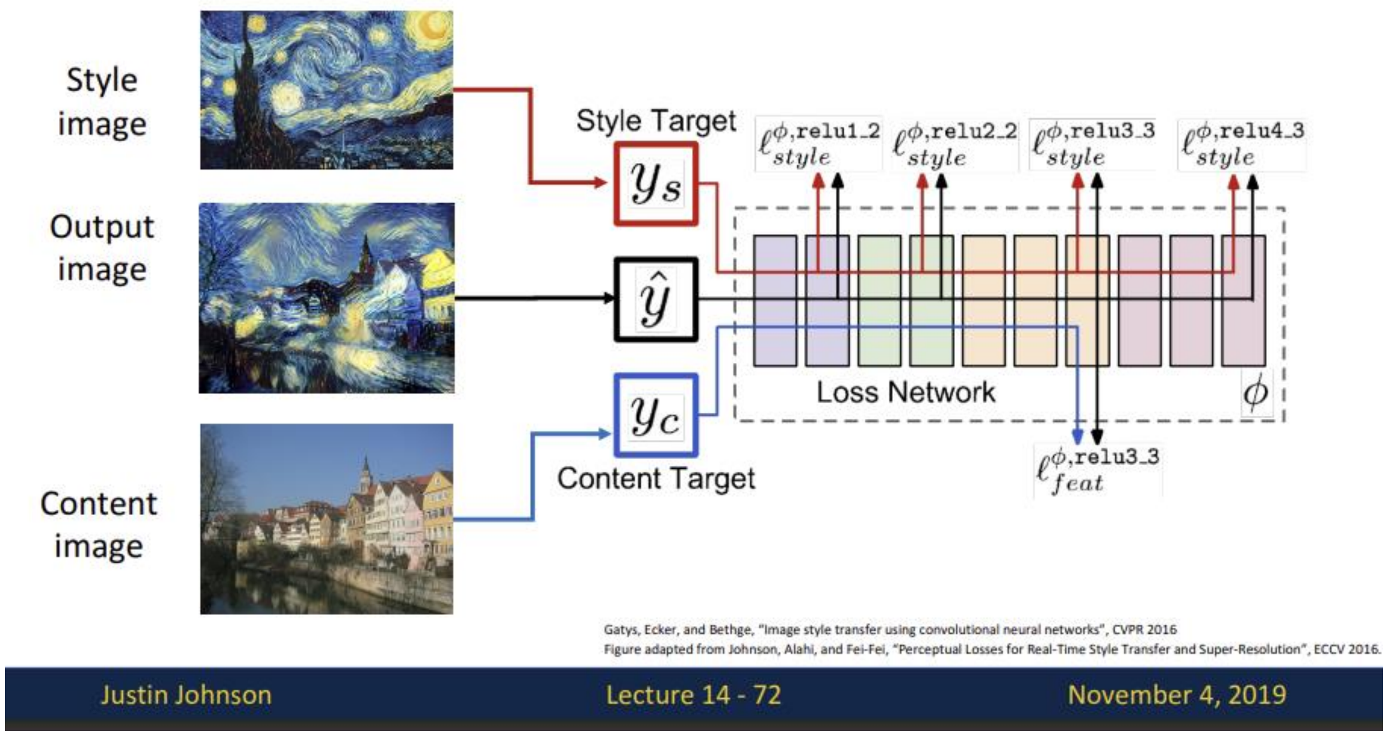
Gram Matrix

- 공간 정보 손실

TRAIN AND TEST

Lecture

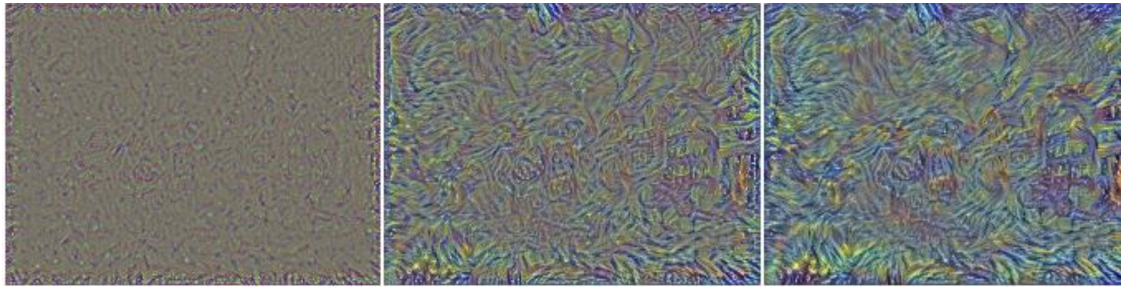
Style Transfer



- Output을 initialize하고 Style image의 gram matrix를 따르도록 gradient ascent

Lecture

Style Transfer



More weight to content loss ⟷ More weight to style loss

Larger style image ⟷ Smaller style image

TRAIN AND TEST

Paper Review

Image Style Transfer Using Convolution Neural Network

TRAIN AND TEST

Paper

1. Abstract – goal
2. Introduction – 등장 배경 + CNN 활용
3. Method/ Model – A Neural Algorithm of Artistic Style
4. Result/ Discussion

TRAIN AND TEST

Paper

Abstract

- **Previous Method**

Problem: Semantic information을 표현하는 **image representation**이 없음

Solution: content 와 style을 분리해보자

- **A Neural Algorithm of Artisric Style**

CNN을 통해 image representation을 추출하는 방식을 제안

이는 image content, style의 separate, recombine을 가능하게 함

*** Image의 content와 style을 분리하는 method를 제시하겠다!!**

Paper

Introduction

- **Style transfer = Texture transfer**

Texture transfer: semantic content는 보존하면서 texture를 소스 이미지에 합성

등장 배경: 이전 연구에서는 저수준 이미지 특징만을 활용

(ex. face+illumination condition, font+handwriting)

- **CNN**

CNN기반의 parametric texture model에 이미지 representation을 변경하는 방법을 combine

CNN을 어떤식으로 이용하는가?

TRAIN AND TEST

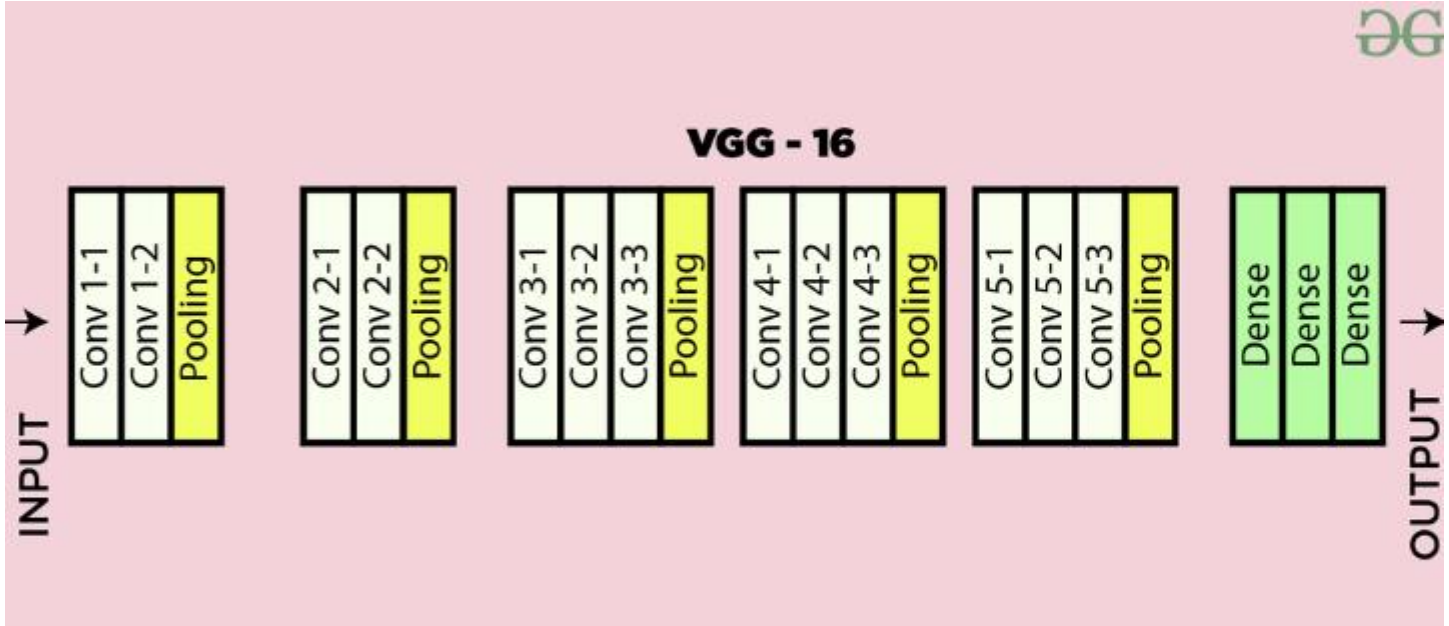
Paper

Method

VGG + Gram Matrices

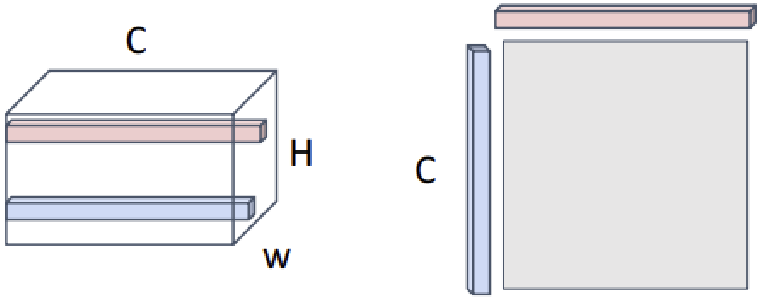
- **VGG**

제일 상위의 fc layer는 사용하지 않고 image net으로 학습시킨 weight를 사용



- **Gram Matrices**

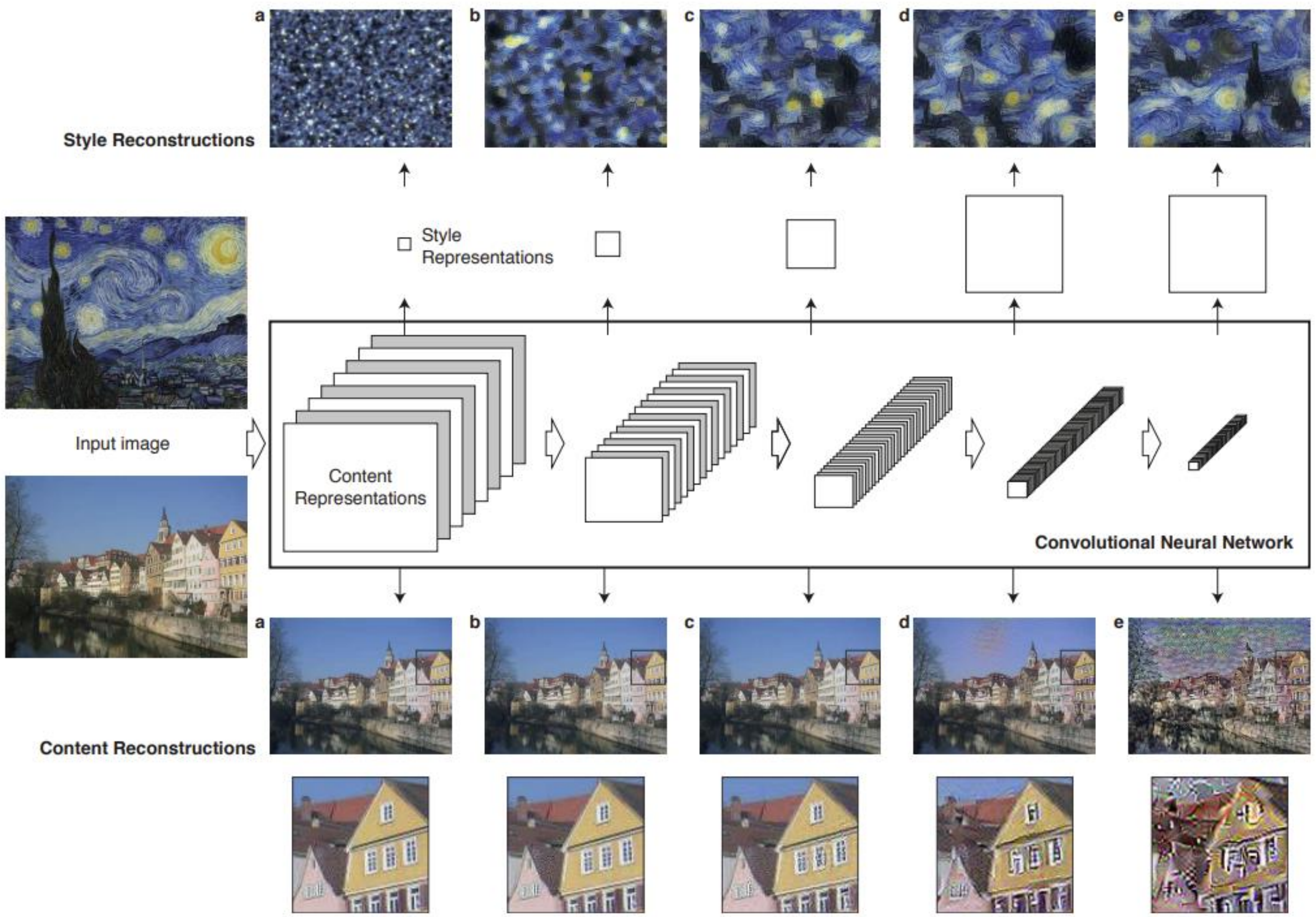
CNN결과 feature에 대해서 correlation을 구하도록 함



Paper

Method

- CNN의 image representation은 각 처리 단계별 필터링된 이미지 집합으로 표현
- processing stage를 따라 필터 수 증가
- 필터링된 이미지의 크기는 일부 다운샘플링 메커니즘(max pooling)에 의해 감소



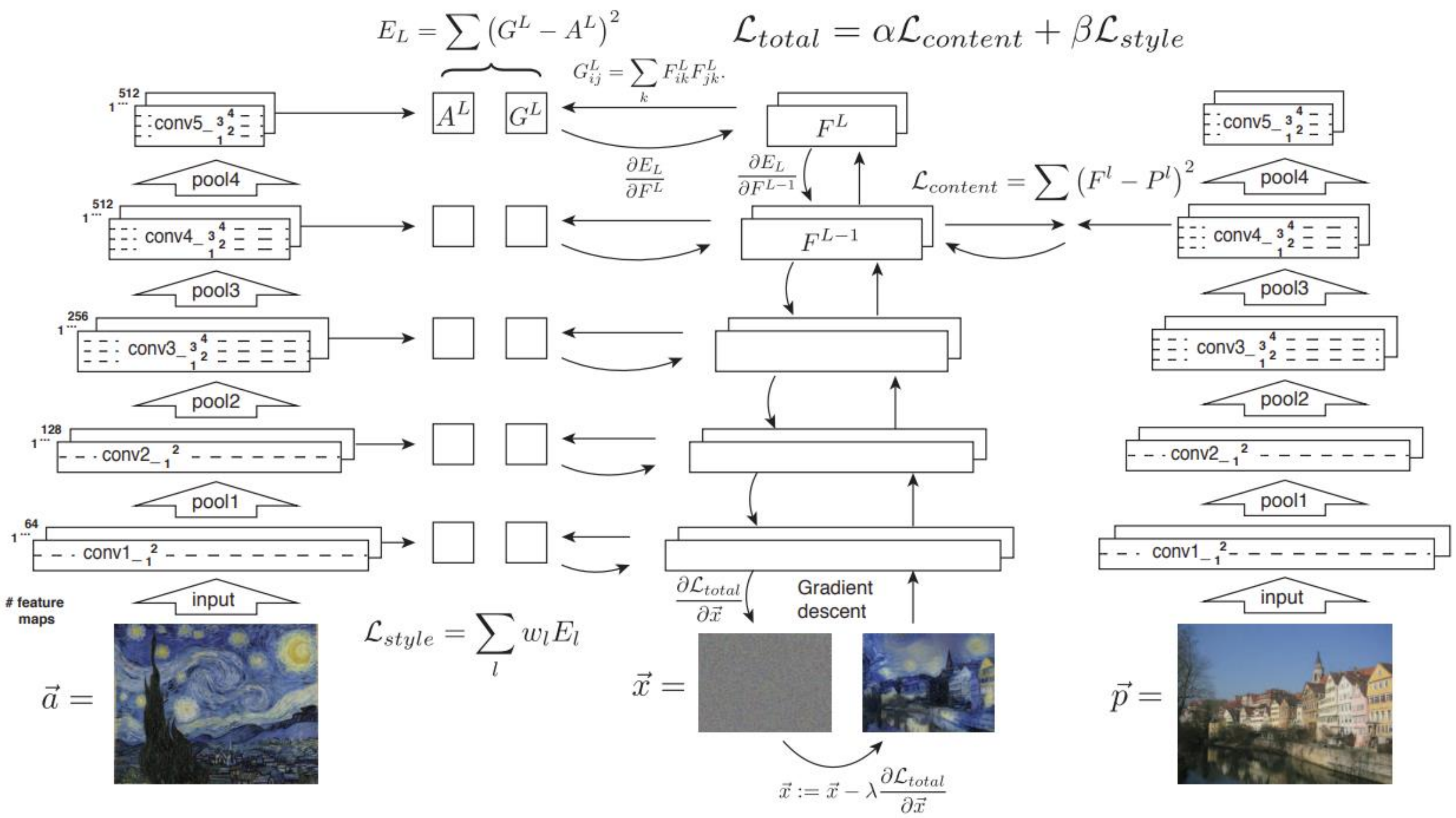
TNT
TRAIN AND TEST

Paper

A: style image
P: content image
X: generated image

Model

1. Content, style feature 추출/저장
2. A의 style representation인 A_l이 계산/저장
3. P의 content representation P_l 계산
4. random white noise image x의 style feature **G_l**과content feature 인 **F_l**이 계산
5. G_l과 A_l & F_l 과 P_l loss 계산
6. 각 loss를 linear combination해서 total loss→ gradient descent

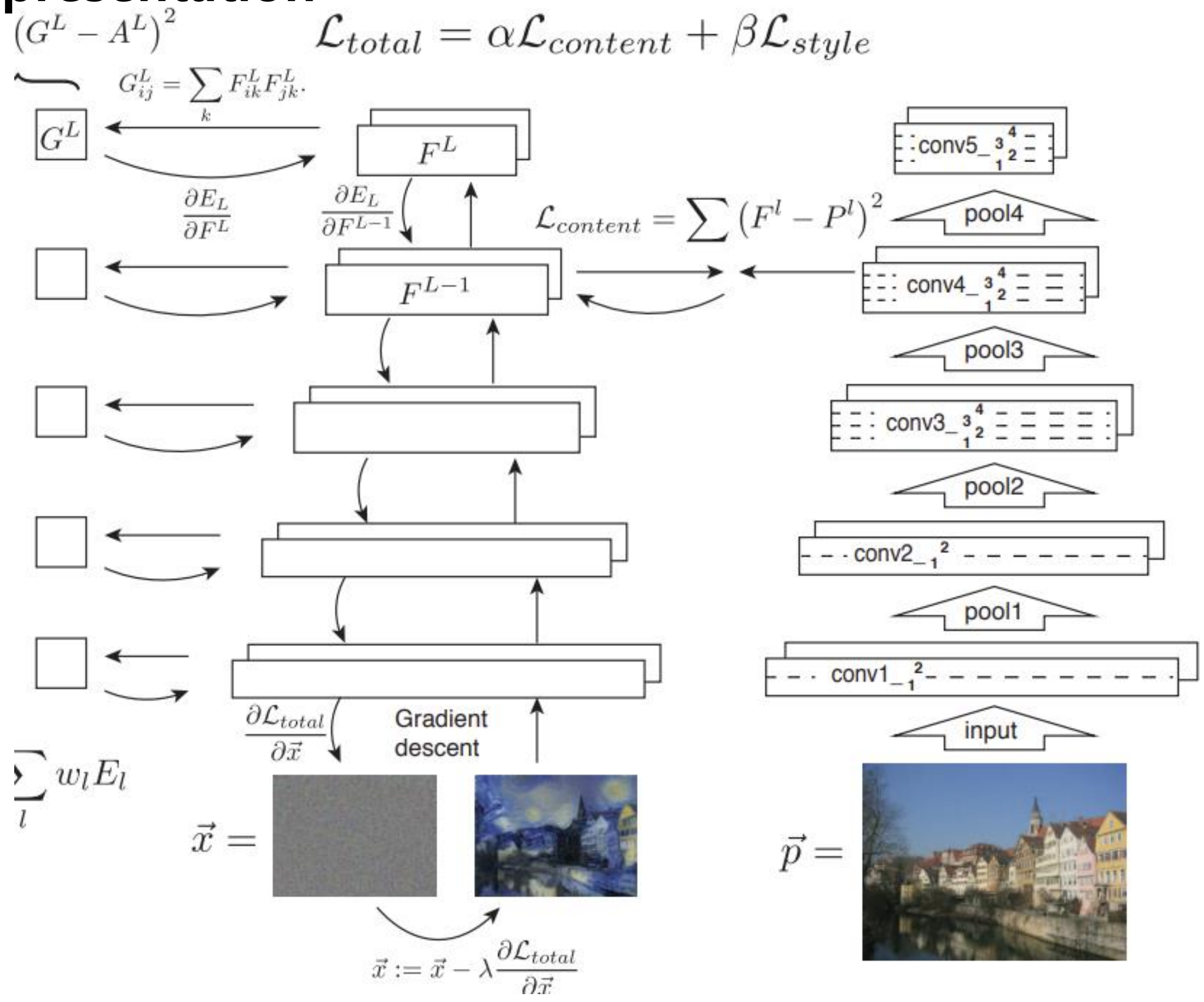


Paper

A: style image
P: content image
X: generated image

Model

content representation



- L번째 레이어의 F_ij와 P_ij를 계산
- F_ij는 레이어 l의 j번째 위치에서 i번째 필터의 활성화를 의미

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}\left(F^l_{ij} - P^l_{ij}\right)^2 .$$

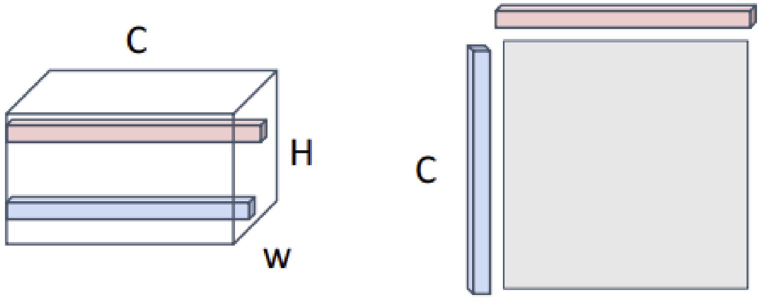
$$\frac{\partial \mathcal{L}_{\text{content}}}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 , \end{cases}$$

Paper

A: style image
P: content image
X: generated image

Model

Style representation

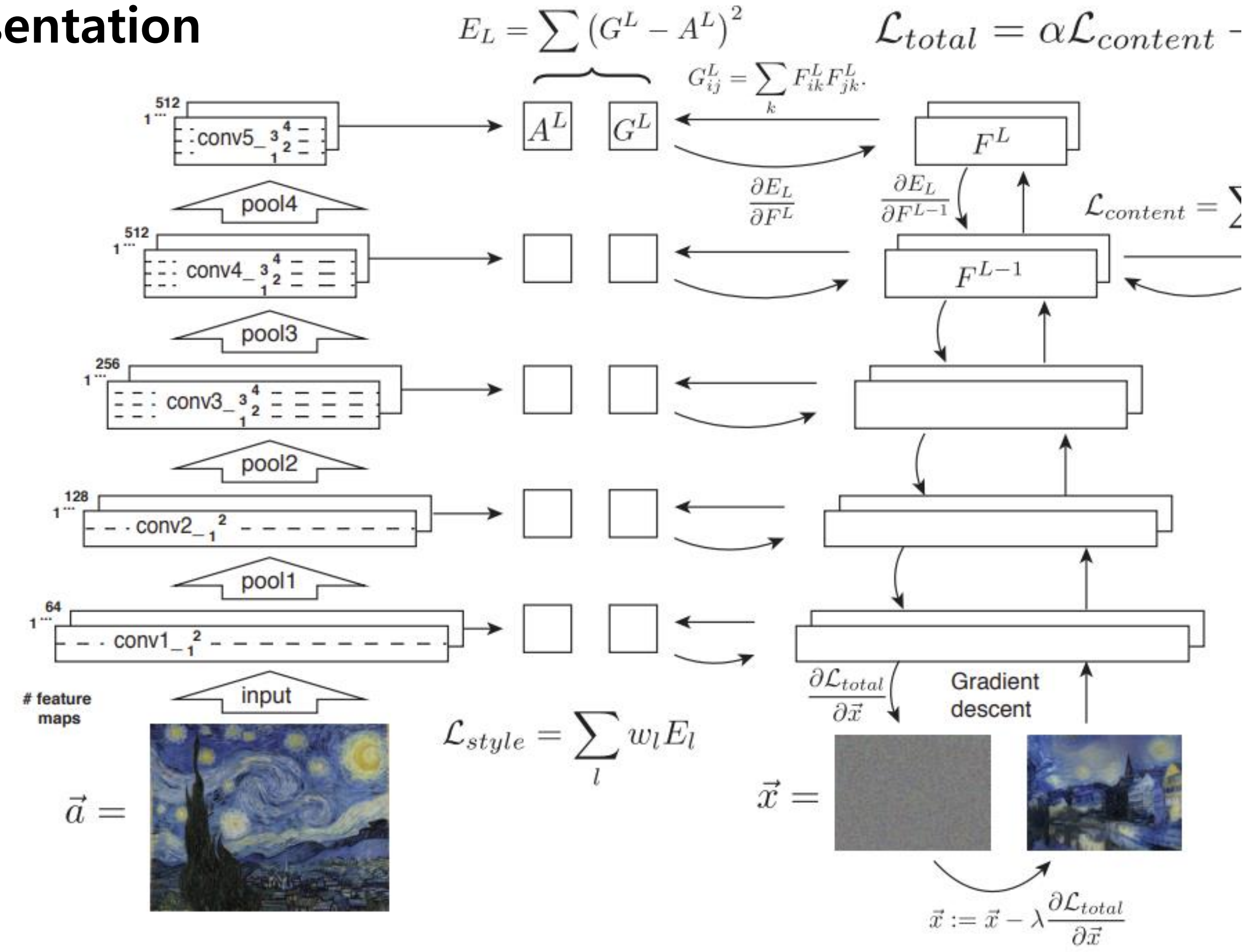


G_l(i,j)는 feature map의 i와 j의 내적

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G^l_{ij} - A^l_{ij}\right)^2$$

$$\mathcal{L}_{\text{style}}(\vec{a},\vec{x}) = \sum_{l=0} w_l E_l,$$

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2}\left(\left(F^l\right)^{\mathrm{T}}\left(G^l - A^l\right)\right)_{ji} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 \end{cases}$$



Paper

Result/Discussion

Result

- **Trade-off between content and style matching**

content를 강조하면 style이 not well-matched

특정한 콘텐츠와 스타일 이미지 쌍에 대해서는 콘텐츠와 스타일 간의 균형을 조정 가능

- **Effect of different layers of the Convolutional Neural Network**

content와 style representation을 고를 수 있음

네트워크의 낮은 레이어에서 매치-> 작품의 질감이 단순히 사진 위에 혼합된것 처럼 보임

네트워크의 높은 레이어에서 매치-> 작품의 질감과 사진의 콘텐츠가 적절하게 병합

Discussion

생성된 이미지의 해상도

일부 저수준의 잡음에 영향을 받음

TRAIN AND TEST

TRAIN AND TEST