Neural Style Transfer

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

Werner Chao

Felipe Perez

Xiyang Chen

March 14th, 2019

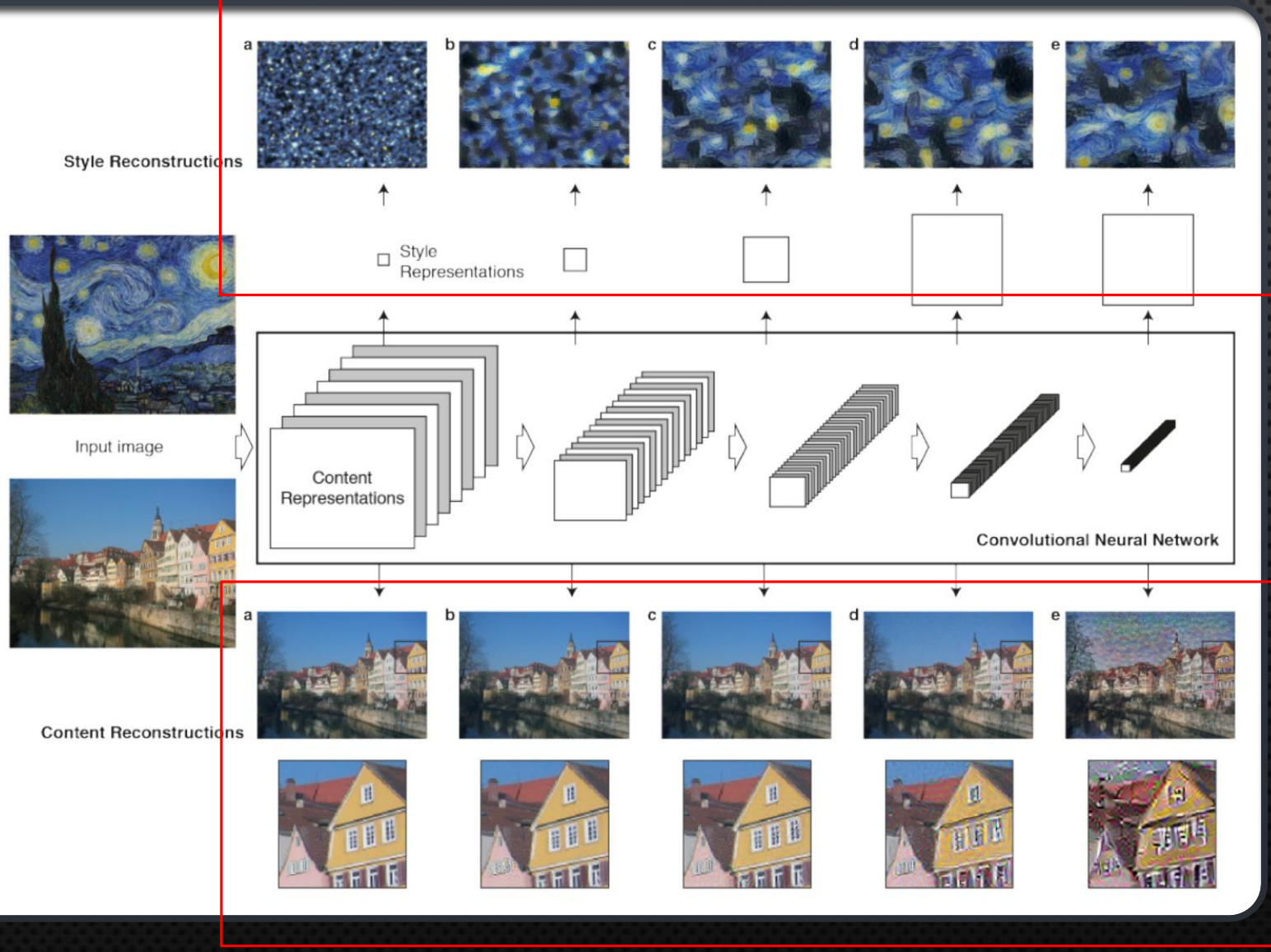
Gatys et al 2015

Motivation

- Humans are great at creative things, such as art
- We don’t understand well why humans are able to do this
- No artificial intelligence system exist that can create art parallel to human
- Can we have an algorithmic understanding how humans create art?

Overall process

Loss Function



Reconstructions from different CNN layers In VGG Network

Content and style in cnn are separable

$$L_{total}(\vec{c}, \vec{s}, \vec{x}) = \alpha L_{content}(\vec{c}, \vec{x}) + \beta L_{style}(\vec{s}, \vec{x})$$

- $\vec{c} = content$
- $\vec{s} = style$
- $\vec{x} = white\ noise$
- $^{\alpha}/_{\beta} = 1e-3\ or\ 1e-4$

Important notes

- No model learning: frozen VGG network
- image reconstruction only
- Target:
  - activation from cnn layer(s) in pre-trained vgg network
- Only 1 pair of content and style image are used

Content loss

$$L_{content}(\vec{c}, \vec{x}, l) = \frac{1}{2} \sum_{i,j,k} (F^l_{i,j,k} - C^l_{i,j,k})^2$$

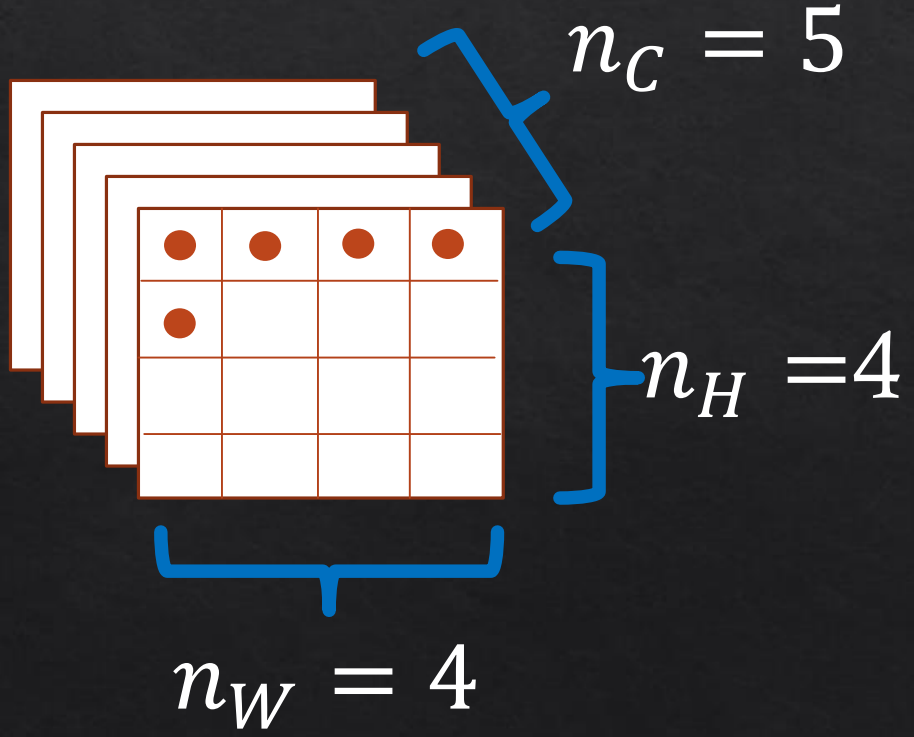
White noise

Target content, output from a VGG mid-layer

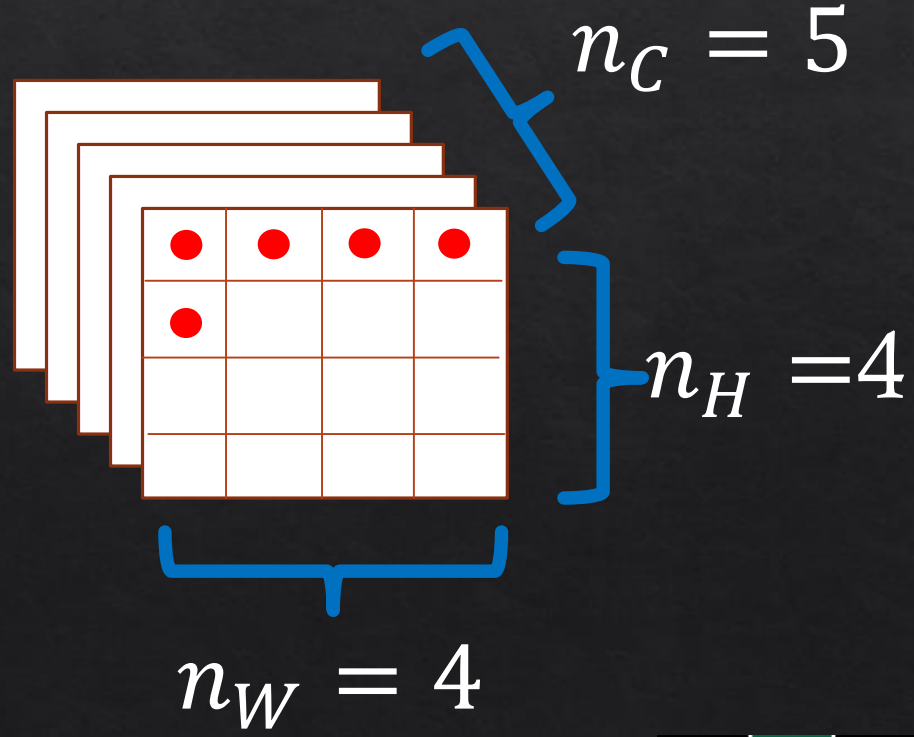
- i: *index of height*
- j: *index of width*
- k: index of channel
- l: certain CNN layer in VGG

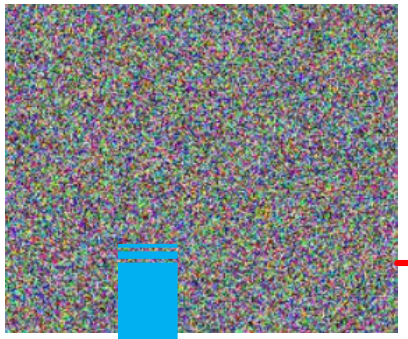
Content loss – output of 1 cnn layer in vgg

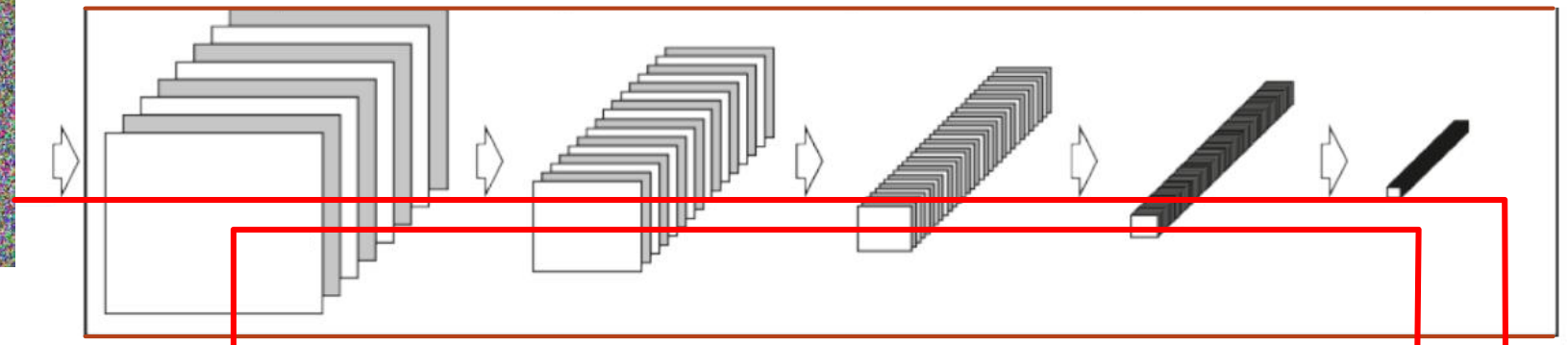
Content Image



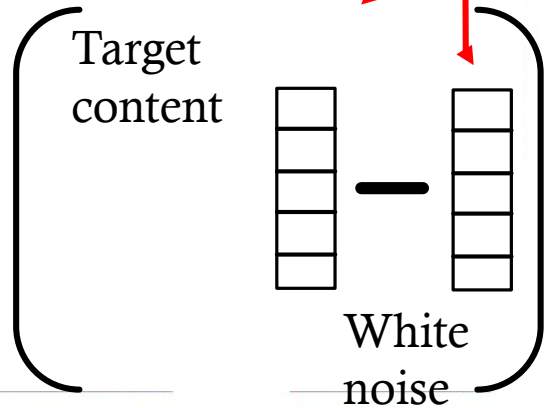
White Noise

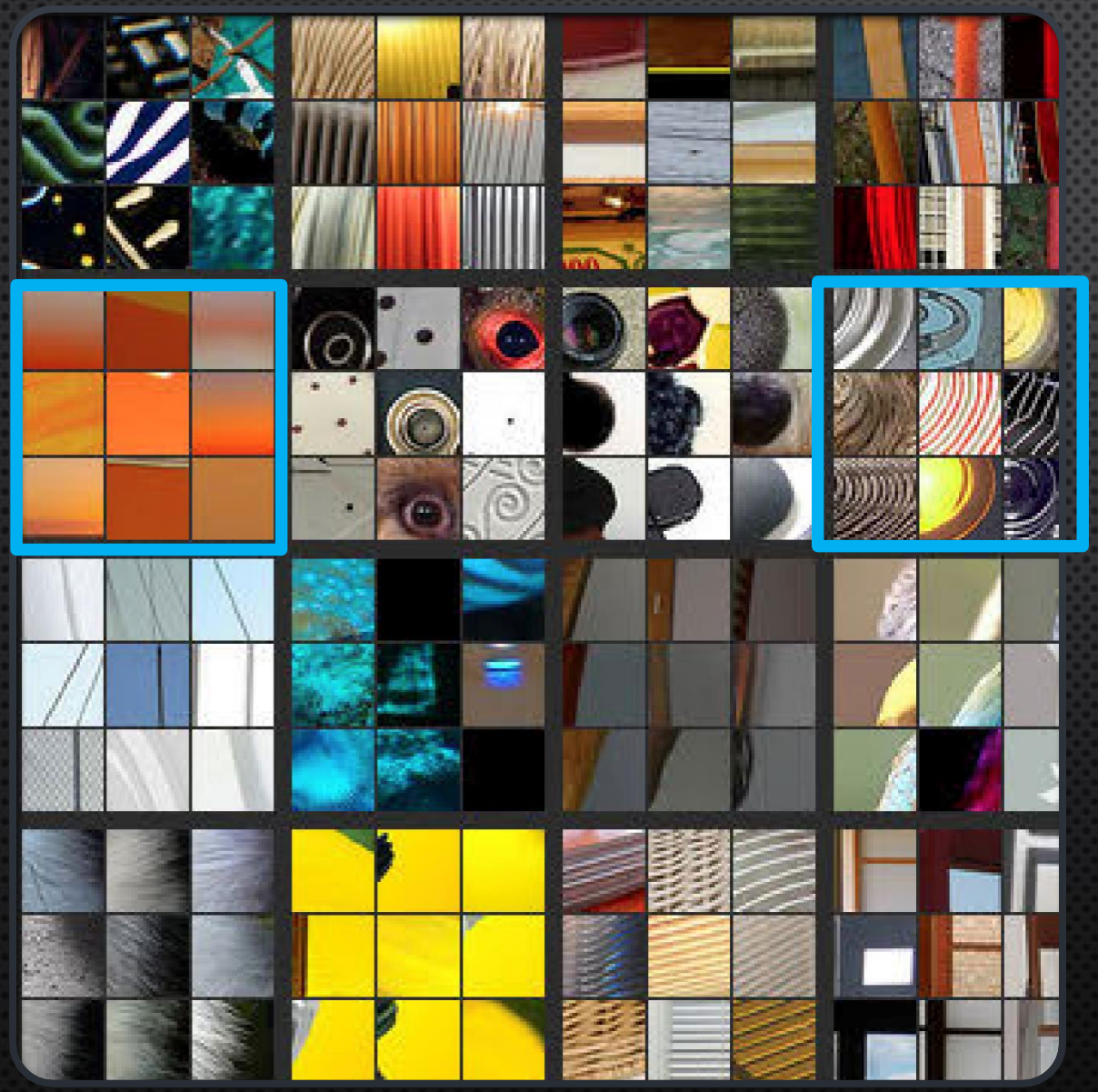






Minimize





Gram matrix - correlation intuition

Style loss

$$L_{style}(\vec{s}, \vec{x}, l) = \frac{1}{(2n_W n_H n_c)^2} \sum_{k,k'} (G^l_{k,k'}, -S^l_{k,k'})^2$$

White noise

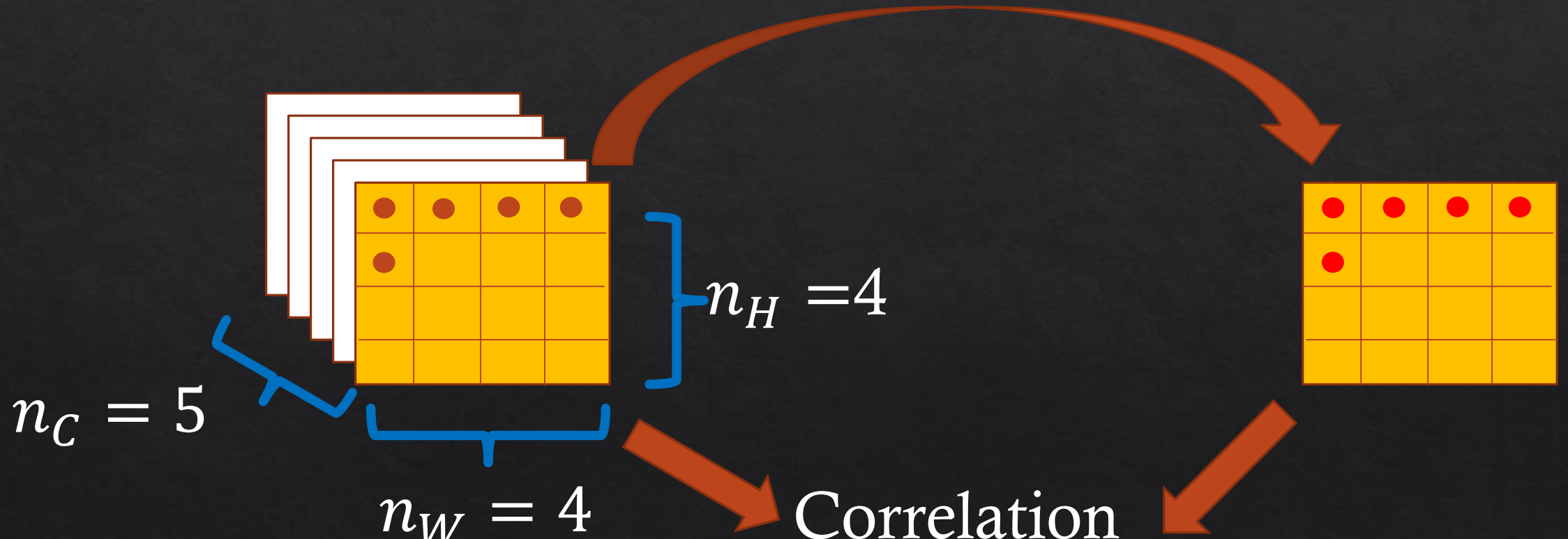
Target style, output from a VGG CNN layer

$$G_{k,k'}{}^l = \sum_{i,j} a^l_{i,j,k}\, a^l_{i,j,k'}$$

- Gram matrix: inner product between channels

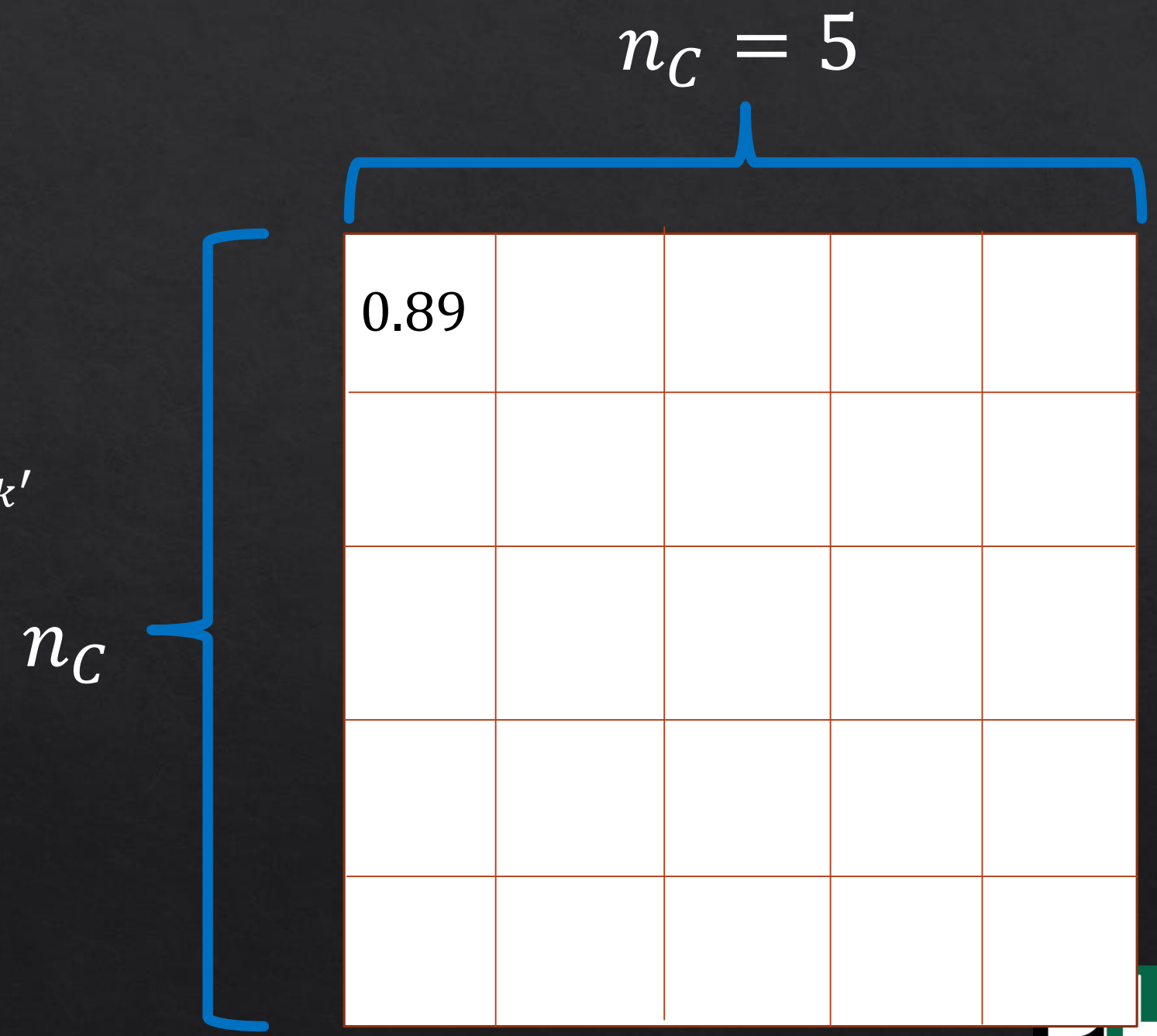
Style loss – gram matrix

Style Image



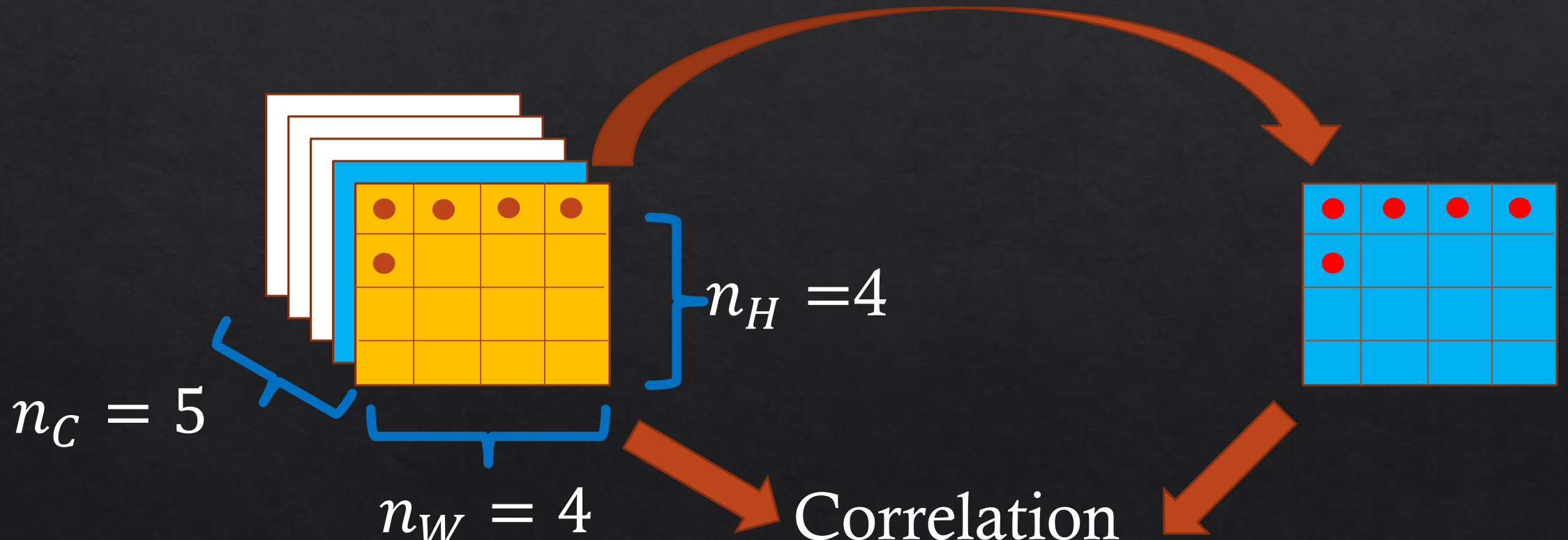
Style loss – gram matrix

$$G_{k,k'}{}^{l} = \sum_{i,j} a^{l}_{i,j,k}\, a^{l}_{i,j,k'}$$



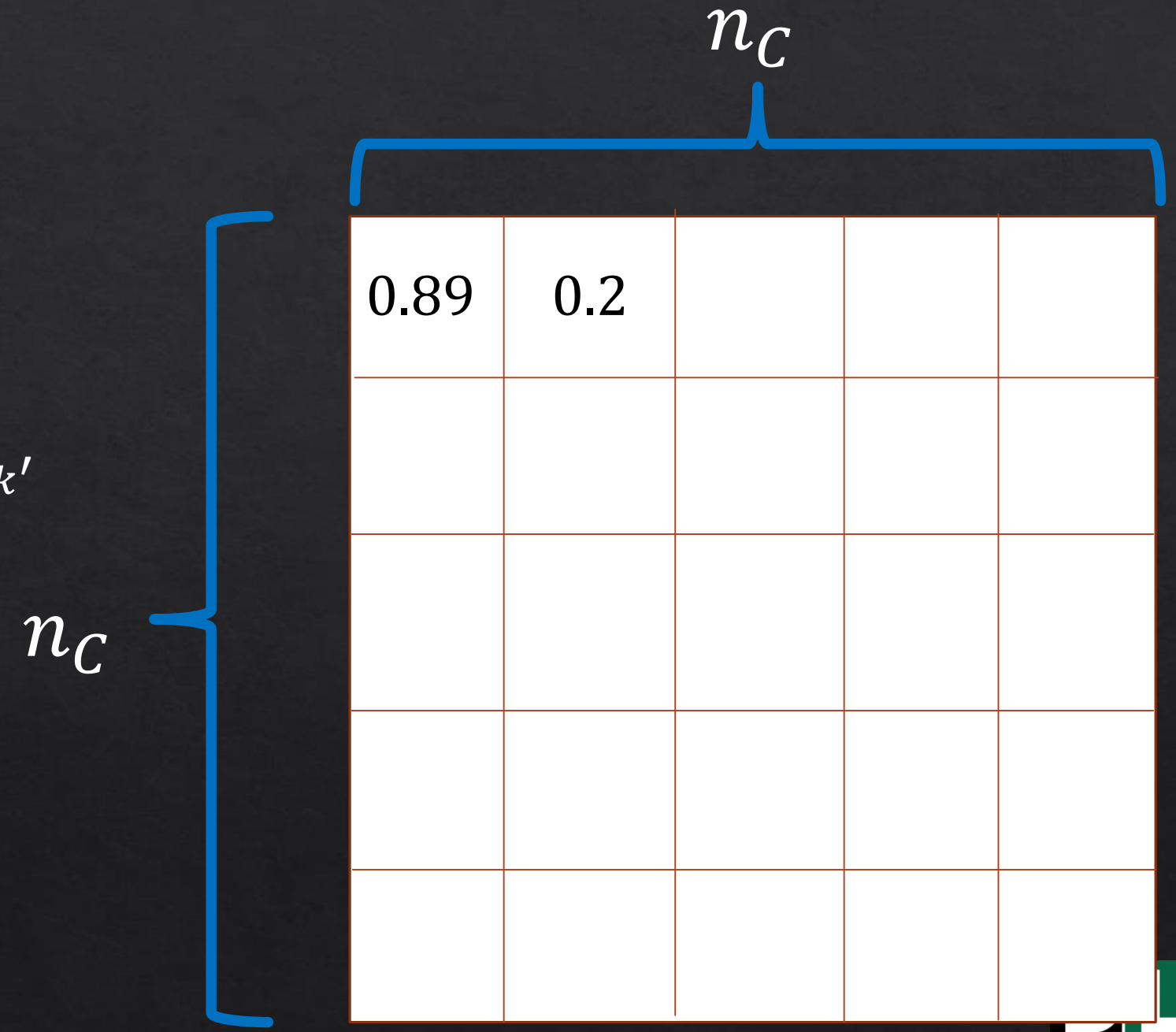
Style loss – gram matrix

Style Image



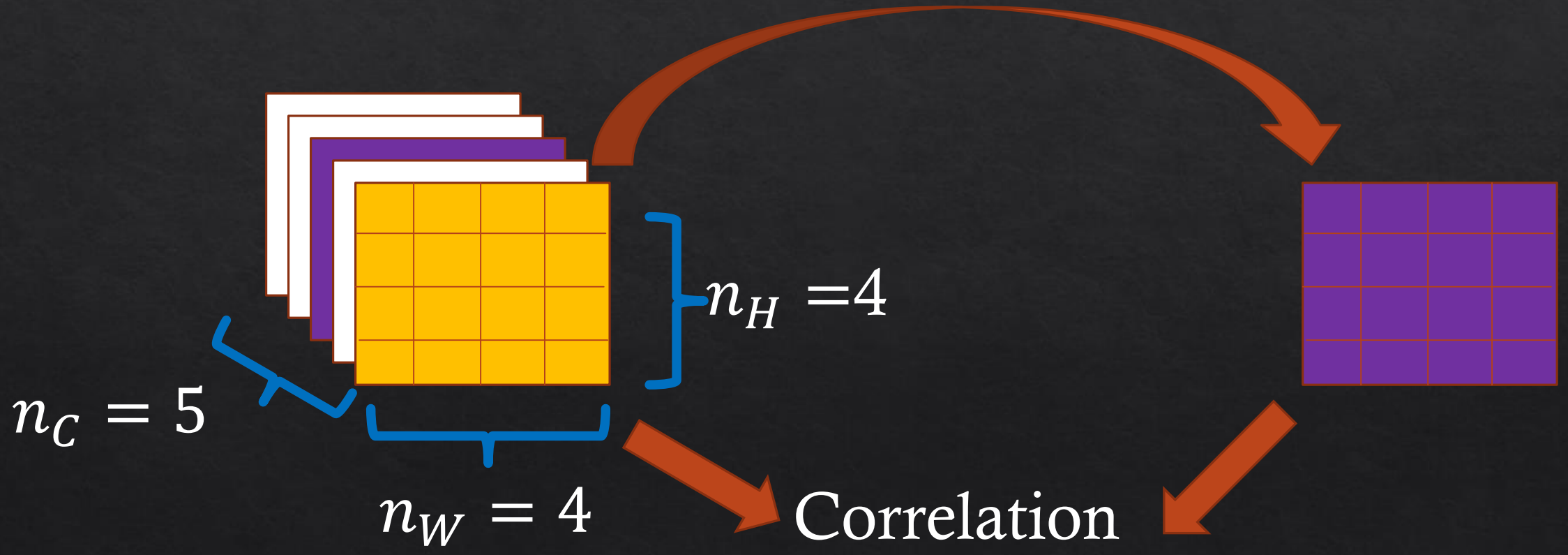
Style loss – gram matrix

$$G_{k,k'}{}^{l} = \sum_{i,j} a^{l}_{i,j,k}\, a^{l}_{i,j,k'}$$



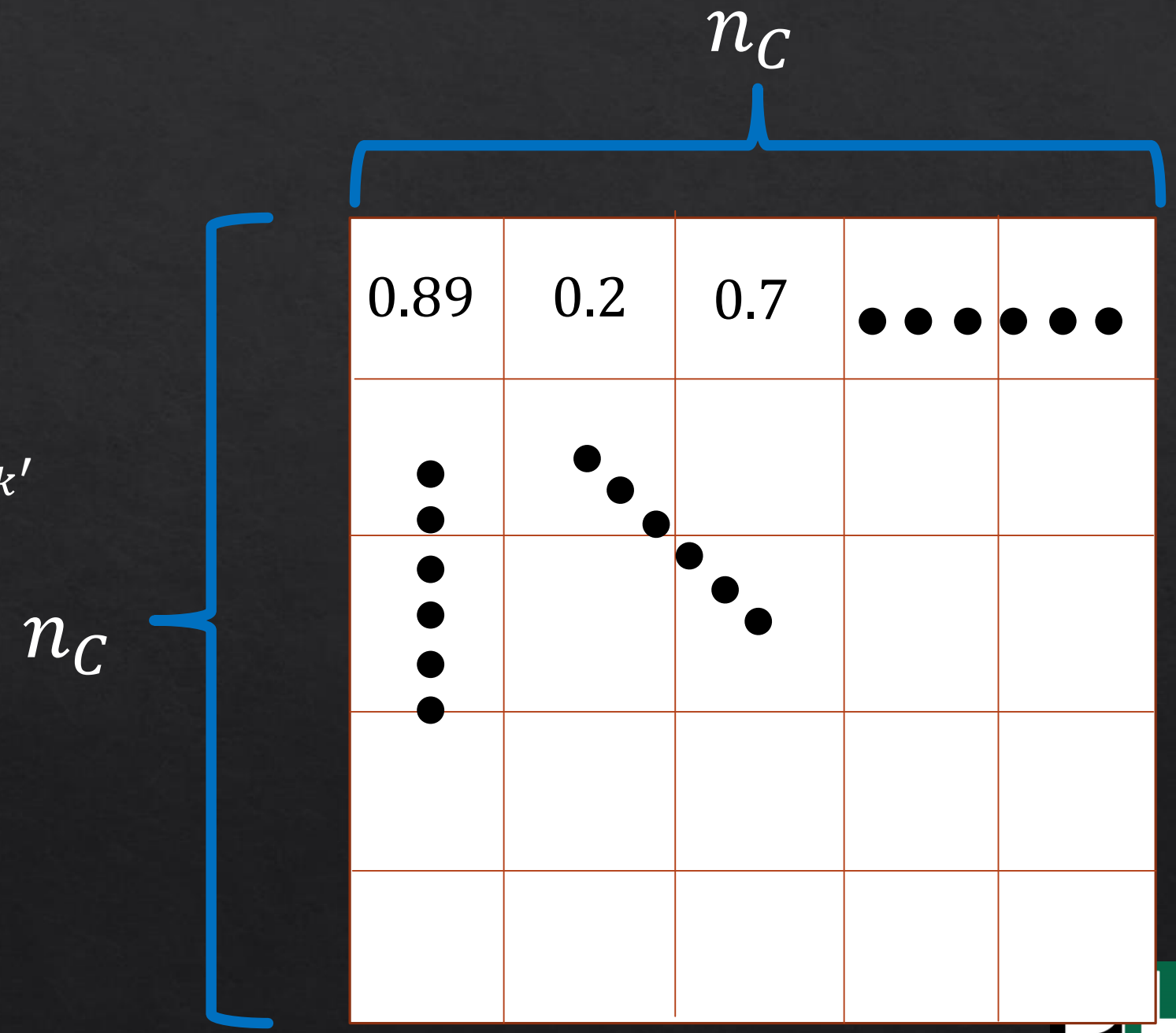
Style loss – gram matrix

Style Image



Style loss – gram matrix

$$G_{k,k'}{}^{l} = \sum_{i,j} a^{l}_{i,j,k}\, a^{l}_{i,j,k'}$$



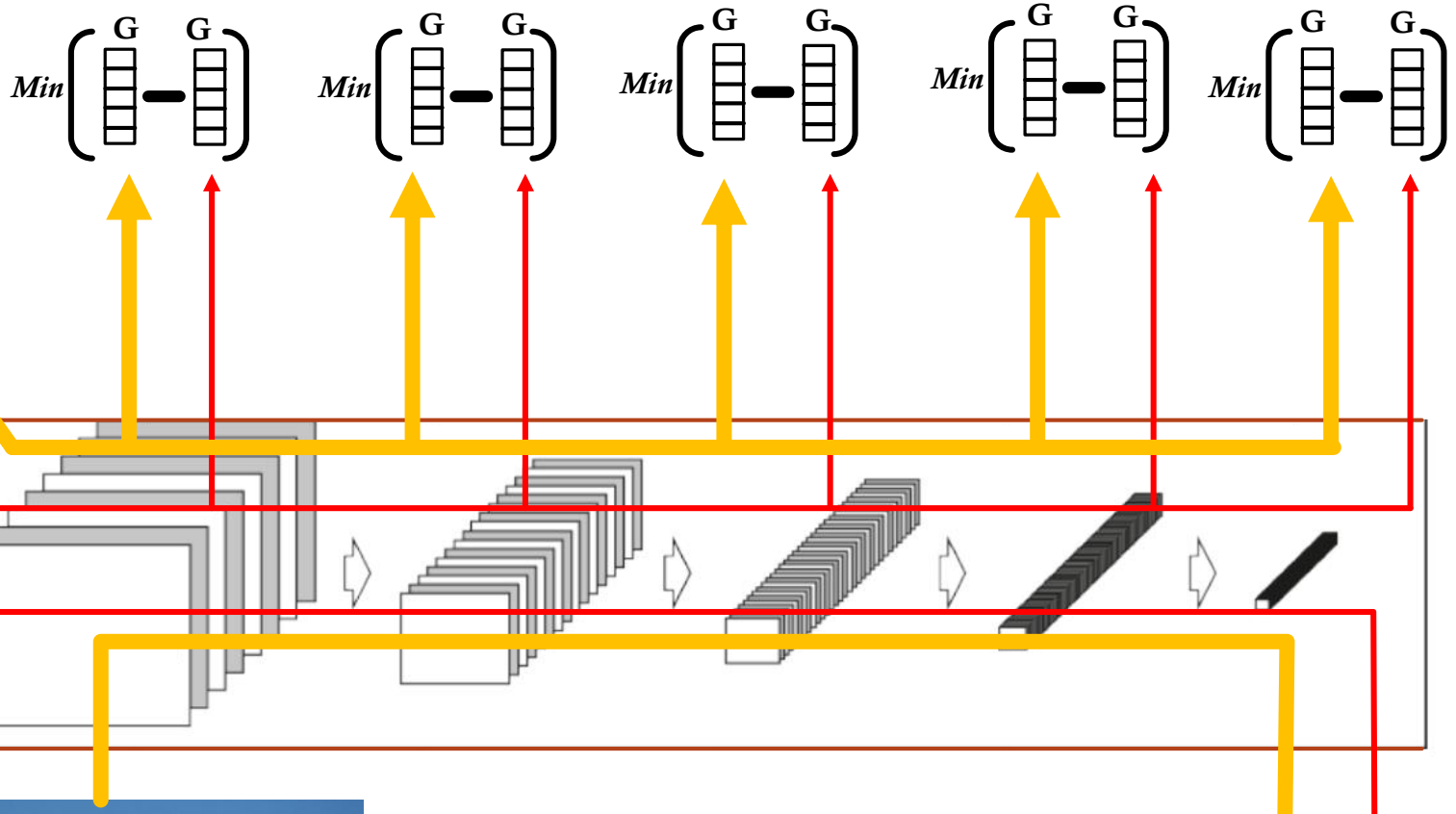
Style loss

$$L_{style}(\vec{s}, \vec{x}, l) = \frac{1}{(2n_W n_H n_c)^2} \sum_{k,k',l} (G^l_{k,k'} - S^l_{k,k'})^2$$

White Noise Gram Matrix

Target Style Gram Matrix

- Gram matrix: inner product between channels



Minimize

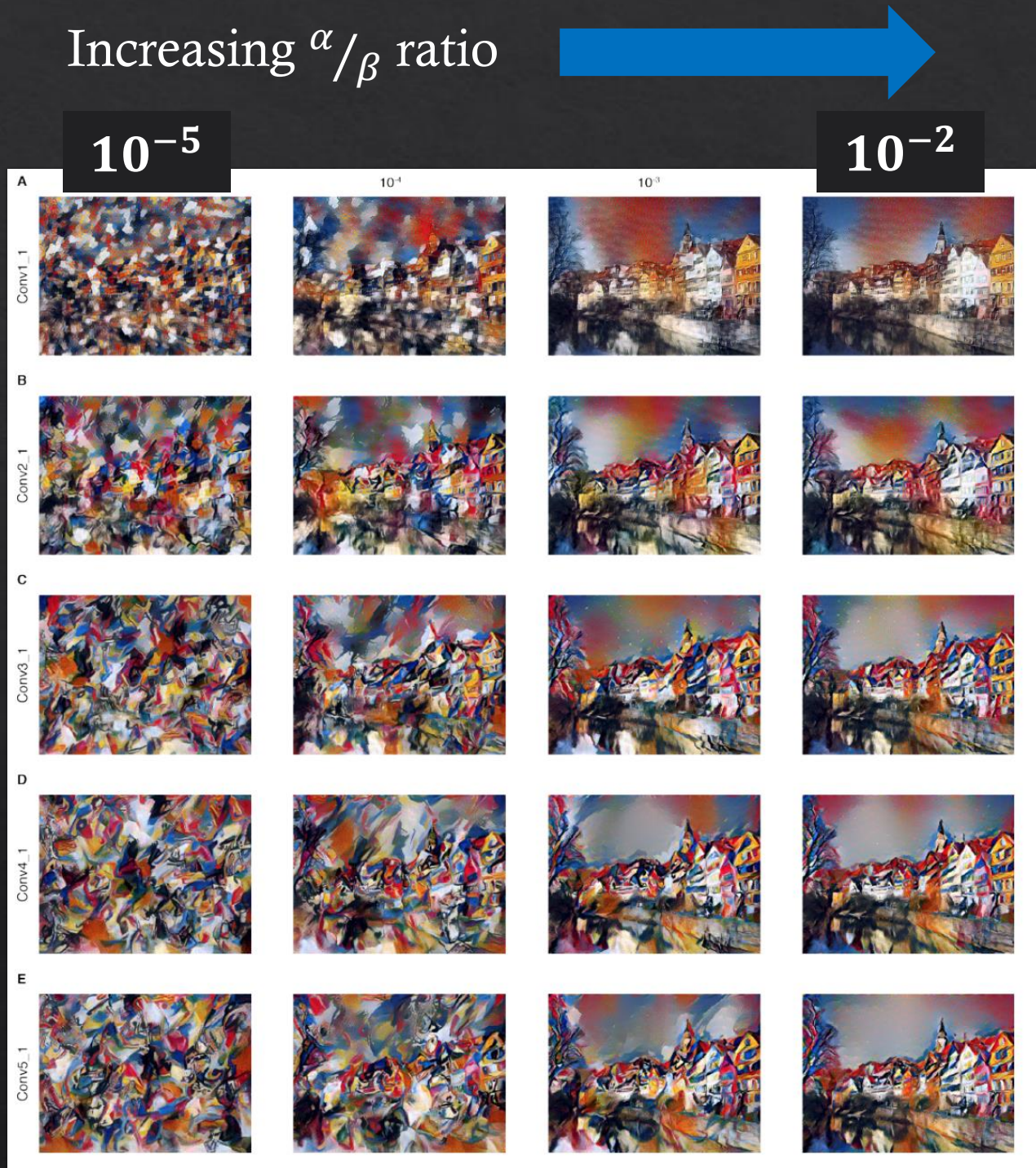
Target content

White noise

Varying layers and α/β

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

Deeper layers for style



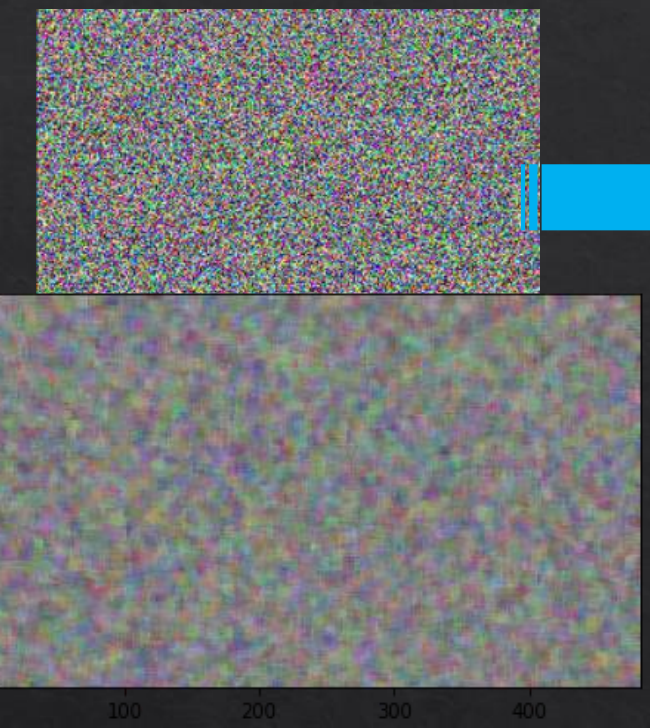
Optimizer - LBFGS

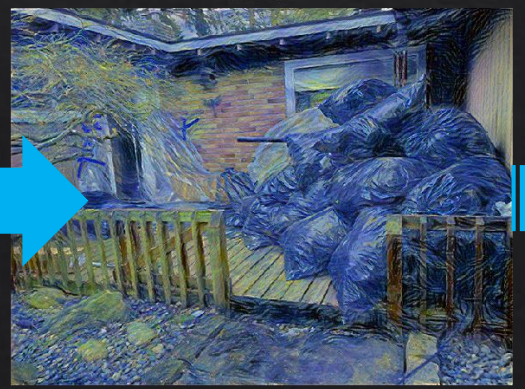
- Tracks the second derivative, hessian matrix
- Terrible if we are doing model learning, millions of weights
- But, works well for image approximation

Epoch 100

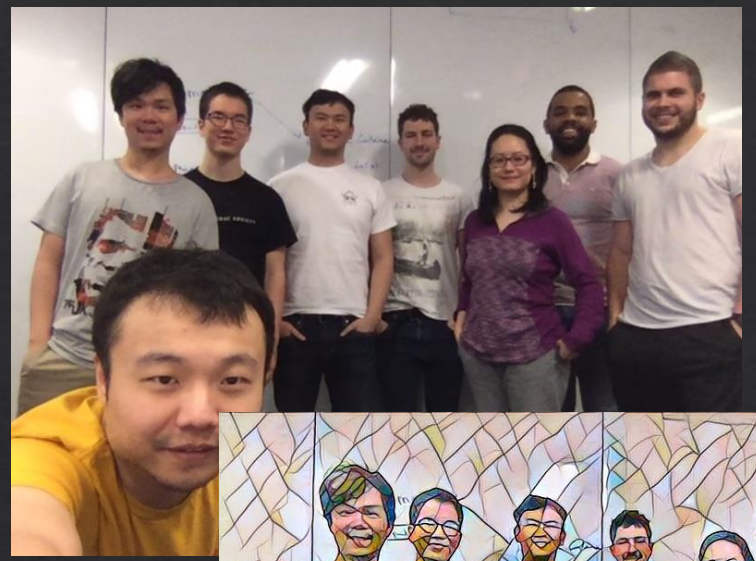
Epoch 300

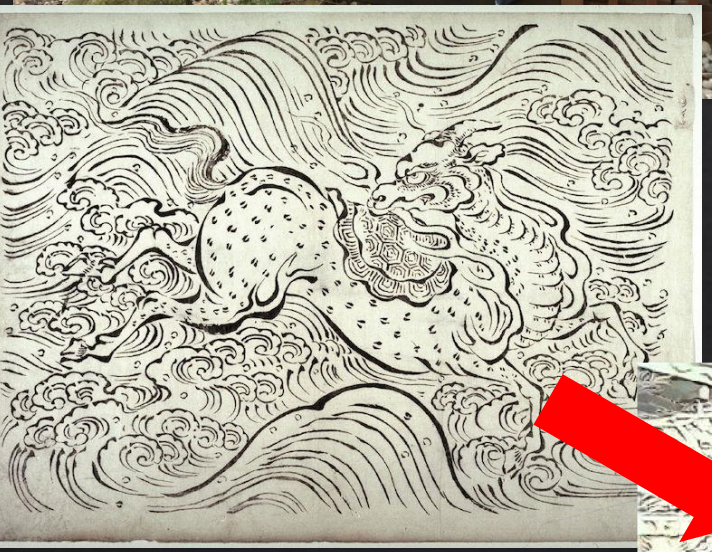
Epoch 500

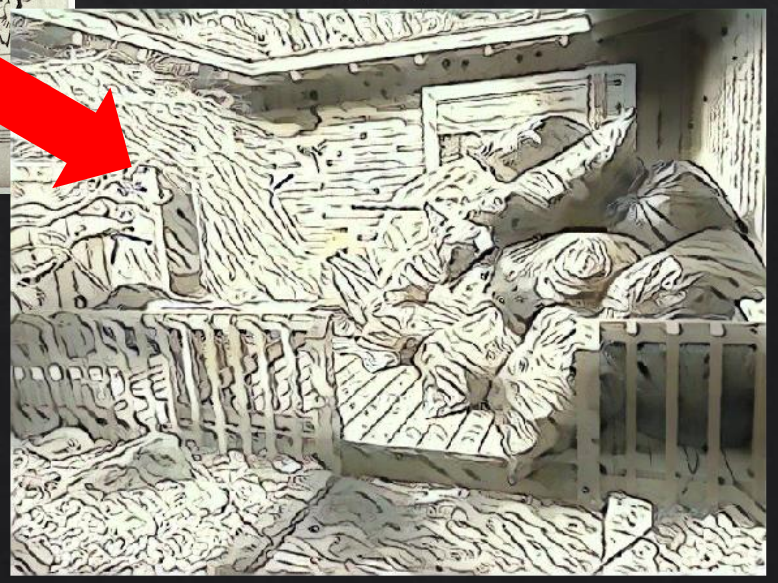


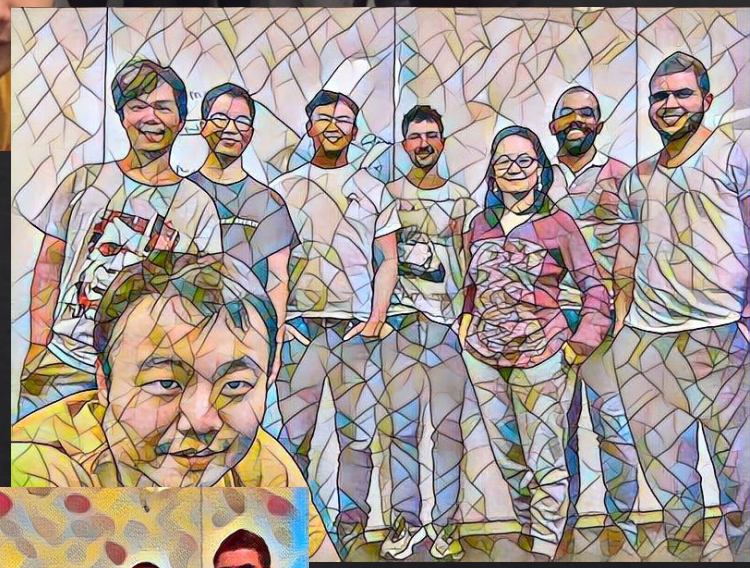


DLRL workshops









Summary

- Content and style are separable mathematically
- The first time that any arbitrary artistic style can be transferred
- In the past is all filter based/hand crafted techniques
- Does not need (content, style) pair ground truth

BREAK

Discussion

- Can we separate content/style for data other than images/videos, such as voice, transactions?
- Demystifying Neural Style Transfer: can we reformulate Gram Matrix to have more flexibility?

Appendix

Code

- Official: https://github.com/leongatys/PytorchNeuralStyleTransfer/blob/master/NeuralStyleTransfer.ipynb
- Major Unofficial: https://pytorch.org/tutorials/advanced/neural_style_tutorial.html
- Paper: https://arxiv.org/abs/1508.06576

Style loss – gram matrix

- Diagonal: which channels are most active
- Off-diagonal: which channel pairs co-occur