Annotated By: Sushant Gautam
Part of 30 Day GAN Paper Reading Challenge
https://github.com/sushant097/30-Days-GANs-Paper-Reading

Nerual Style Transfer

A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

[1]Werner Reichardt Centre for Integrative Neuroscience
and Institute of Theoretical Physics, University of Tübingen, Germany
[2]Bernstein Center for Computational Neuroscience, Tübingen, Germany
[3]Graduate School for Neural Information Processing, Tübingen, Germany
[4]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[5]Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA
*To whom correspondence should be addressed; E-mail: leon.gatys@bethgelab.org

In fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. However, in other key areas of visual perception such as object and face recognition near-human performance was recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks.[1,2] Here we introduce an artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality. The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images. Moreover, in light of the striking similarities between performance-optimised artificial neural networks and biological vision,[3–7] our work offers a path forward to an algorithmic understanding of how humans create and perceive artistic imagery.

Three types of Images:
1. Content Image: An image where we want to transfer style.
2. Style Image: An image where we want to transfer style.
3. Generated Image: An image that contains the final result(pixels i.e weights)

The class of Deep Neural Networks that are most powerful in image processing tasks are called Convolutional Neural Networks. Convolutional Neural Networks consist of layers of small computational units that process visual information hierarchically in a feed-forward manner (Fig 1). Each layer of units can be understood as a collection of image filters, each of which extracts a certain feature from the input image. Thus, the output of a given layer consists of so-called feature maps: differently filtered versions of the input image.

When Convolutional Neural Networks are trained on object recognition, they develop a representation of the image that makes object information increasingly explicit along the processing hierarchy.[8] Therefore, along the processing hierarchy of the network, the input image is transformed into representations that increasingly care about the actual *content* of the image compared to its detailed pixel values. We can directly visualise the information each layer contains about the input image by reconstructing the image only from the feature maps in that layer[9] (Fig 1, content reconstructions, see Methods for details on how to reconstruct the image). Higher layers in the network capture the high-level *content* in terms of objects and their arrangement in the input image but do not constrain the exact pixel values of the reconstruction. (Fig 1, content reconstructions d,e). In contrast, reconstructions from the lower layers simply reproduce the exact pixel values of the original image (Fig 1, content reconstructions a,b,c). We therefore refer to the feature responses in higher layers of the network as the *content representation*.

Higher Layers features -> Content Image representation

To obtain a representation of the *style* of an input image, we use a feature space originally designed to capture texture information.[8] This feature space is built on top of the filter responses in each layer of the network. It consists of the correlations between the different filter responses over the spatial extent of the feature maps (see Methods for details). By including the feature correlations of multiple layers, we obtain a stationary, multi-scale representation of the input image, which captures its texture information but not the global arrangement.

Each layer features map contains details of style image and with correlation of multiple layers, which captures texture information, and style from style image can be transfered to generated image.

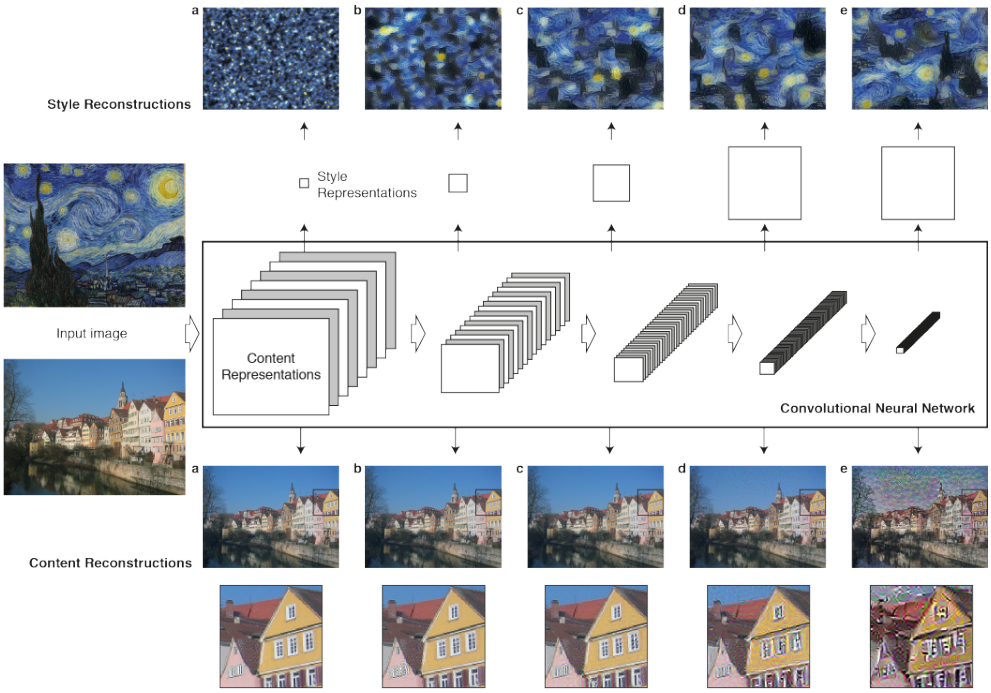


Figure 1: **Convolutional Neural Network (CNN)**. A given input image is represented as a set of filtered images at each processing stage in the CNN. While the number of different filters increases along the processing hierarchy, the size of the filtered images is reduced by some downsampling mechanism (e.g. max-pooling) leading to a decrease in the total number of units per layer of the network. **Content Reconstructions**. We can visualise the information at different processing stages in the CNN by reconstructing the input image from only knowing the network's responses in a particular layer. We reconstruct the input image from from layers 'conv1_1' (**a**), 'conv2_1' (**b**), 'conv3_1' (**c**), 'conv4_1' (**d**) and 'conv5_1' (**e**) of the original VGG-Network. We find that reconstruction from lower layers is almost perfect (**a**,**b**,**c**). In higher layers of the network, detailed pixel information is lost while the high-level content of the image is preserved (**d**,**e**). **Style Reconstructions**. On top of the original CNN representations we built a new feature space that captures the style of an input image. The style representation computes correlations between the different features in different layers of the CNN. We reconstruct the style of the input image from style representations built on different subsets of CNN layers ('conv1_1' (**a**), 'conv1_1' and 'conv2_1' (**b**), 'conv1_1', 'conv2_1' and 'conv3_1' (**c**), 'conv1_1', 'conv2_1', 'conv3_1' and 'conv4_1' (**d**), 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' (**e**)). This creates images that match the style of a given image on an increasing scale while discarding information of the global arrangement of the scene.

Again, we can visualise the information captured by these style feature spaces built on different layers of the network by constructing an image that matches the style representation of a given input image (Fig 1, style reconstructions).[10,11] Indeed reconstructions from the style features produce texturised versions of the input image that capture its general appearance in terms of colour and localised structures. Moreover, the size and complexity of local image structures from the input image increases along the hierarchy, a result that can be explained by the increasing receptive field sizes and feature complexity. We refer to this multi-scale representation as *style representation*.

The key finding of this paper is that the representations of content and style in the Convolutional Neural Network are separable. That is, we can manipulate both representations independently to produce new, perceptually meaningful images. To demonstrate this finding, we generate images that mix the content and style representation from two different source images. In particular, we match the content representation of a photograph depicting the "Neckarfront" in Tübingen, Germany and the style representations of several well-known artworks taken from different periods of art (Fig 2).

The images are synthesised by finding an image that simultaneously matches the content representation of the photograph and the style representation of the respective piece of art (see Methods for details). While the global arrangement of the original photograph is preserved, the colours and local structures that compose the global scenery are provided by the artwork. Effectively, this renders the photograph in the style of the artwork, such that the appearance of the synthesised image resembles the work of art, even though it shows the same content as the photograph.

As outlined above, the style representation is a multi-scale representation that includes multiple layers of the neural network. In the images we have shown in Fig 2, the style representation included layers from the whole network hierarchy. Style can also be defined more locally by

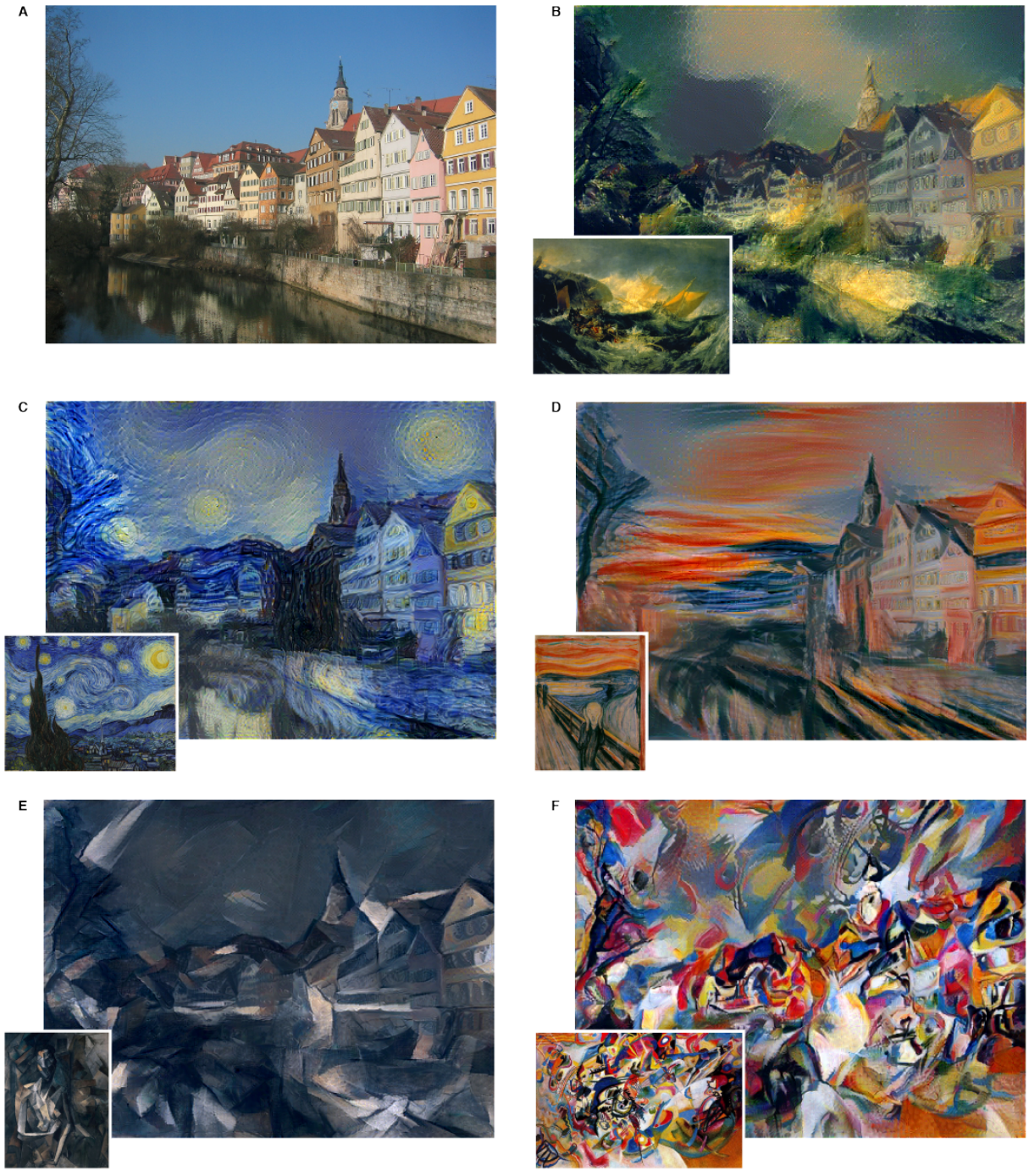


Figure 2: Images that combine the content of a photograph with the style of several well-known artworks. The images were created by finding an image that simultaneously matches the content representation of the photograph and the style representation of the artwork (see Methods). The original photograph 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.

including only a smaller number of lower layers, leading to different visual experiences (Fig 3, along the rows). When matching the style representations up to higher layers in the network, local images structures are matched on an increasingly large scale, leading to a smoother and more continuous visual experience. Thus, the visually most appealing images are usually created by matching the style representation up to the highest layers in the network (Fig 3, last row).

Of course, image content and style cannot be completely disentangled. When synthesising an image that combines the content of one image with the style of another, there usually does not exist an image that perfectly matches both constraints at the same time. However, the loss function we minimise during image synthesis contains two terms for content and style respectively, that are well separated (see Methods). We can therefore smoothly regulate the emphasis on either reconstructing the content or the style (Fig 3, along the columns). A strong emphasis on style will result in images that match the appearance of the artwork, effectively giving a texturised version of it, but hardly show any of the photograph's content (Fig 3, first column). When placing strong emphasis on content, one can clearly identify the photograph, but the style of the painting is not as well-matched (Fig 3, last column). For a specific pair of source images one can adjust the trade-off between content and style to create visually appealing images.

Here we present an artificial neural system that achieves a separation of image content from style, thus allowing to recast the content of one image in the style of any other image. We demonstrate this by creating new, artistic images that combine the style of several well-known paintings with the content of an arbitrarily chosen photograph. In particular, we derive the neural representations for the content and style of an image from the feature responses of high-performing Deep Neural Networks trained on object recognition. To our knowledge this is the first demonstration of image features separating content from style in whole natural images.

Figure 3: Detailed results for the style of the painting *Composition VII* by Wassily Kandinsky. The rows show the result of matching the style representation of increasing subsets of the CNN layers (see Methods). We find that the local image structures captured by the style representation increase in size and complexity when including style features from higher layers of the network. This can be explained by the increasing receptive field sizes and feature complexity along the network's processing hierarchy. The columns show different relative weightings between the content and style reconstruction. The number above each column indicates the ratio α/β between the emphasis on matching the content of the photograph and the style of the artwork (see Methods).

Previous work on separating content from style was evaluated on sensory inputs of much lesser complexity, such as characters in different handwriting or images of faces or small figures in different poses.[12,13]

In our demonstration, we render a given photograph in the style of a range of well-known artworks. This problem is usually approached in a branch of computer vision called non-photorealistic rendering (for recent review see[14]). Conceptually most closely related are methods using texture transfer to achieve artistic style transfer.[15–19] However, these previous approaches mainly rely on non-parametric techniques to directly manipulate the pixel representation of an image. In contrast, by using Deep Neural Networks trained on object recognition, we carry out manipulations in feature spaces that explicitly represent the high level content of an image.

Features from Deep Neural Networks trained on object recognition have been previously used for style recognition in order to classify artworks according to the period in which they were created.[20] There, classifiers are trained on top of the raw network activations, which we call content representations. We conjecture that a transformation into a stationary feature space such as our style representation might achieve even better performance in style classification.

In general, our method of synthesising images that mix content and style from different sources, provides a new, fascinating tool to study the perception and neural representation of art, style and content-independent image appearance in general. We can design novel stimuli that introduce two independent, perceptually meaningful sources of variation: the appearance and the content of an image. We envision that this will be useful for a wide range of experimental studies concerning visual perception ranging from psychophysics over functional imaging to even electrophysiological neural recordings. In fact, our work offers an algorithmic understanding of how neural representations can independently capture the content of an image and the style in which it is presented. Importantly, the mathematical form of our style representa-

tions generates a clear, testable hypothesis about the representation of image appearance down to the single neuron level. The style representations simply compute the correlations between different types of neurons in the network. Extracting correlations between neurons is a biologically plausible computation that is, for example, implemented by so-called complex cells in the primary visual system (V1).[21] Our results suggest that performing a complex-cell like computation at different processing stages along the ventral stream would be a possible way to obtain a content-independent representation of the appearance of a visual input.

All in all it is truly fascinating that a neural system, which is trained to perform one of the core computational tasks of biological vision, automatically learns image representations that allow the separation of image content from style. The explanation could be that when learning object recognition, the network has to become invariant to all image variation that preserves object identity. Representations that factorise the variation in the content of an image and the variation in its appearance would be extremely practical for this task. Thus, our ability to abstract content from style and therefore our ability to create and enjoy art might be primarily a preeminent signature of the powerful inference capabilities of our visual system.

Methods

The results presented in the main text were generated on the basis of the VGG-Network,[22] a Convolutional Neural Network that rivals human performance on a common visual object recognition benchmark task[23] and was introduced and extensively described in.[22] We used the feature space provided by the 16 convolutional and 5 pooling layers of the 19 layer VGG-Network. We do not use any of the fully connected layers.The model is publicly available and can be explored in the caffe-framework.[24] For image synthesis we found that replacing the max-pooling operation by average pooling improves the gradient flow and one obtains slightly more appealing results, which is why the images shown were generated with average pooling.

Generally each layer in the network defines a non-linear filter bank whose complexity increases with the position of the layer in the network. Hence a given input image $\vec{x}$ is encoded in each layer of the CNN by the filter responses to that image. A layer with N_l distinct filters has N_l feature maps each of size M_l, where M_l is the height times the width of the feature map. So the responses in a layer l can be stored in a matrix $F^l \in \mathcal{R}^{N_l \times M_l}$ where F^l_{ij} is the activation of the i^{th} filter at position j in layer l. To visualise the image information that is encoded at different layers of the hierarchy (Fig 1, content reconstructions) we perform gradient descent on a white noise image to find another image that matches the feature responses of the original image. So let $\vec{p}$ and $\vec{x}$ be the original image and the image that is generated and P^l and F^l their respective feature representation in layer l. We then define the squared-error loss between the two feature representations

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

The derivative of this loss with respect to the activations in layer l equals

$$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 \, . \end{cases} \tag{2}$$

from which the gradient with respect to the image $\vec{x}$ can be computed using standard error back-propagation. Thus we can change the initially random image $\vec{x}$ until it generates the same response in a certain layer of the CNN as the original image $\vec{p}$. The five content reconstructions in Fig 1 are from layers 'conv1_1' (a), 'conv2_1' (b), 'conv3_1' (c), 'conv4_1' (d) and 'conv5_1' (e) of the original VGG-Network.

On top of the CNN responses in each layer of the network we built a style representation that computes the correlations between the different filter responses, where the expectation is taken over the spatial extend of the input image. These feature correlations are given by the Gram matrix $G^l \in \mathcal{R}^{N_l \times N_l}$, where G^l_{ij} is the inner product between the vectorised feature map

Gram Matrix: The gram matrix is used to capture the "distribution of features" of a set of feature maps called the style matrix.The gram matrix is a correlation operation i.e. dot product of feature maps at a layer that summarizes the activations that co-occur. As texture(style) has strong locality and when we capture activations that co-occur a lot — we capture locality.

By finding this matrix, we get encoded correlated activations closer to the target(the style we want to capture) retrieves the style. And, Gram matrix is position invariant — it's based on statistics at individual points in feature maps.

i and j in layer l:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l. \tag{3}$$

To generate a texture that matches the style of a given image (Fig 1, style reconstructions), we use gradient descent from a white noise image to find another image that matches the style representation of the original image. This is done by minimising the mean-squared distance between the entries of the Gram matrix from the original image and the Gram matrix of the image to be generated. So let $\vec{a}$ and $\vec{x}$ be the original image and the image that is generated and A^l and G^l their respective style representations in layer l. The contribution of that layer to the total loss is then

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

and the total loss is

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{5}$$

where w_l are weighting factors of the contribution of each layer to the total loss (see below for specific values of w_l in our results). The derivative of E_l with respect to the activations in layer l can be computed analytically:

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left((F^l)^{\mathrm{T}} \left(G^l - A^l\right)\right)_{ji} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0\,. \end{cases} \tag{6}$$

The gradients of E_l with respect to the activations in lower layers of the network can be readily computed using standard error back-propagation. The five style reconstructions in Fig 1 were generated by matching the style representations on layer 'conv1_1' (a), 'conv1_1' and 'conv2_1' (b), 'conv1_1', 'conv2_1' and 'conv3_1' (c), 'conv1_1', 'conv2_1', 'conv3_1' and 'conv4_1' (d), 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' (e).

To generate the images that mix the content of a photograph with the style of a painting (Fig 2) we jointly minimise the distance of a white noise image from the content representation

of the photograph in one layer of the network and the style representation of the painting in a number of layers of the CNN. So let $\vec{p}$ be the photograph and $\vec{a}$ be the artwork. The loss function we minimise is

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x}) \tag{7}$$

where α and β are the weighting factors for content and style reconstruction respectively. For the images shown in Fig 2 we matched the content representation on layer 'conv4_2' and the style representations on layers 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' ($w_l = 1/5$ in those layers, $w_l = 0$ in all other layers) . The ratio α/β was either 1×10^{-3} (Fig 2 B,C,D) or 1×10^{-4} (Fig 2 E,F). Fig 3 shows results for different relative weightings of the content and style reconstruction loss (along the columns) and for matching the style representations only on layer 'conv1_1' (A), 'conv1_1' and 'conv2_1' (B), 'conv1_1', 'conv2_1' and 'conv3_1' (C), 'conv1_1', 'conv2_1', 'conv3_1' and 'conv4_1' (D), 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' (E). The factor w_l was always equal to one divided by the number of active layers with a non-zero loss-weight w_l.

Acknowledgments This work was funded by the German National Academic Foundation (L.A.G.), the Bernstein Center for Computational Neuroscience (FKZ 01GQ1002) and the German Excellency Initiative through the Centre for Integrative Neuroscience Tübingen (EXC307)(M.B., A.S.E, L.A.G.)

References and Notes

1. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105 (2012). URL `http://papers.nips.cc/paper/4824-imagenet`.

2. Taigman, Y., Yang, M., Ranzato, M. & Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 1701–1708 (IEEE, 2014). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6909616.

3. Güçlü, U. & Gerven, M. A. J. v. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. *The Journal of Neuroscience* **35**, 10005–10014 (2015). URL http://www.jneurosci.org/content/35/27/10005.

4. Yamins, D. L. K. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* 201403112 (2014). URL http://www.pnas.org/content/early/2014/05/08/1403112111.

5. Cadieu, C. F. *et al.* Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Comput Biol* **10**, e1003963 (2014). URL http://dx.doi.org/10.1371/journal.pcbi.1003963.

6. Kümmerer, M., Theis, L. & Bethge, M. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet. In *ICLR Workshop* (2015). URL /media/publications/1411.1045v4.pdf.

7. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Comput Biol* **10**, e1003915 (2014). URL http://dx.doi.org/10.1371/journal.pcbi.1003915.

8. Gatys, L. A., Ecker, A. S. & Bethge, M. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *arXiv:1505.07376 [cs, q-bio]* (2015). URL http://arxiv.org/abs/1505.07376. ArXiv: 1505.07376.

9. Mahendran, A. & Vedaldi, A. Understanding Deep Image Representations by Inverting Them. *arXiv:1412.0035 [cs]* (2014). URL http://arxiv.org/abs/1412.0035. ArXiv: 1412.0035.

10. Heeger, D. J. & Bergen, J. R. Pyramid-based Texture Analysis/Synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, 229–238 (ACM, New York, NY, USA, 1995). URL http://doi.acm.org/10.1145/218380.218446.

11. Portilla, J. & Simoncelli, E. P. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision* **40**, 49–70 (2000). URL http://link.springer.com/article/10.1023/A%3A1026553619983.

12. Tenenbaum, J. B. & Freeman, W. T. Separating style and content with bilinear models. *Neural computation* **12**, 1247–1283 (2000). URL http://www.mitpressjournals.org/doi/abs/10.1162/089976600300015349.

13. Elgammal, A. & Lee, C.-S. Separating style and content on a nonlinear manifold. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, I–478 (IEEE, 2004). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315070.

14. Kyprianidis, J. E., Collomosse, J., Wang, T. & Isenberg, T. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *Visualization and Computer*

Graphics, IEEE Transactions on **19**, 866–885 (2013). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6243138.

15. Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. & Salesin, D. H. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 327–340 (ACM, 2001). URL http://dl.acm.org/citation.cfm?id=383295.

16. Ashikhmin, N. Fast texture transfer. *IEEE Computer Graphics and Applications* **23**, 38–43 (2003).

17. Efros, A. A. & Freeman, W. T. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 341–346 (ACM, 2001). URL http://dl.acm.org/citation.cfm?id=383296.

18. Lee, H., Seo, S., Ryoo, S. & Yoon, K. Directional Texture Transfer. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '10, 43–48 (ACM, New York, NY, USA, 2010). URL http://doi.acm.org/10.1145/1809939.1809945.

19. Xie, X., Tian, F. & Seah, H. S. Feature Guided Texture Synthesis (FGTS) for Artistic Style Transfer. In *Proceedings of the 2Nd International Conference on Digital Interactive Media in Entertainment and Arts*, DIMEA '07, 44–49 (ACM, New York, NY, USA, 2007). URL http://doi.acm.org/10.1145/1306813.1306830.

20. Karayev, S. *et al.* Recognizing image style. *arXiv preprint arXiv:1311.3715* (2013). URL http://arxiv.org/abs/1311.3715.

21. Adelson, E. H. & Bergen, J. R. Spatiotemporal energy models for the perception of motion. *JOSA A* **2**, 284–299 (1985). URL http://www.opticsinfobase.org/josaa/fulltext.cfm?uri=josaa-2-2-284.

22. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* (2014). URL http://arxiv.org/abs/1409.1556. ArXiv: 1409.1556.

23. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]* (2014). URL http://arxiv.org/abs/1409.0575. ArXiv: 1409.0575.

24. Jia, Y. *et al.* Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, 675–678 (ACM, 2014). URL http://dl.acm.org/citation.cfm?id=2654889.

Neural Style Transfer(Summary)

- The concept of style transfer is to transfer a style or texture to an input image, and the method to achieve a style-transferred image with a deep neural network called Neural Style Transfer (NST). It means NST is a technique that takes two images — a content image and a style reference image (that can be an artwork by a famous painter or new texture) — and mix them together so the output look likes the content image, but "painted" in the style of a reference image.

- In NST we update the pixels of an image (Weights to be updated of a result image) and the frozen weights of the Pre-Trained Network like VGG-19. We keep updating the final image pixels according to loss until it is desired output i.e minimum loss.

- Caputring Content Image: CNN pretrained object detection captures high-level semantic information in shallow layers as welll as low-level pixel information in deep layers. Later convolutional layers learn features such as more complex textures and patterns and used to represent image content.

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

In which $\vec{p}$ and $\vec{x}$ are the original image and the generated image. F and P are their feature maps from the pre-trained VGG net in layer l.

- Capturing Style Image: Representing style of image should be position invariant, so one might use a statistics over the whole feature map to represent the style of an image. Its main objective is to enforce the details of style images in the generated image. For that, there should be a similar correlation of activations between the style image and the generated image i.e. measured by correlation matrix called Gram Matrix, and defined as:

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

Given a $C \times H \times W$ feature map. Its corresponding Gram Matrix is a $C \times C$ matrix, with C_{ij} computed as the element-wise product and summation across feature map in channel i and channel j. So it can be interpreted as the correlation between feature map F_i and F_j.

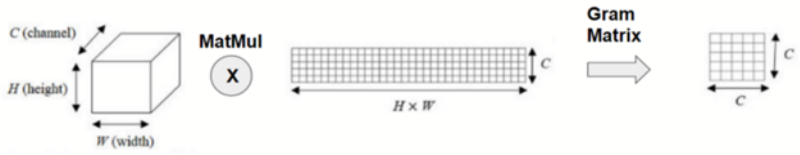


Fig: Concept of Gram Matrix

- We computed the style loss as a weighted sum of the mean square errors between the Gram Matrices computed in layer l1,l2,l3...ln.

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

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{l} w_l E_l$$

- **Total Loss:**

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

Where alpha and beta are weights for content and style, respectively. They can be tweaked to alter our final result.

✔ Training STEPS

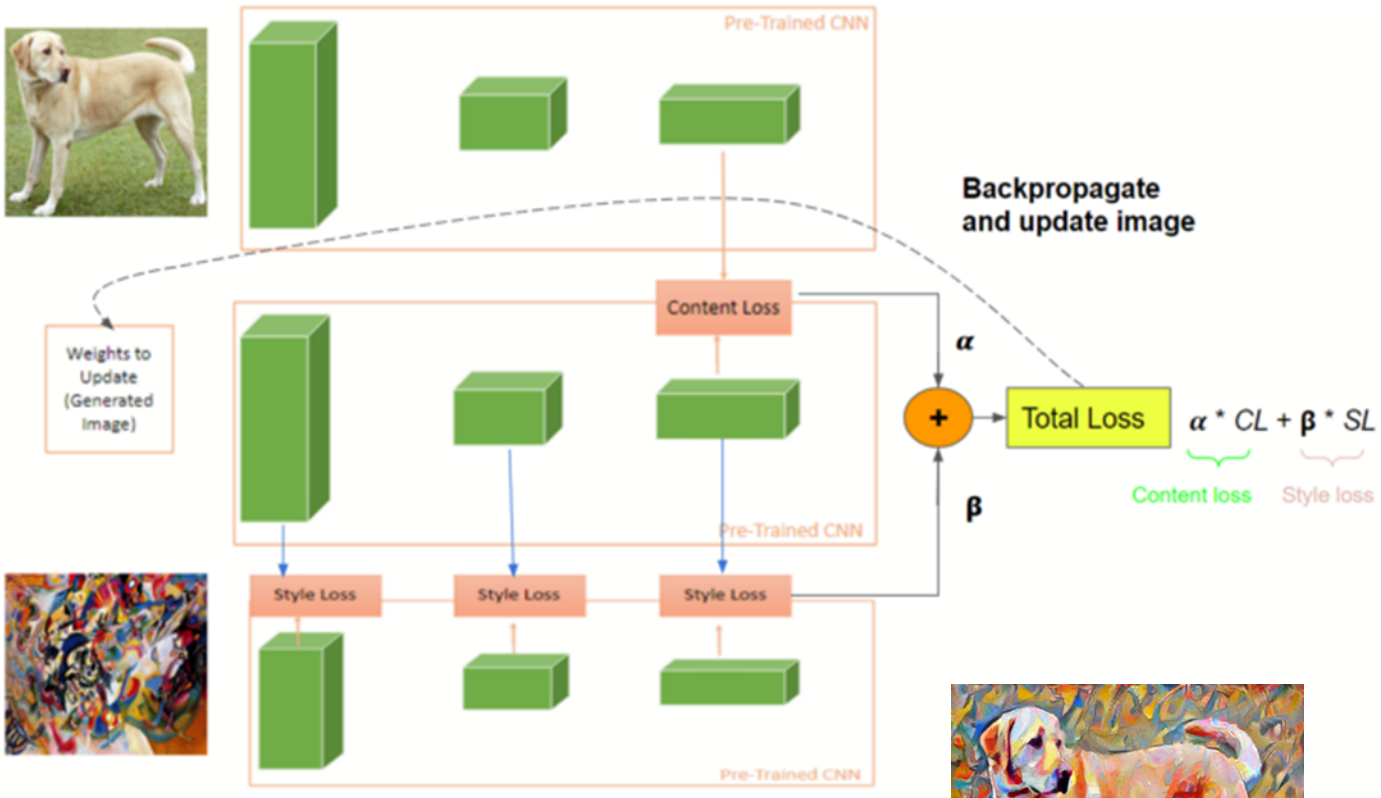
1. Compute features using pre-trained models like VGG, ResNet for each content, style, and generated image.
2. Compute content loss and style loss.
3. Compute total combined loss.
4. Backpropagate Gradient to update generated image weights pixels, while pre-trained models weights were frozen.

− Limitation

Drawback of this algorithm is inefficient and slow and it needs many optimization iterations to generate a transferred image

Other Paper like Feed-Forward Style Transfer by Johnson et al. overcome this issue.

+ One Image Summary:



Applications of Neural Style Transfer:

Photo and video editors

Commercial art

Gaming

Virtual Reality

Fig: Final Generated Image with style transferred on Content Image.