CVPR 2016

Image Style Transfer
Using Convolutional Neural Networks

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

고민수

Summary

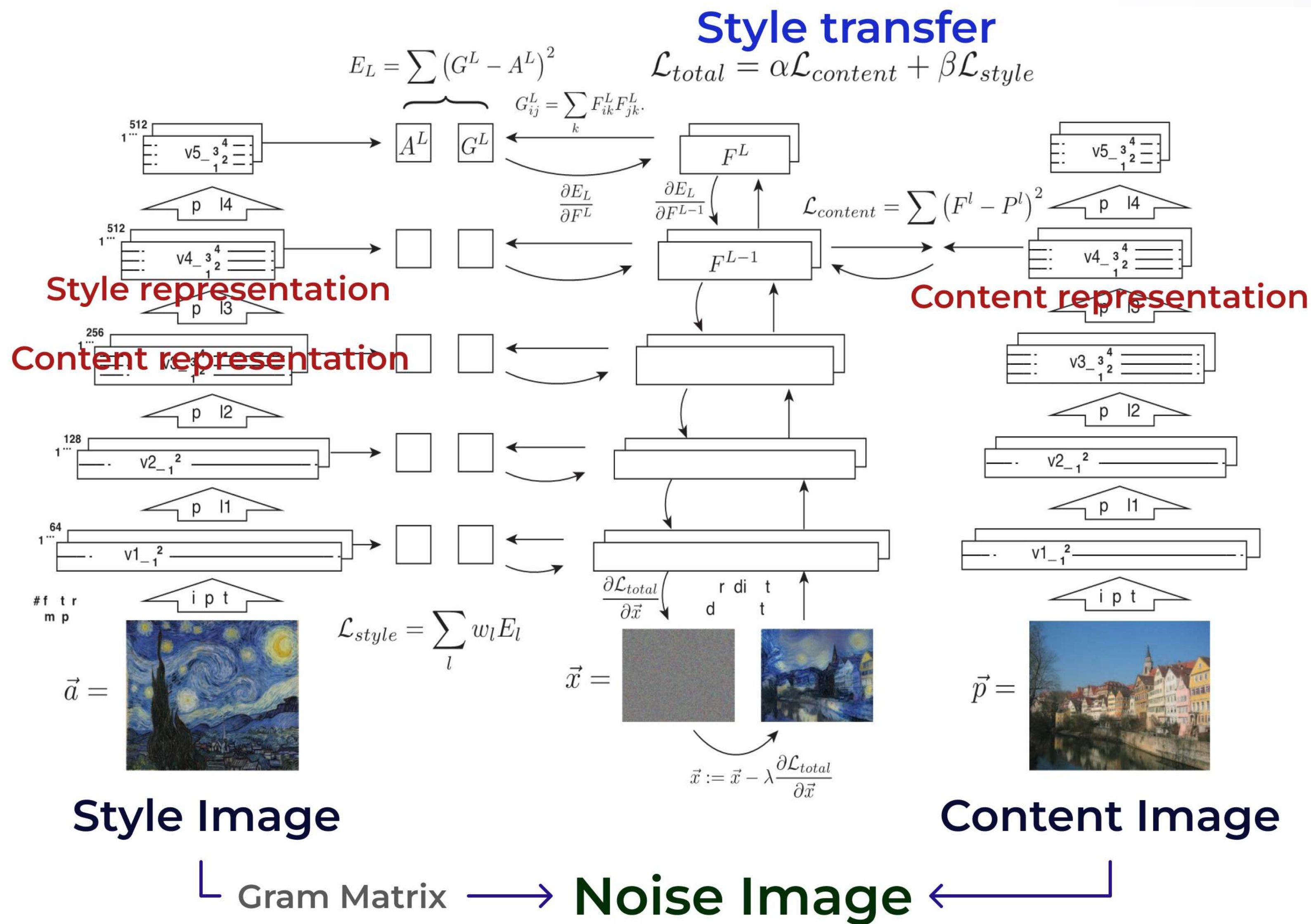**Content Image**

Content representation ↓

Noise Image

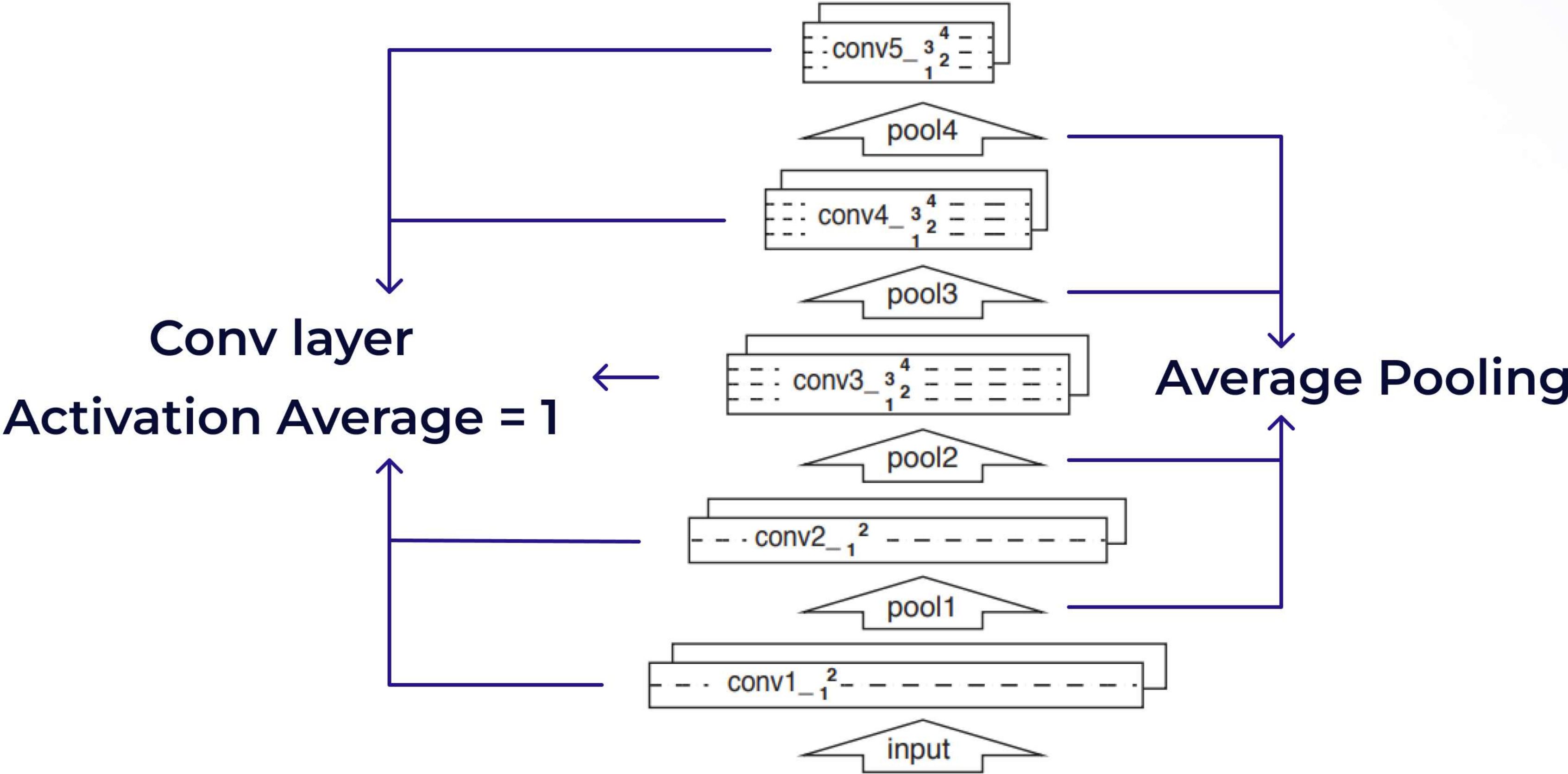Style transfer ↑ Gram Matrix

Style Image

Summary

Deep image representations

Style transfer

$$E_L = \sum \left(G^L - A^L\right)^2 \qquad \mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

$$G^L_{ij} = \sum_k F^L_{ik} F^L_{jk}.$$

$$\frac{\partial E_L}{\partial F^L} \qquad \frac{\partial E_L}{\partial F^{L-1}} \qquad \mathcal{L}_{content} = \sum \left(F^l - P^l\right)^2$$

Style representation

Content representation

$$\mathcal{L}_{style} = \sum_l w_l E_l$$

$$\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$$

$$\vec{a} = \qquad \vec{x} = \qquad \vec{p} =$$

$$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$$

Style Image   **Content Image**

Gram Matrix ⟶ **Noise Image** ⟵

Deep image representations

Deep image representations

Pretrained VGG19
(object recognition and localisation)

No FC layers

Conv layer

Activation Average = 1

Average Pooling

conv5_3 4 2 1

pool4

conv4_3 4 2 1

pool3

conv3_3 4 2 1

pool2

conv2_1 2

pool1

conv1_1 2

input

Content representation

Content Loss

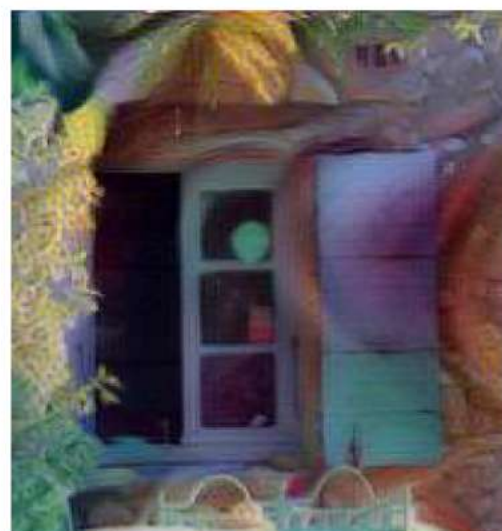$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^l - P_{ij}^l \right)^2$$

feature map F

feature map P

Noise Image $\vec{x}$

Content Image $\vec{p}$

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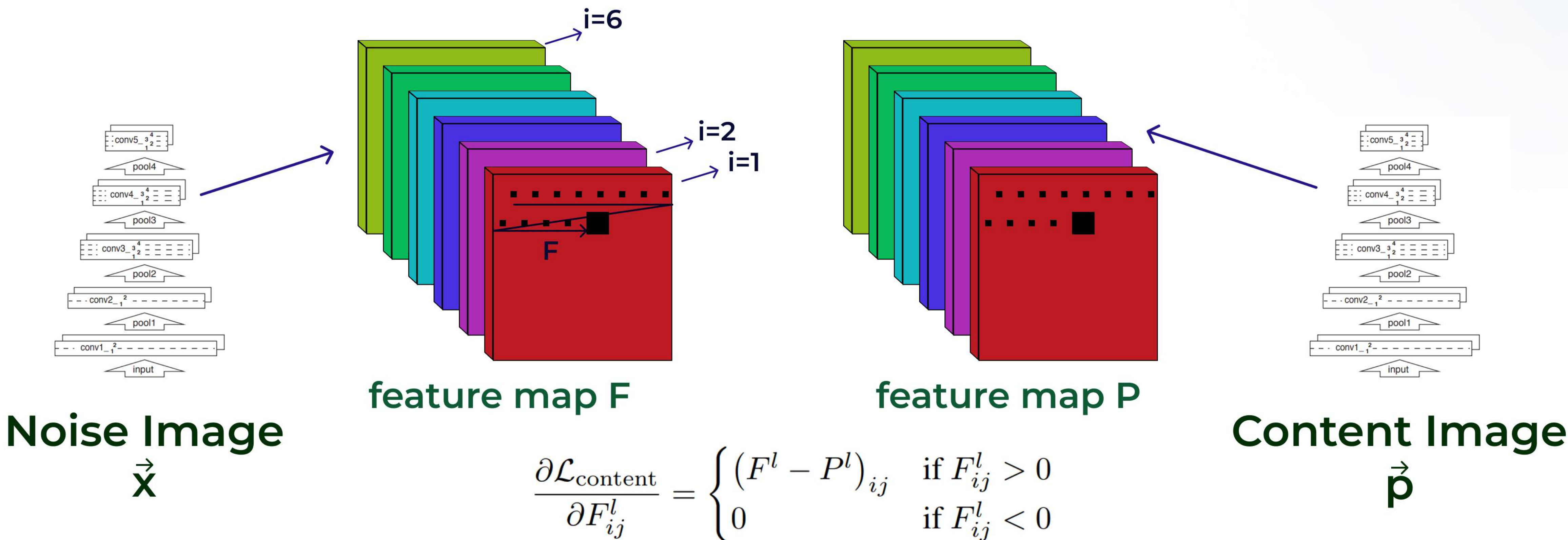
Content representation

Content Image

Conv2_2

Conv4_2

Content representation

higher layers

Content Representations

Convolutional Neural Network

b

c

d

e
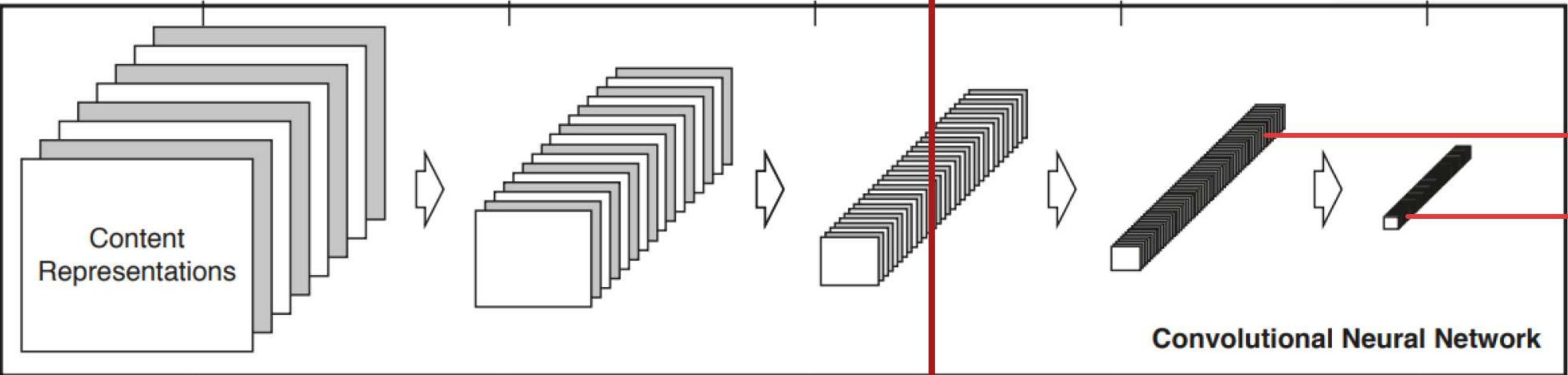
Style representation

Style : Correlations between the different channel response

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

Channel size= 6

Each channel : i or j

k
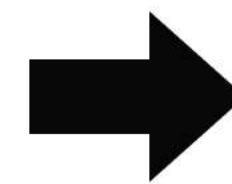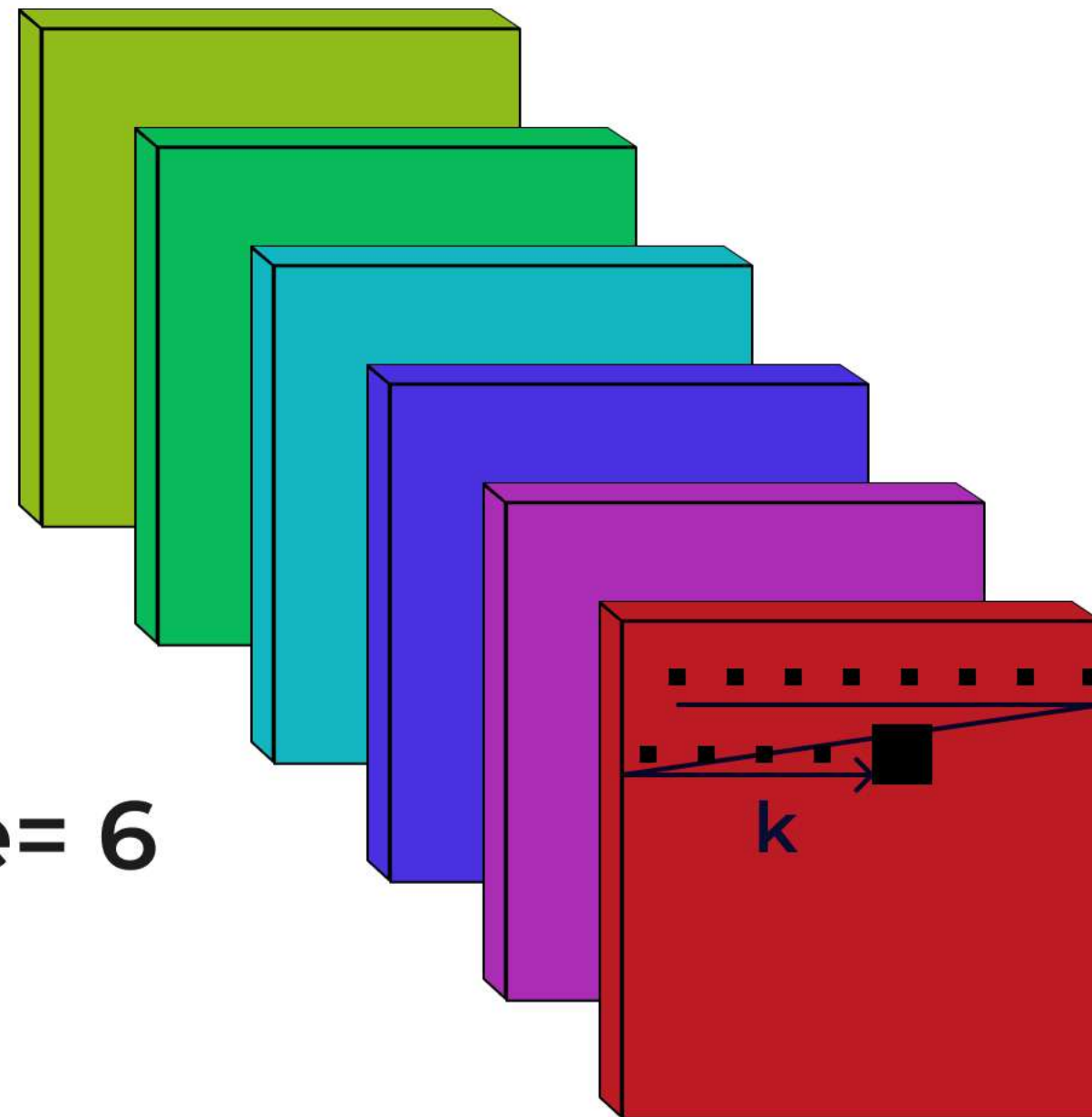
G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}
G_{21}	G_{22}	G_{23}	G_{24}	G_{25}	G_{26}
G_{31}	G_{32}	G_{33}	G_{34}	G_{35}	G_{36}
G_{41}	G_{42}	G_{43}	G_{44}	G_{45}	G_{46}
G_{51}	G_{52}	G_{53}	G_{54}	G_{55}	G_{56}
G_{61}	G_{62}	G_{63}	G_{64}	G_{65}	G_{66}

Style representation

Style Loss

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - A_{ij}^l \right)^2$$

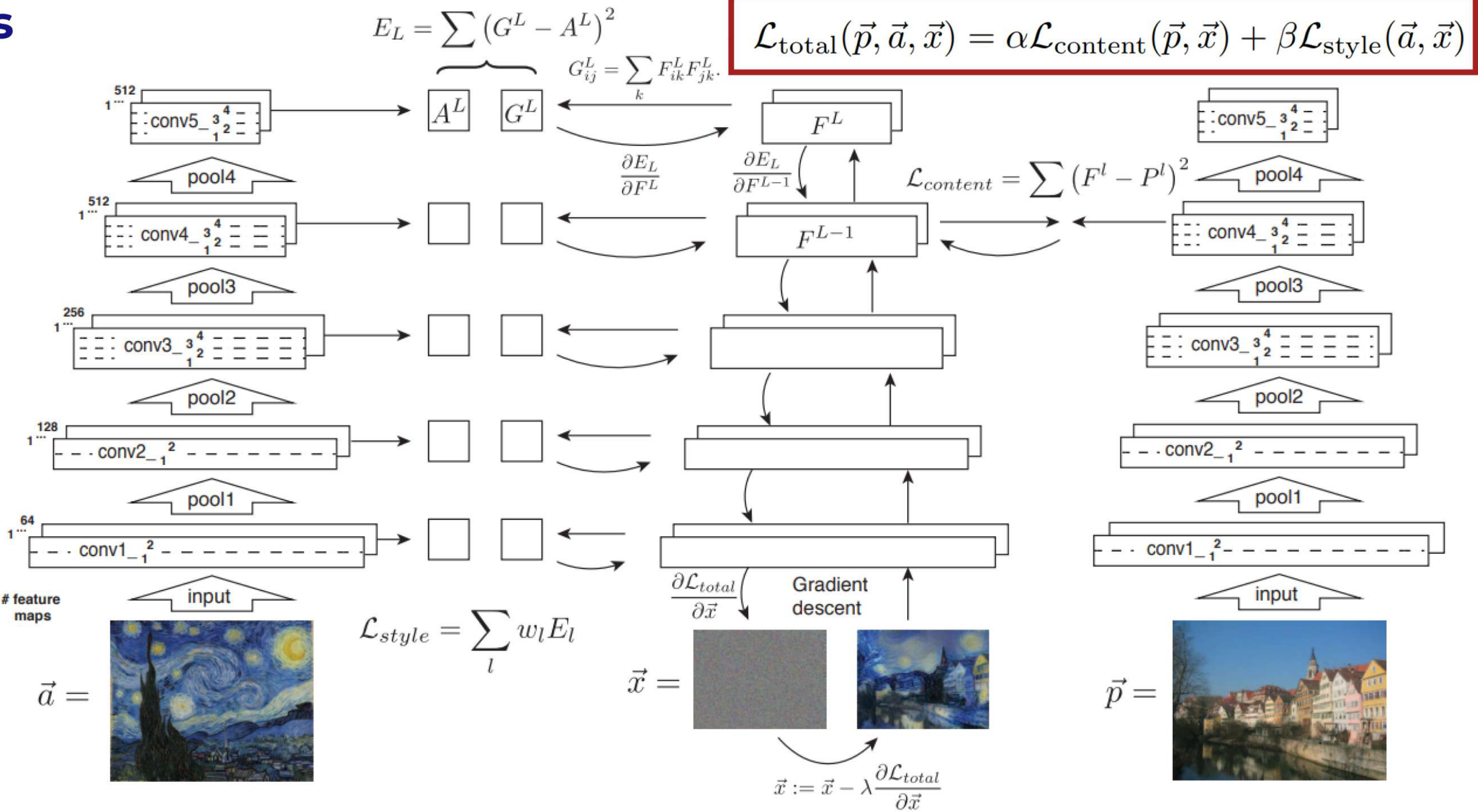$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$
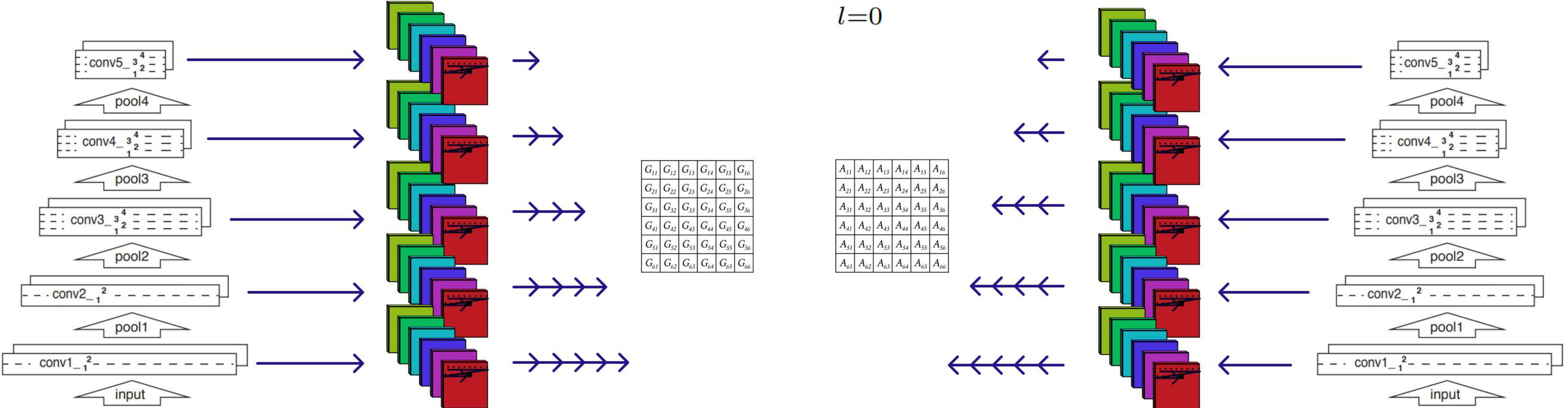
Noise Image
$\vec{x}$

Style Image
$\vec{a}$

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

Style transfer

Total Loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta\mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

- Optimisation strategy : L-BFGS
- Same size (Style image, Content image)

Result

Trade-off between content and style matching

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

α/β = 0.1, 0.01, 0.001, 0.0001

Content $\longleftrightarrow$ **Style**

Content : **Conv4_2**

Style : **Conv1_1, Conv2_1**
Conv3_1, Conv4_1
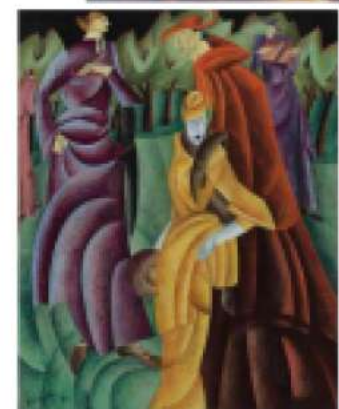**Conv5_1**

Effect of different layers of the CNN

Content Image

Conv2_2

Conv4_2

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

Conv layer = Conv2_2, Conv4_2

Content $\longleftrightarrow$ **Style**

α/β = 0

Initialisation of gradient descent

Start with Noise image

Start with Content image

Start with Style image

- Initialisations is not strong effect on outcome of the synthesis procedure
- Initialising with a fixed image always deterministically leads to the same outcome

Thank you

Image Style Transfer
Using Convolutional Neural Networks

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

고민수