Practices in visual computing 2

Lab1: Style Transfer

Simon Fraser University
Spring 2025

TAs

Amir Alimohammadi (MSc in CS)

aaa324@sfu.ca

Sai Raj Perla (PhD in CS)

srp7@sfu.ca

Assignments

1. Pose Regression with PoseNet (31 Jan)
2. Image to image translation with CycleGAN (21 Feb)
3. Diffusion (14 Mar)
4. Neural fields or Deep fake (4 Apr)

The assignment release dates may be subject to change.

What is Style Transfer?

The process of blending the “style” (colors, textures, brush strokes) of one image onto the “content” (structure, shapes, layout) of another image.

Use Cases: Creating artistic renditions of photos, design mockups, enhancing images, etc.

Key Idea: We keep the content of one image while borrowing the style from another.

High-Level Process

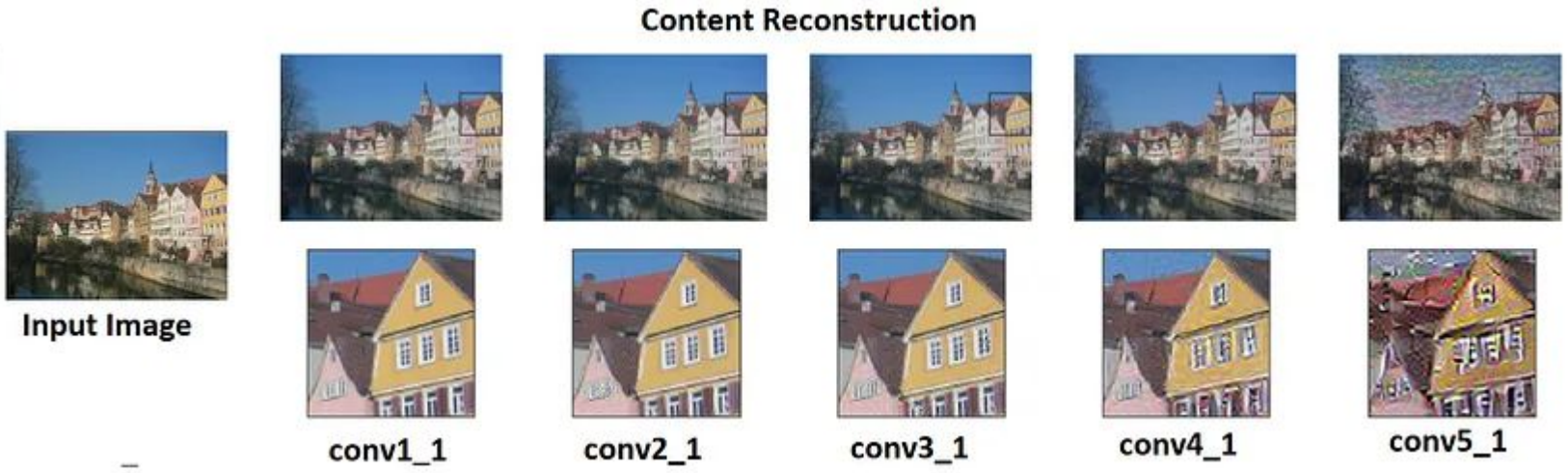
Content Image: The image whose structure and layout we want to preserve.

Style Image: The image whose color and texture we want to use as style.

Generated Image: An image that retains the content image’s structure but is stylized like the style image.

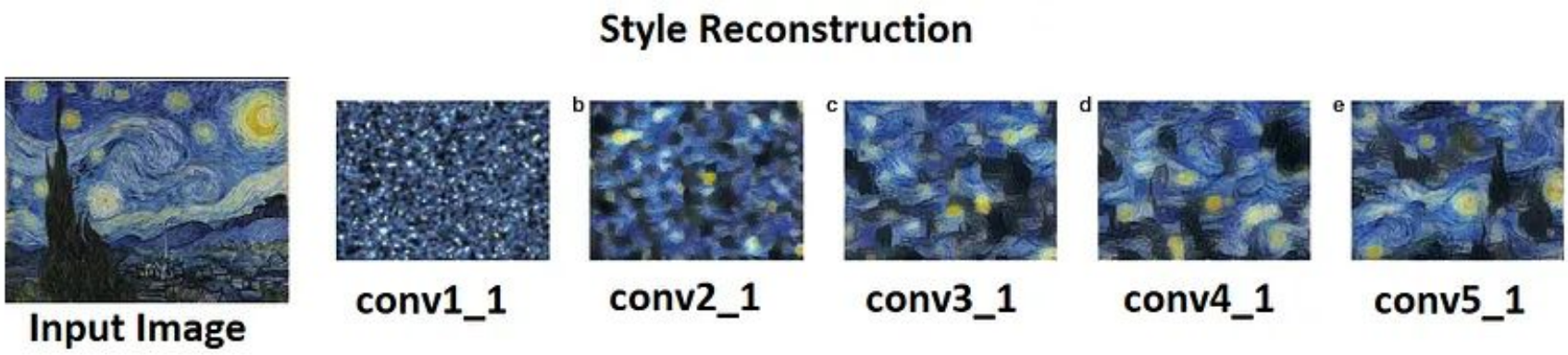
Content Representation

CNNs process images hierarchically: lower layers capture fine details, while higher layers focus on abstract content, forming high-level representations.

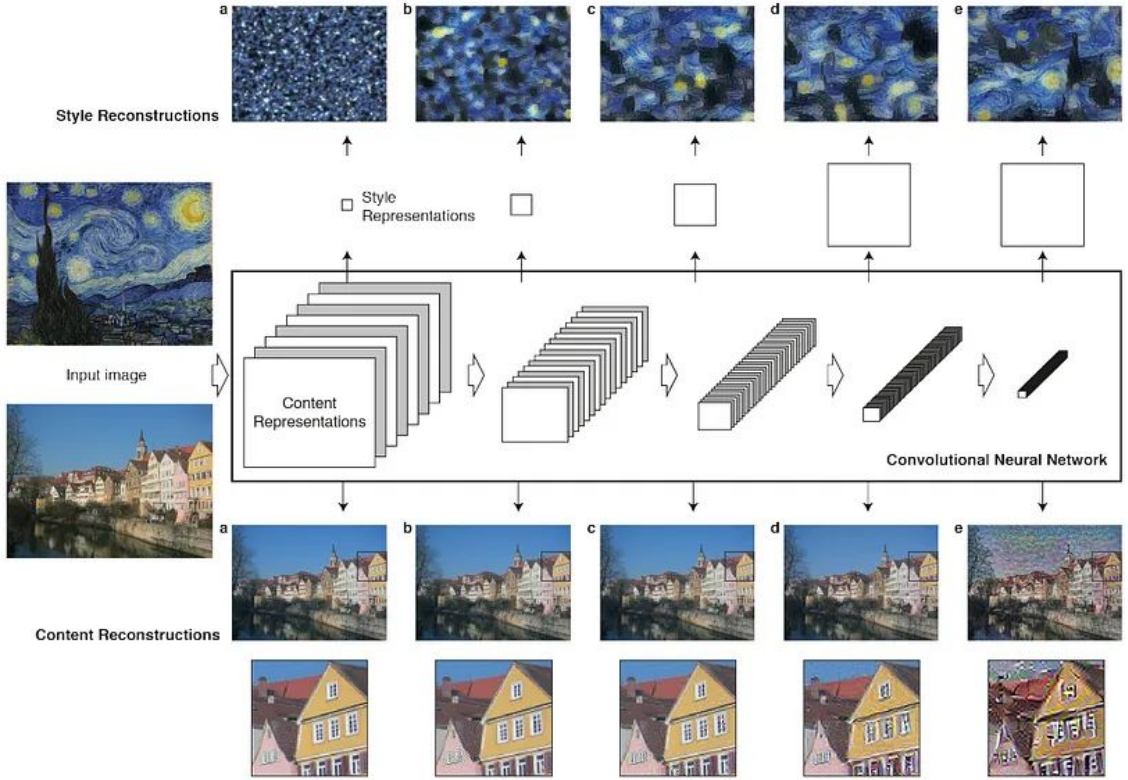


Style Representation

Style representation in CNNs captures texture information by analyzing correlations between filter responses across layers, creating multi-scale representations of an image's style while ignoring its global arrangement.



The Model Architecture



A Neural Algorithm of Artistic Style, L. Gatys et al (2015)

Loss Function

Adjust two weights, α (content) and β (style), to control the final outcome. Increasing β emphasizes the artwork's appearance, making the result look more like the painting's texture.

Increasing α preserves details from the photograph, making it more recognizable.

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

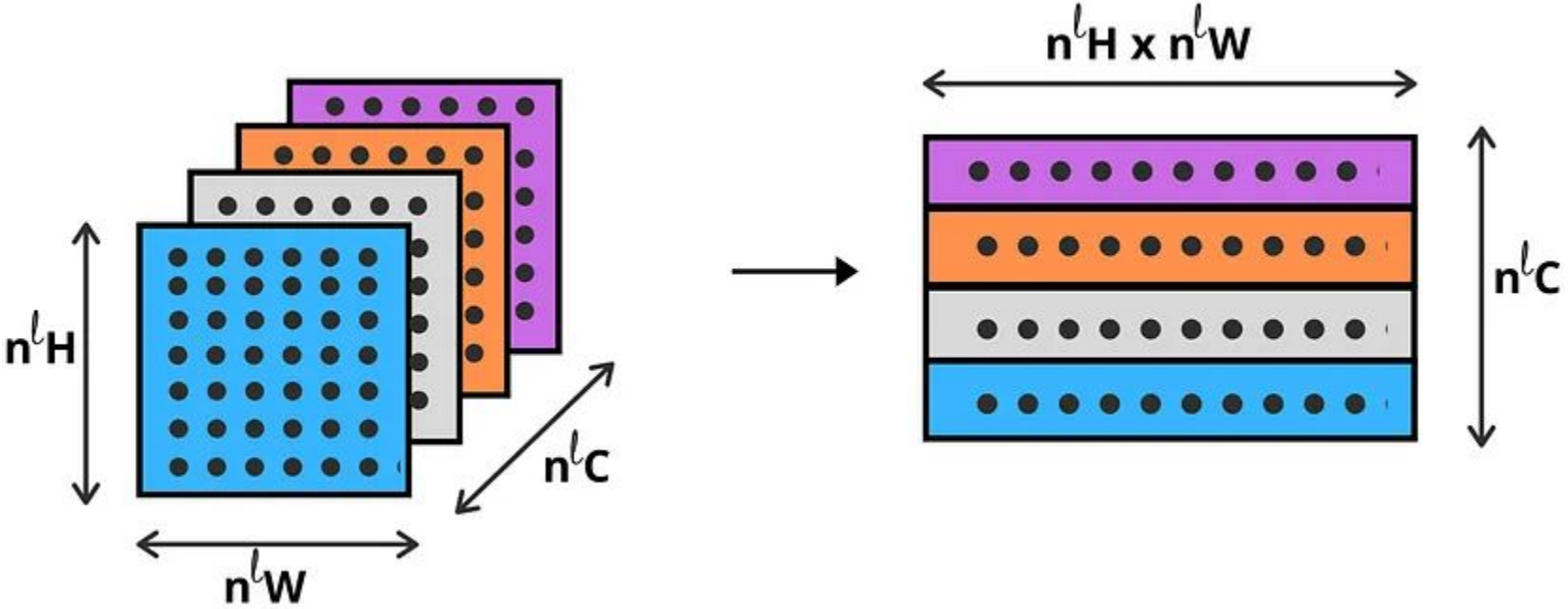
Content Loss

Pass both the content image and the base image through the CNN, extracting their intermediate feature maps.

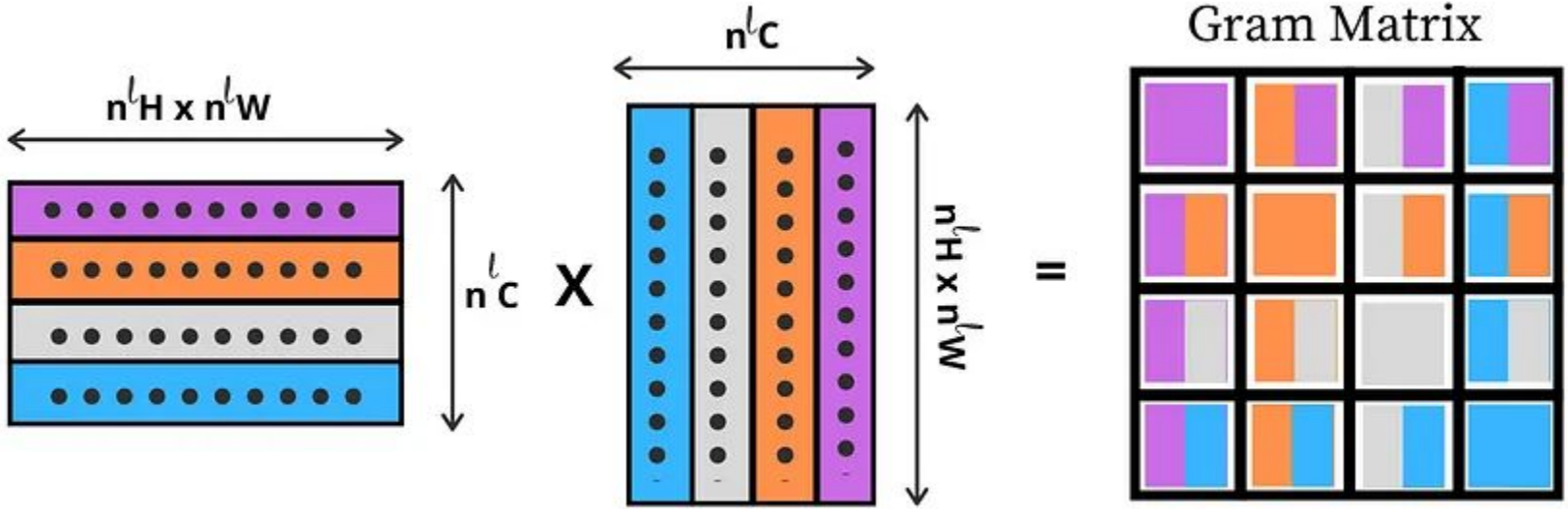
Compare these feature maps by calculating the squared error (Euclidean distance) between them.

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

Unrolling the 3D feature maps to 2D



Feature Correlations with Gram Matrix



Feature Correlations with Gram Matrix

The Gram matrix captures correlations between filter responses in a layer.

Calculated as the dot product of the unrolled feature map and its transpose.

Dimension: C*C (number of channels in layer l).

G and A are gram matrices of the features of the style image and the input image.

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

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

Final Result

Implementation Details

Model: VGG-19 (pretrained, layers: 0, 5, 10, 19, 28)

Epochs: 7000

Optimizer: Adam

Learning Rate: 0.004

Content Weight (α): 8

Style Weight (β): 70

Reference

https://towardsdatascience.com/implementing-neural-style-transfer-using-pytorch-fd8d43fb7bfa

A Neural Algorithm of Artistic Style, L. Gatys et al (2015)