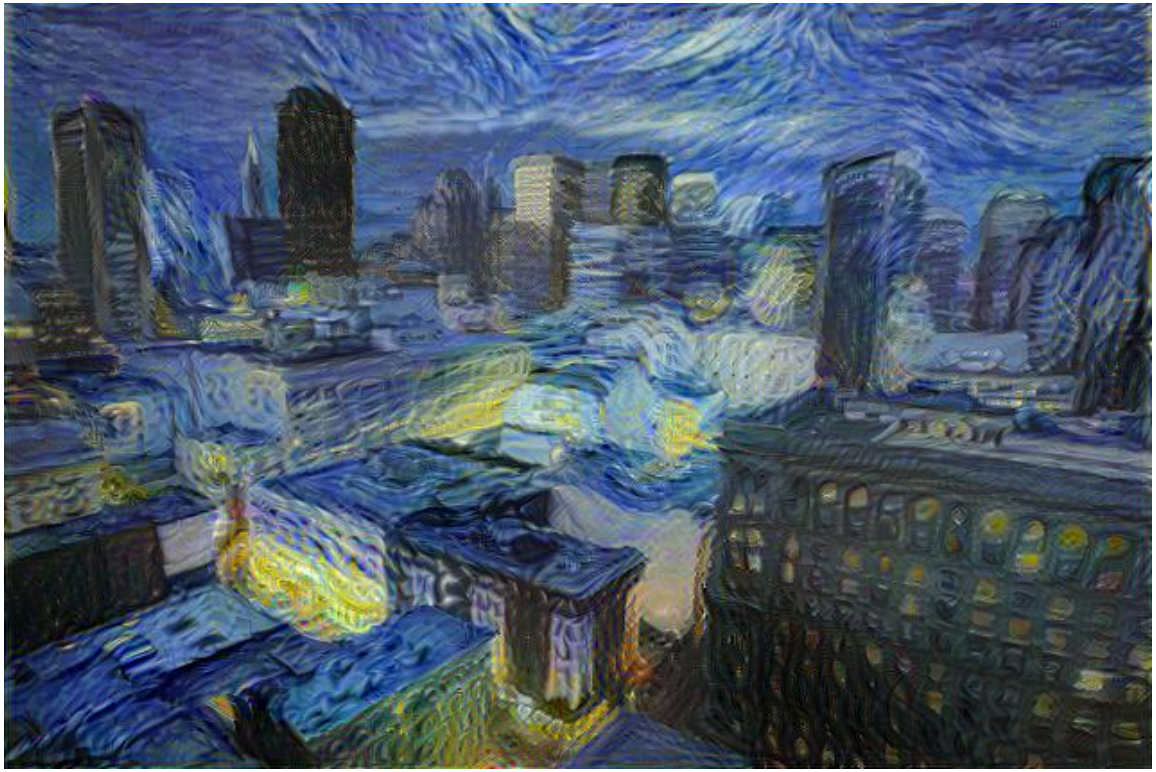


Image Style Transfer Using Convolutional Neural Networks

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

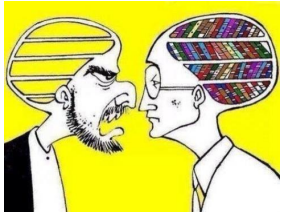
Originally created by: Nancy Iskander (niskander@dgp.toronto.edu)

Modified and Presented by: Wenjian Hu(wjhu@ucdavis.edu)

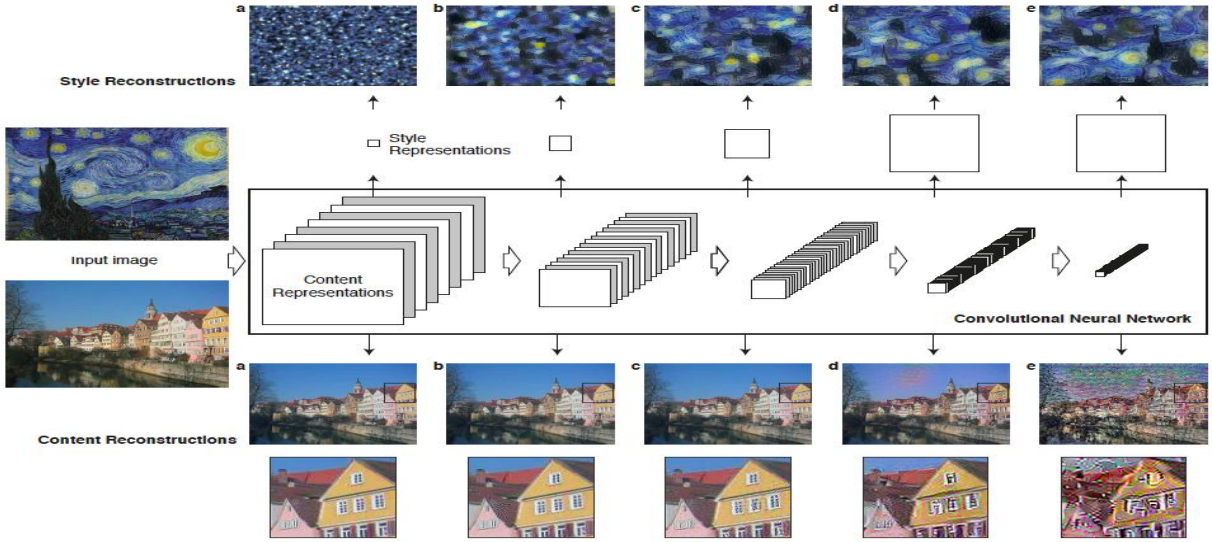
Overview of Method

- Content: Global structure. Style: Colours; local structures
  Like naturalistic, photographic, abstract, symbolistic



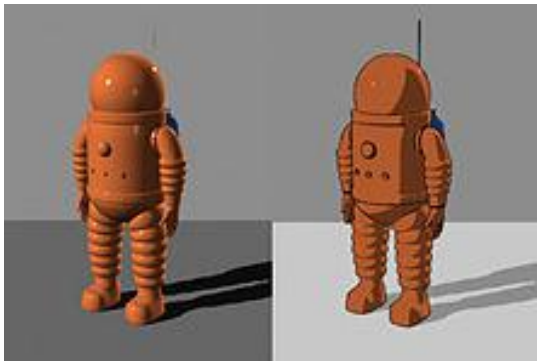
- Use CNNs to capture style from one image and content from another image.



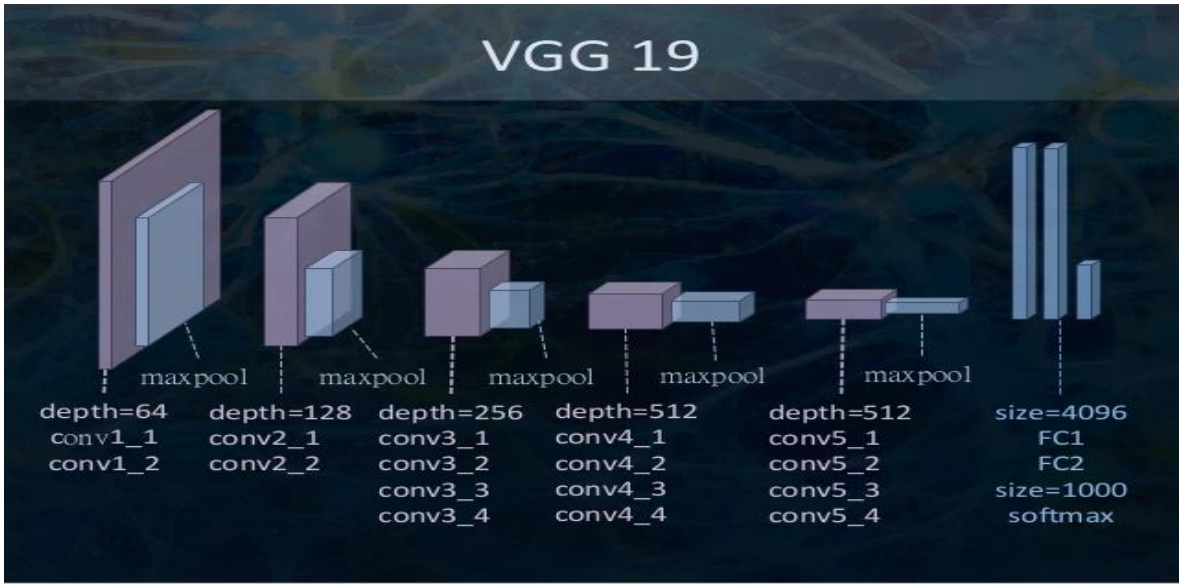
- Feature representations → Content
  Filter correlations → Style

Motivation for method

- Non photorealistic rendering(NPR) style/texture transfer methods



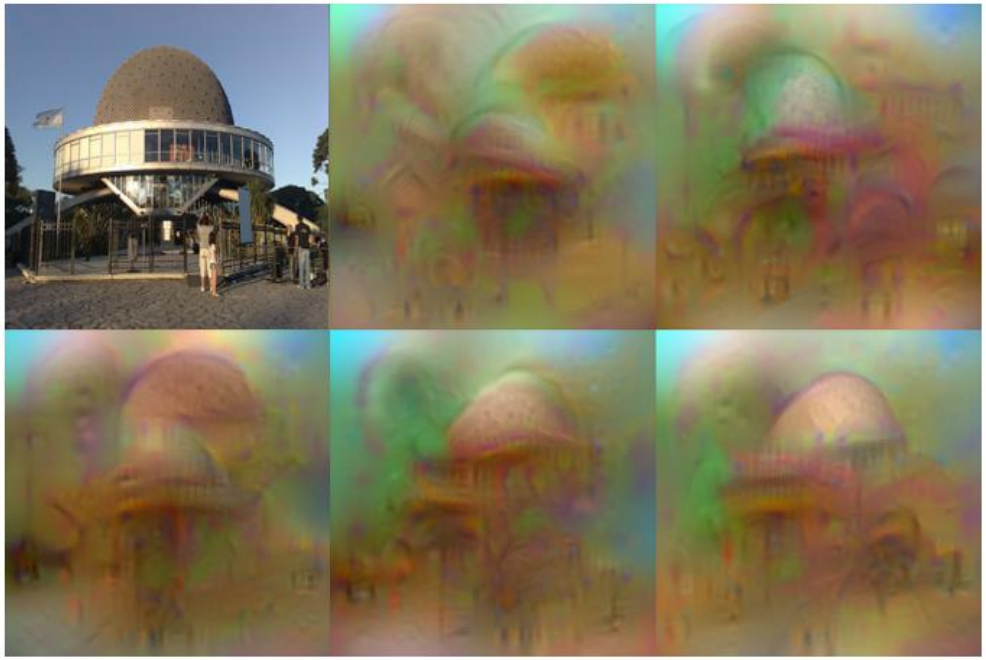
- Deep Neural Networks (VGG) manipulations in feature spaces



Reconstructing an image from a convolutional layer

- Representation function: $\Phi : \Re^{H \times W \times C} \rightarrow \Re^{d}$ (image space to feature space)
- Target Representation: $\Phi_0 = \Phi(x_0)$ (x0 is the original image)
- We need to find: $x \in \Re^{H \times W \times C}$ by minimizing:

$$x^* = \arg\min_{x \in \Re^{H \times W \times C}} l(\Phi(x), \Phi_0) + \lambda R(x)$$



"Understanding Deep Image Representations by Inverting Them", by Aravindh Mahendran and Andrea Vedaldi.

Content Reconstruction

Image reconstructed from layers
(a)'conv1_1',
(b)'conv2_1',
(c)'conv3_1',
(d)'conv4_1' and
(e)'conv5_1'
of the original VGG-Network

Content Loss Function

- Filters (Depths) at layer l: N_l
- The height times the width of the feature map at layer l: M_l
- Response at layer l: $F_l \in \Re^{N_l \times M_l}$

  $F^l{}_{ij}$ represents the ith filter at position j in layer l
- Original image: $\vec{p}$
- We generate image: $\vec{x}$ (randomly initialized)
- Squared-error loss:

$$L_{content} = \frac{1}{2}\sum_{i,j}(F^l{}_{ij} - P^l{}_{ij})^2$$

Style Reconstruction

Style representations (filer correlations) from:
(a) 'conv1_1',
(b) 'conv1_1', 'conv2_1',
(c) 'conv1_1', 'conv2_1', 'conv3_1',
(d) 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1',
(e) 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1', 'conv5_1'.

Style Loss Function

- Filter correlations are given by the Gram matrix:

$$G^l \in \Re^{N_l \times N_l}$$

- $G^l{}_{ij}$ is the inner product between the filters i and j in layer l:

$$G^l{}_{ij} = \sum_k F^l{}_{ik} F^l{}_{jk}$$

- The loss at layer l:

$$E_l = \frac{1}{4N^2{}_l M^2{}_l} \sum_{i,j} (G^l{}_{ij} - A^l{}_{ij})^2$$

A <-> original image
G <-> generated image

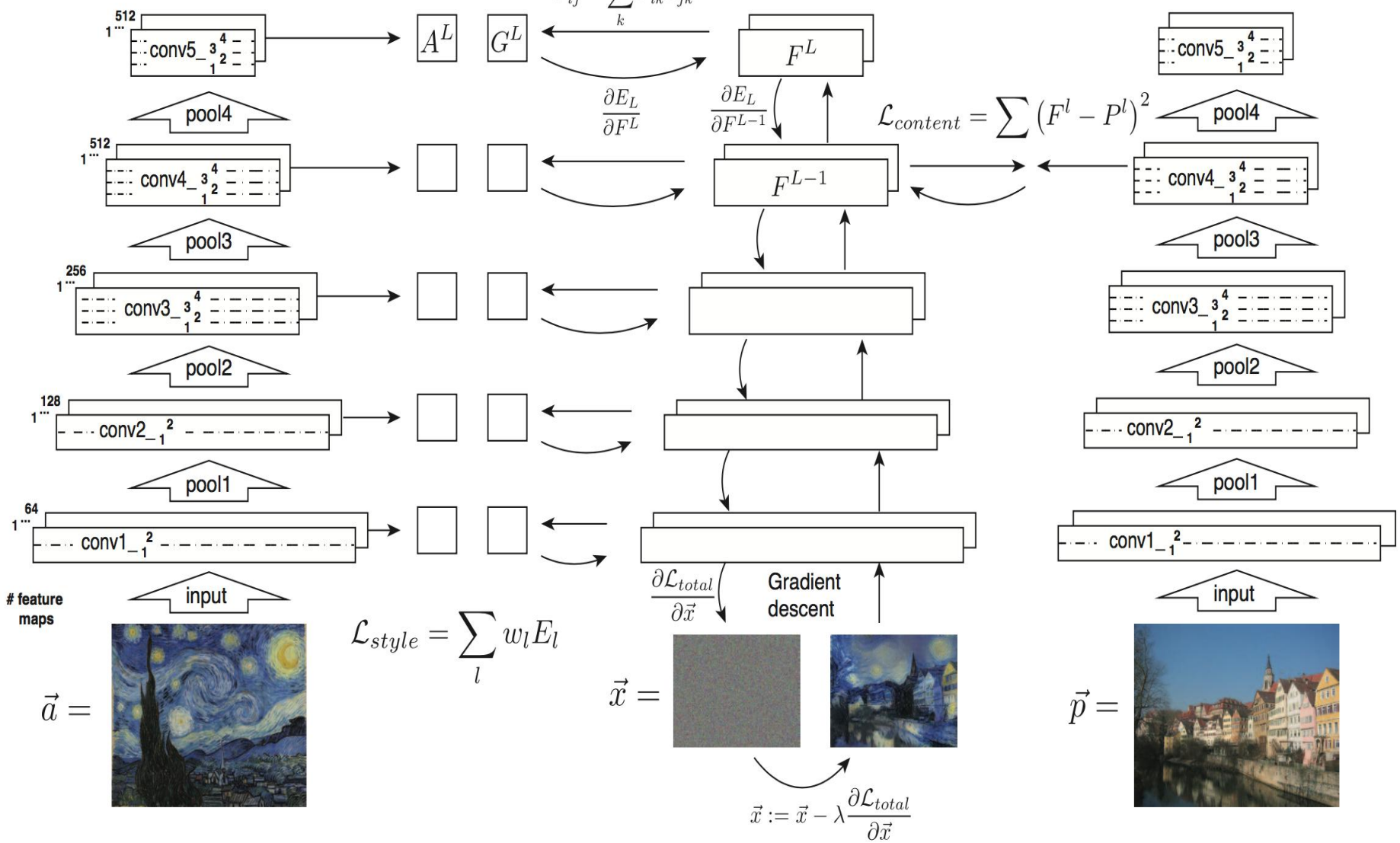
- The total style loss:

$$L_{style} = \sum_{l=0}^{L} w_l E_l$$

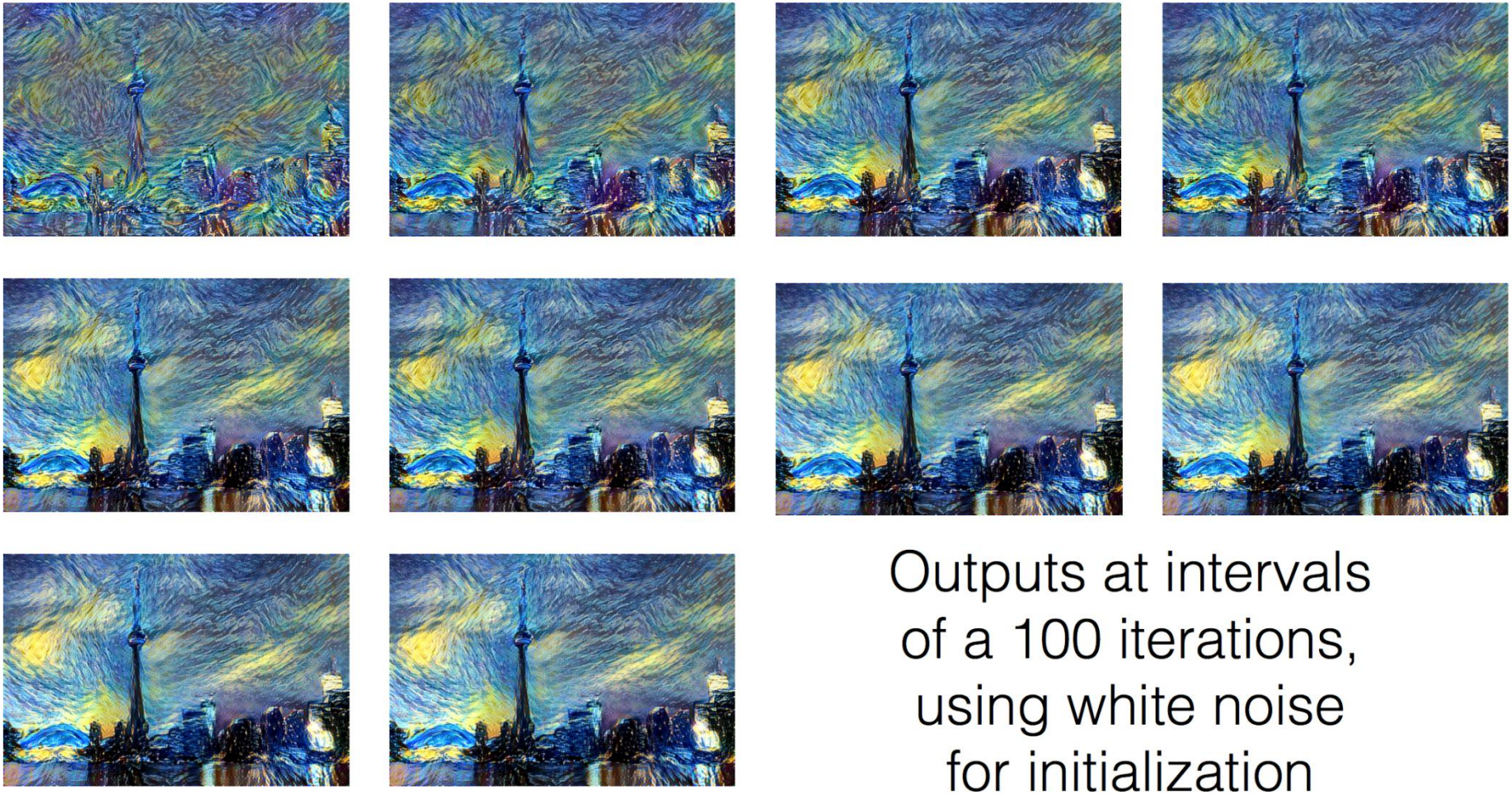
The Total Loss Function

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

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



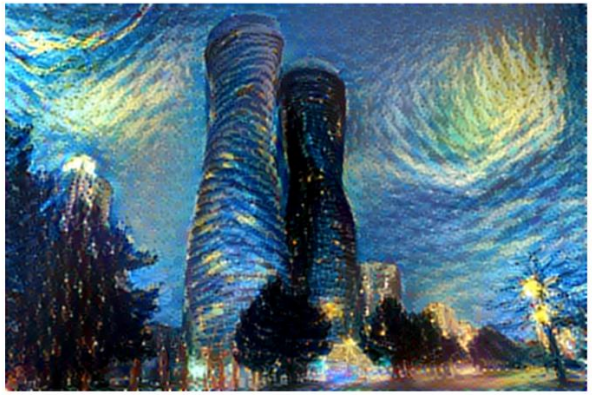
$$\mathcal{L}_{style} = \sum_l w_l E_l$$



Outputs at intervals of a 100 iterations, using white noise for initialization

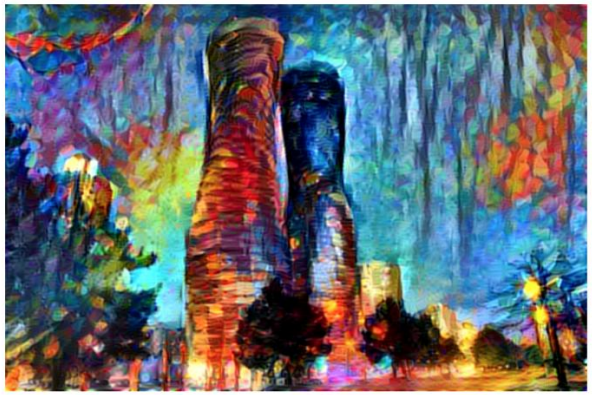
show image every 10 iterations

Content image

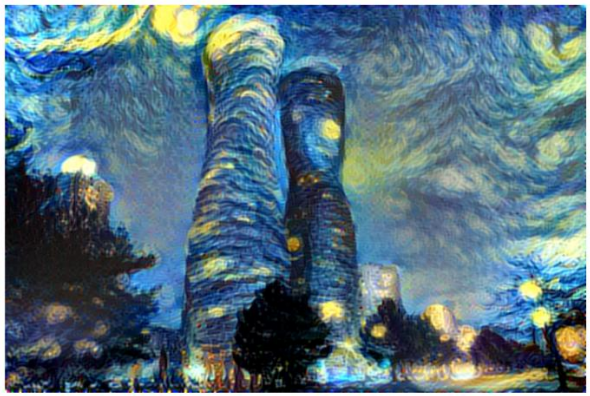


Large scale of cropped Starry Night as style image (emphasizes dark foreground)

Large scale of full Starry night as style image, initialized with content image

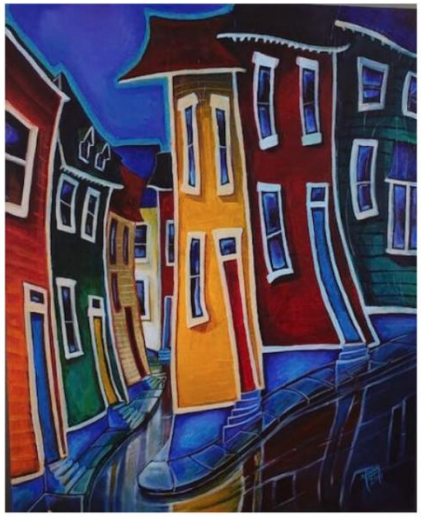


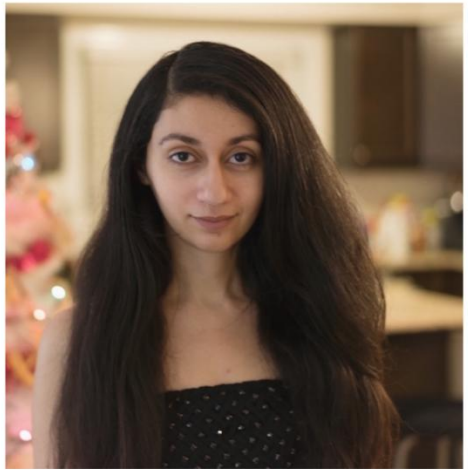
Using Leonid Afremov painting as style image

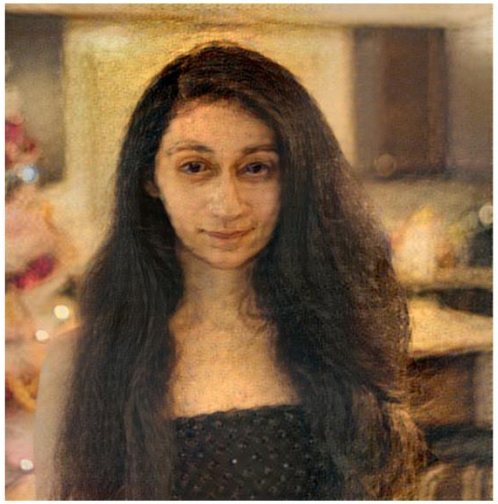


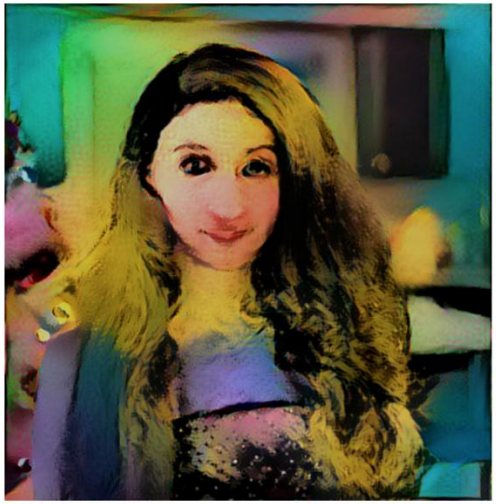
Smaller scale of style (using convolution layers closer to the input layer)

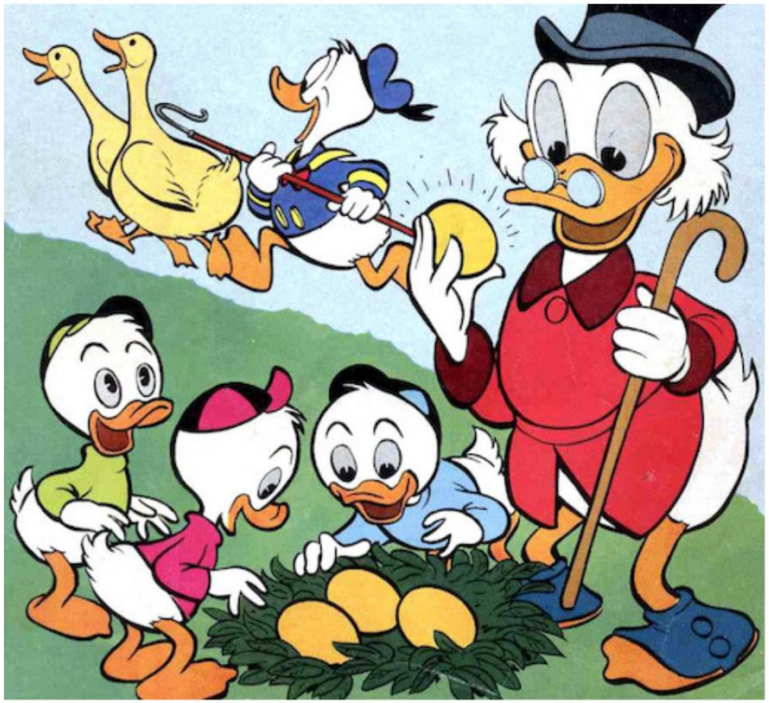
Large scale of full Starry night as style image, initialized with white noise

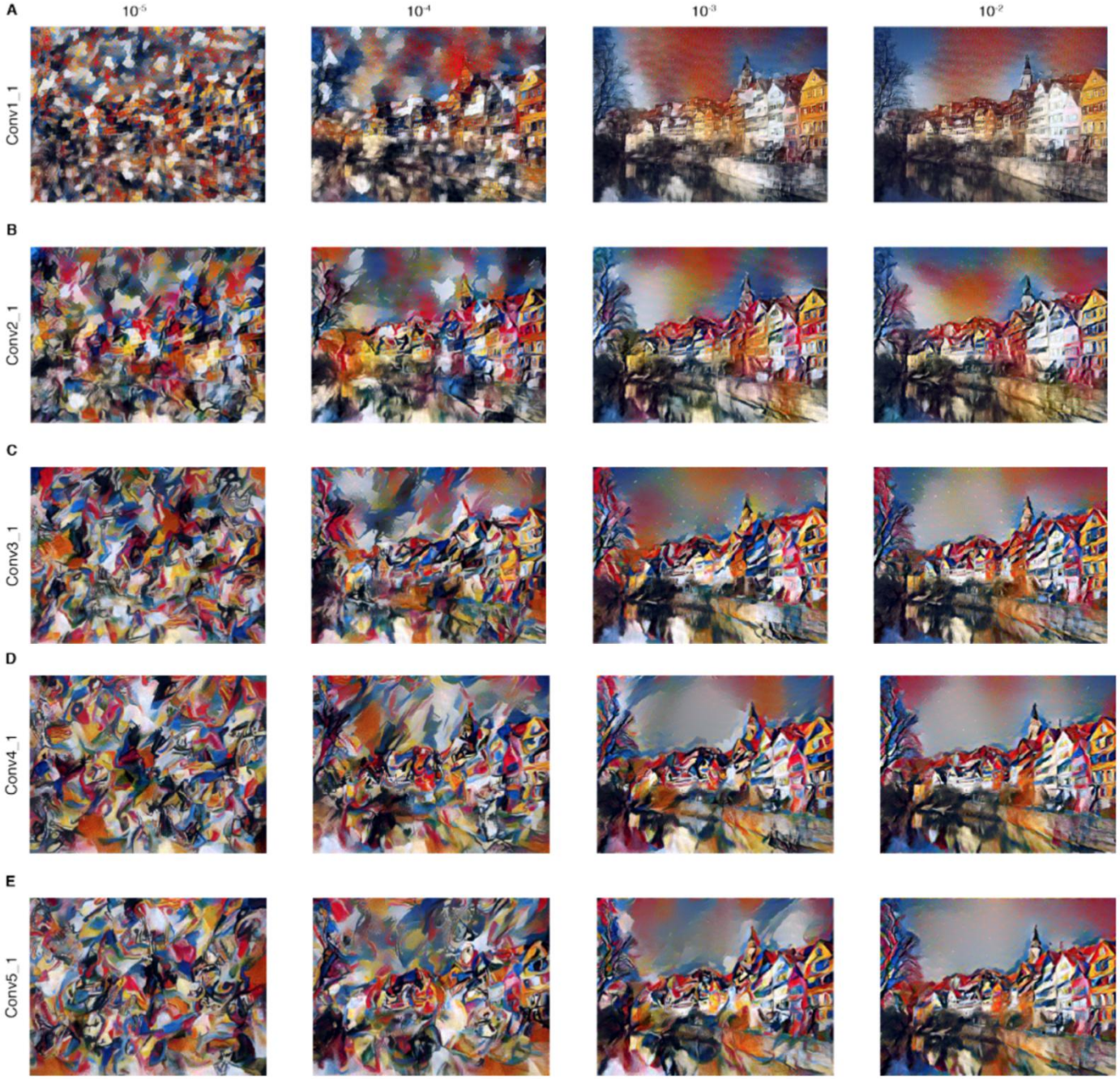











A
10^{-5}
10^{-4}
10^{-3}
10^{-2}
Conv1_1
B
Conv2_1
C
Conv3_1
D
Conv4_1
E
Conv5_1

Conclusion

- The method appears to work very well and is easy to implement.

- New method of mixing content and style from different sources.

- Useful for studying the neural representation of art, style and content-independent image appearance.

Thank you!!