CVPR 2016

Image Style Transfer Using Convolutional Neural Networks

Leon A. Gatys

Centre for Integrative Neuroscience, University of T¨ ubingen, Germany

Bernstein Center for Computational Neuroscience, T¨ubingen, Germany

Graduate School of Neural Information Processing, University of T¨ ubingen, Germany

Alexander S. Ecker

Centre for Integrative Neuroscience, University of T¨ ubingen, Germany

Bernstein Center for Computational Neuroscience, T¨ ubingen, Germany

Max Planck Institute for Biological Cybernetics, T¨ ubingen, Germany

Baylor College of Medicine, Houston, TX, USA

Matthias Bethge

Centre for Integrative Neuroscience, University of T¨ ubingen, Germany

Bernstein Center for Computational Neuroscience, T¨ ubingen, Germany

Max Planck Institute for Biological Cybernetics, T¨ ubingen, Germany

Introduction: What is style transfer?

Merging **content** from one image with the artistic **style** of another

Image Source: [Tuebingen Neckarfront with beautiful old houses.]

Image Source: [The Starry Night by Vincent van Gogh, 1889]

Problem Definition: Why style transfer is challenging?

- Content and style are deeply intertwined. Separation of content and style of a natural image is extremely difficult.

- Traditional methods rely on low-level features of the target image to inform the texture transfer (synthesise photorealistic natural textures by **resampling** the pixels of a given source texture)

Solution: *A Neural Algorithm of Artistic Style*

A texture transfer algorithm that constrains a texture synthesis method by feature representations from state-of-the-art Convolutional Neural Networks.

Background: Hierarchical Features Through CNNs

CNNs learn hierarchical features through:

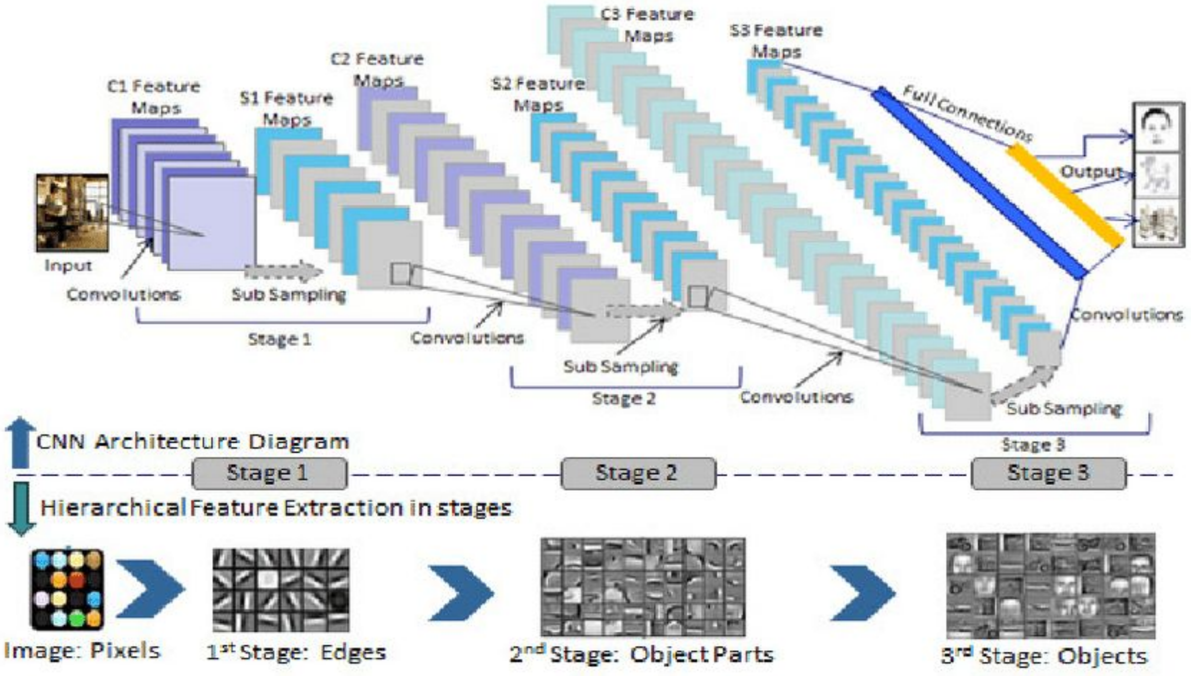


Image Source: Learning hierarchy of visual features in CNN architecture

- Lower layers: Basic pixel-based attributes (edges, colors, gradients etc.)
- Higher layer: High level details (objects & shapes, scenes & context, faces & expressions etc.)

- The key insight to this paper: CNNs can encode texture (style) and object structure (content) separately.

Content Representation

- Content representations of images are extracted from higher layers of a pre-trained CNN (VGG-19 with 16 convolutional layers and 5 pooling layers).
- Content informations are encoded within higher layers of the CNN as feature maps.
- To visualize these content information encoded at different layers, gradient descent can be performed on a white noise image, using the following loss function.

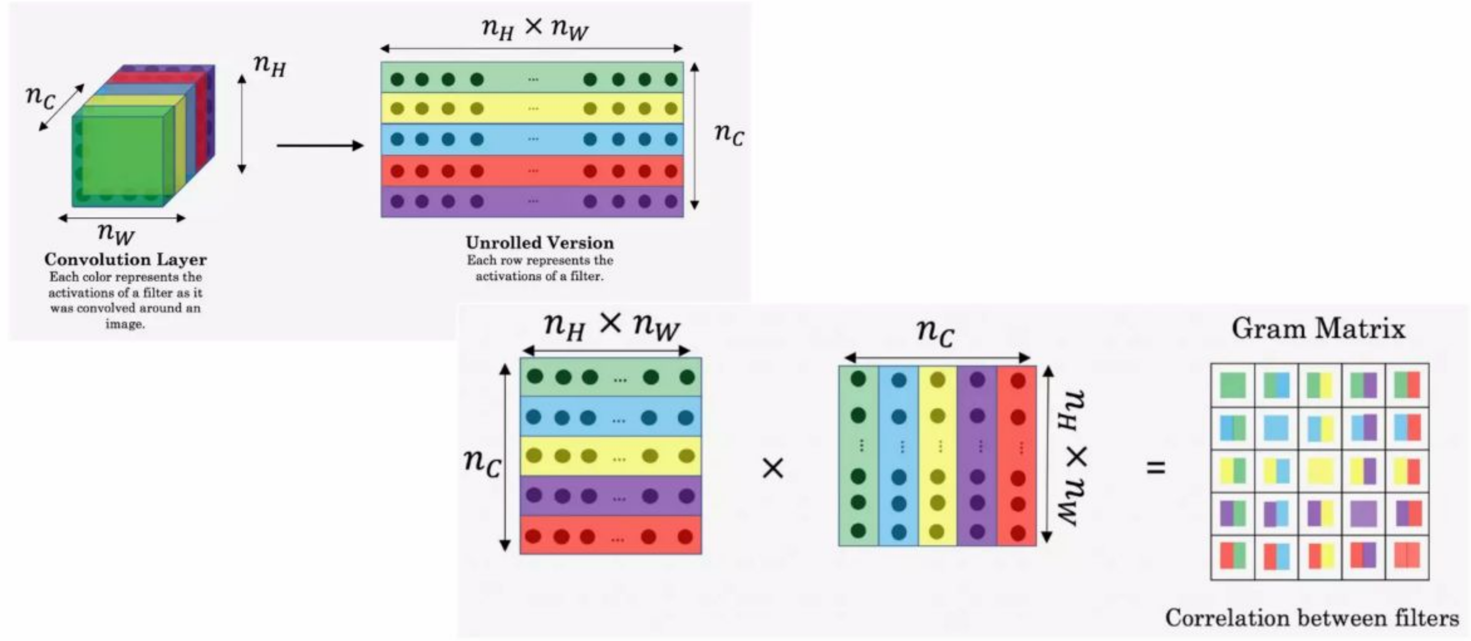
$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}\left(F^l_{ij} - P^l_{ij}\right)^2$$

F^l_{ij}: feature representation (of the i^{th} filter at j^{th} position) of the generated image in the l^{th} layer.

P^l_{ij}: feature representation (of the i^{th} filter at j^{th} position) of the original content image in the l^{th} layer.

Style Representation

- A feature space that captures texture information was designed to obtain style representation of an image.
- The feature space consists with the correlations between different filters of a layer.
- Feature correlations are stored in a Gram matrix.



Multiple layers contribute to capturing stationary, multi-scale texture patterns.

Image Source: Demystifying Neural Style Transfer

Style Transfer Algorithm

- Synthesise a new image that simultaneously matches content from one image and the style of an artistic image.

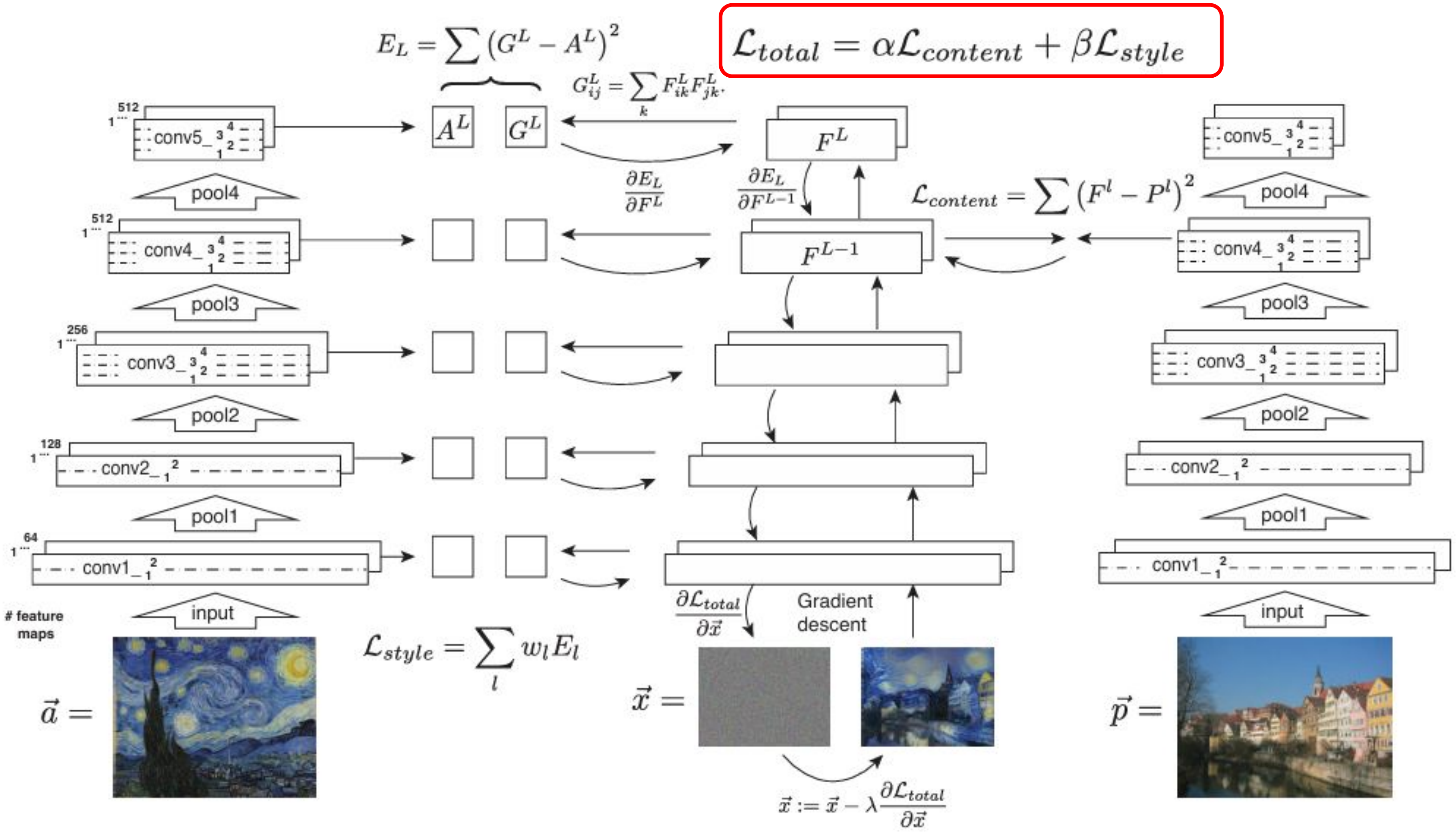


Image Source: [Style Transfer Algorithm]

Style Transfer Equation

- The total loss is calculated by:

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

- Tuning parameters:

  1. High **α**: more content preserved
  2. High ***β***: stronger style influence

- Optimization Method: L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)

Results of Style Transfer

- Key findings:
  a. Content and style representation of a CNN is well separable (in theory).
  b. We can smoothly regulate the emphasis on either reconstructing the content or the style (by manipulating $\boldsymbol{\alpha}/\boldsymbol{\beta}$ ratio).

Figure 3. Images that combine the content of a photograph with the style of several well-known artworks. The images were created by finding an image that simultaneously matches the content representation of the photograph and the style representation of the artwork. The original photograph depicting the Neckarfront in Tübingen, Germany, is shown in **A** (Photo: Andreas Praefcke). The painting that provided the style for the respective generated image is shown in the bottom left corner of each panel. **B** *The Shipwreck of the Minotaur* by J.M.W. Turner, 1805. **C** *The Starry Night* by Vincent van Gogh, 1889. **D** *Der Schrei* by Edvard Munch, 1893. **E** *Femme nue assise* by Pablo Picasso, 1910. **F** *Composition VII* by Wassily Kandinsky, 1913.

Image Source: [Images that combine the content of a photograph with the style of several well-known artworks.]

Effect of Content Layer Selection

- Matching from lower layers → more detailed pixel information retained, but the texture is merely blended over the image (weaker style transfer).
- Matching from higher layers → the fine structure of the image is altered such that it agrees with the style of the artwork (better fusion of content and style).

Effect of Style Layers

- Matching the style representation upto higher layers preserves local images structures an increasingly large scale.

- The style loss for a particular layer:

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

G^l_{ij}: style representation (of the i^{th} filter at j^{th} position) of the generated image in the l^{th} layer.

A^l_{ij}: style representation (of the i^{th} filter at j^{th} position) of the original image in the l^{th} layer.

- The loss from each individual layer will contribute to the total style loss as follows:

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

Trade-off Between Content & Style

- Adjusting the **α/β** ratio of the following loss equation will affect the results.

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

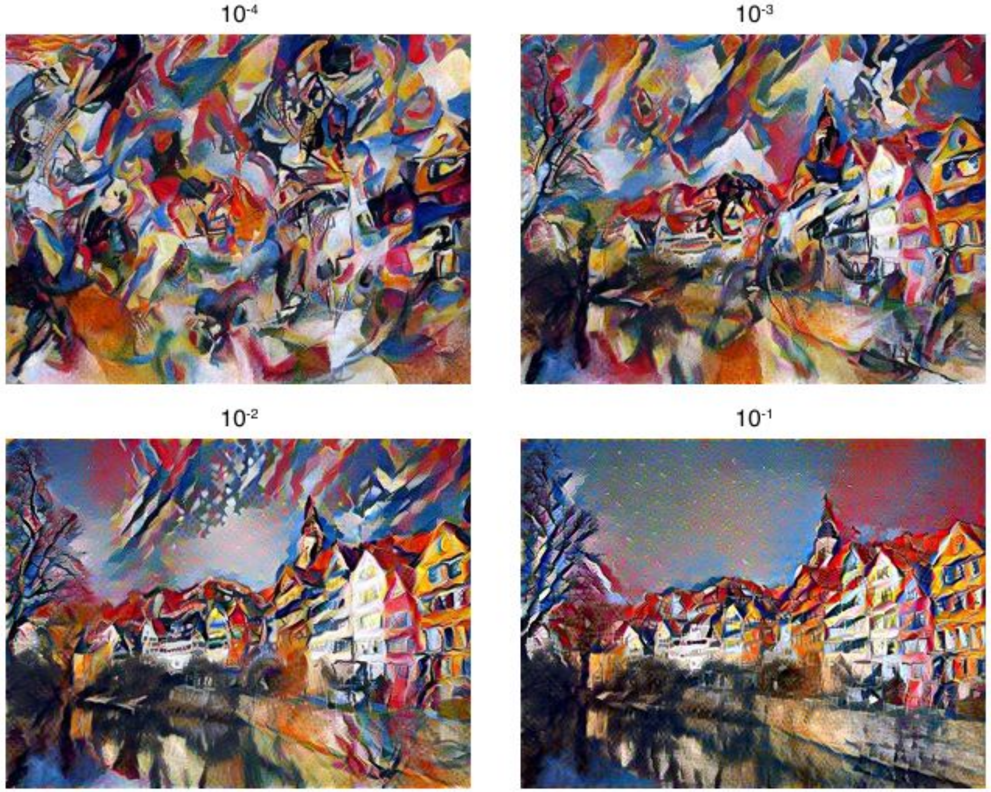


Figure 4. Relative weighting of matching content and style of the respective source images. The ratio α/β between matching the content and matching the style increases from top left to bottom right. A high emphasis on the style effectively produces a texturised version of the style image (top left). A high emphasis on the content produces an image with only little stylisation (bottom right). In practice one can smoothly interpolate between the two extremes.

Image Source: [Relative weighting of matching content and style]

Impact of the Initialization

- We can initialize the style transfer algorithm either on white noise or some deterministic content.

  - white noise → diverse outputs
  - content image → more stable, deterministic results

Figure 6. Initialisation of the gradient descent. **A** Initialised from the content image. **B** Initialised from the style image. **C** Four samples of images initialised from different white noise images. For all images the ratio α/β was equal to 1×10^{-3}

Image Source: [Initializations of the gradient descent]

Photorealistic Style Transfer

Figure 7. Photorealistic style transfer. The style is transferred from a photograph showing New York by night onto a picture showing London by day. The image synthesis was initialised from the content image and the ratio α/β was equal to 1×10^{-2}

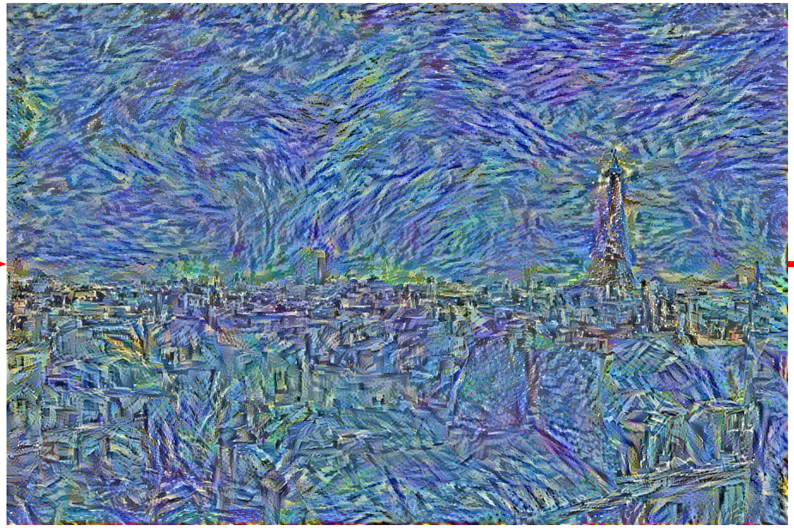
- Key challenge: It’s hard to maintain the realism while changing textures (applying styles).

Limitations

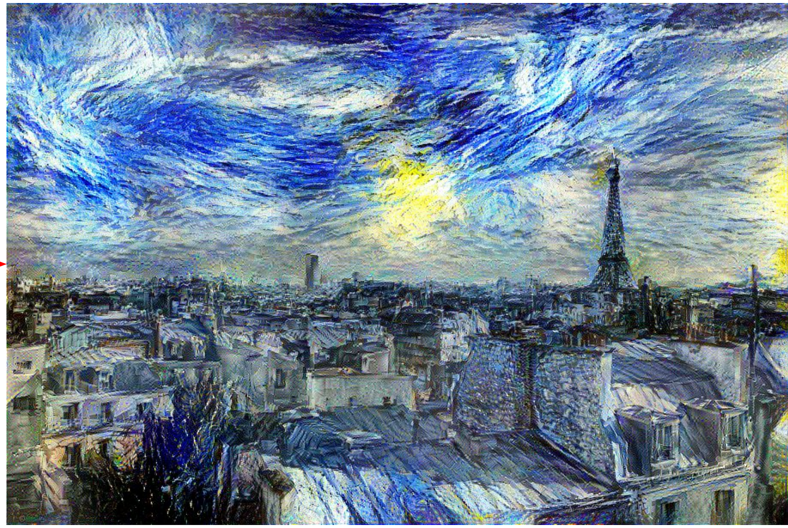
- When the style is too dominant, some content loss can occur.
- It's very difficult to selectively control how much style affects which parts.
- Limitations in computation:

  a. Optimization is iterating and higher iterations are needed for better visual quality.
  b. The speed of synthesis process heavily depends on the image resolution.

Original Image



Style Transferred (after 100 iterations)



Style Transferred (after 3000 iterations)

Let’s dive into the code!

Thank You