Image Style Transfer Using Convolutional Neural Networks

Leon A. Gatys
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Graduate School of Neural Information Processing, University of Tübingen, Germany
leon.gatys@bethgelab.org

Alexander S. Ecker
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Baylor College of Medicine, Houston, TX, USA

Matthias Bethge
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract

Rendering the semantic content of an image in different styles is a difficult image processing task. Arguably, a major limiting factor for previous approaches has been the lack of image representations that explicitly represent semantic information and, thus, allow to separate image content from style. Here we use image representations derived from Convolutional Neural Networks optimised for object recognition, which make high level image information explicit. We introduce A Neural Algorithm of Artistic Style *that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of numerous well-known artworks. Our results provide new insights into the deep image representations learned by Convolutional Neural Networks and demonstrate their potential for high level image synthesis and manipulation.*

1. Introduction

Transferring the style from one image onto another can be considered a problem of texture transfer. In texture transfer the goal is to synthesise a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. For texture synthesis there exist a large range of powerful non-parametric algorithms that can synthesise photorealistic natural textures by resampling the pixels of a given source texture [7, 30, 8, 20]. Most previous texture transfer algorithms rely on these non-parametric methods for texture synthesis while using different ways to preserve the structure of the target image. For instance, Efros and Freeman introduce a correspondence map that includes features of the target image such as image intensity to constrain the texture synthesis procedure [8]. Hertzman et al. use image analogies to transfer the texture from an already stylised image onto a target image[13]. Ashikhmin focuses on transferring the high-frequency texture information while preserving the coarse scale of the target image [1]. Lee et al. improve this algorithm by additionally informing the texture transfer with edge orientation information [22].

Although these algorithms achieve remarkable results, they all suffer from the same fundamental limitation: they use only low-level image features of the target image to inform the texture transfer. Ideally, however, a style transfer algorithm should be able to extract the semantic image content from the target image (e.g. the objects and the general scenery) and then inform a texture transfer procedure to render the semantic content of the target image in the style of the source image. Therefore, a fundamental prerequisite is to find image representations that independently model variations in the semantic image content and the style in which

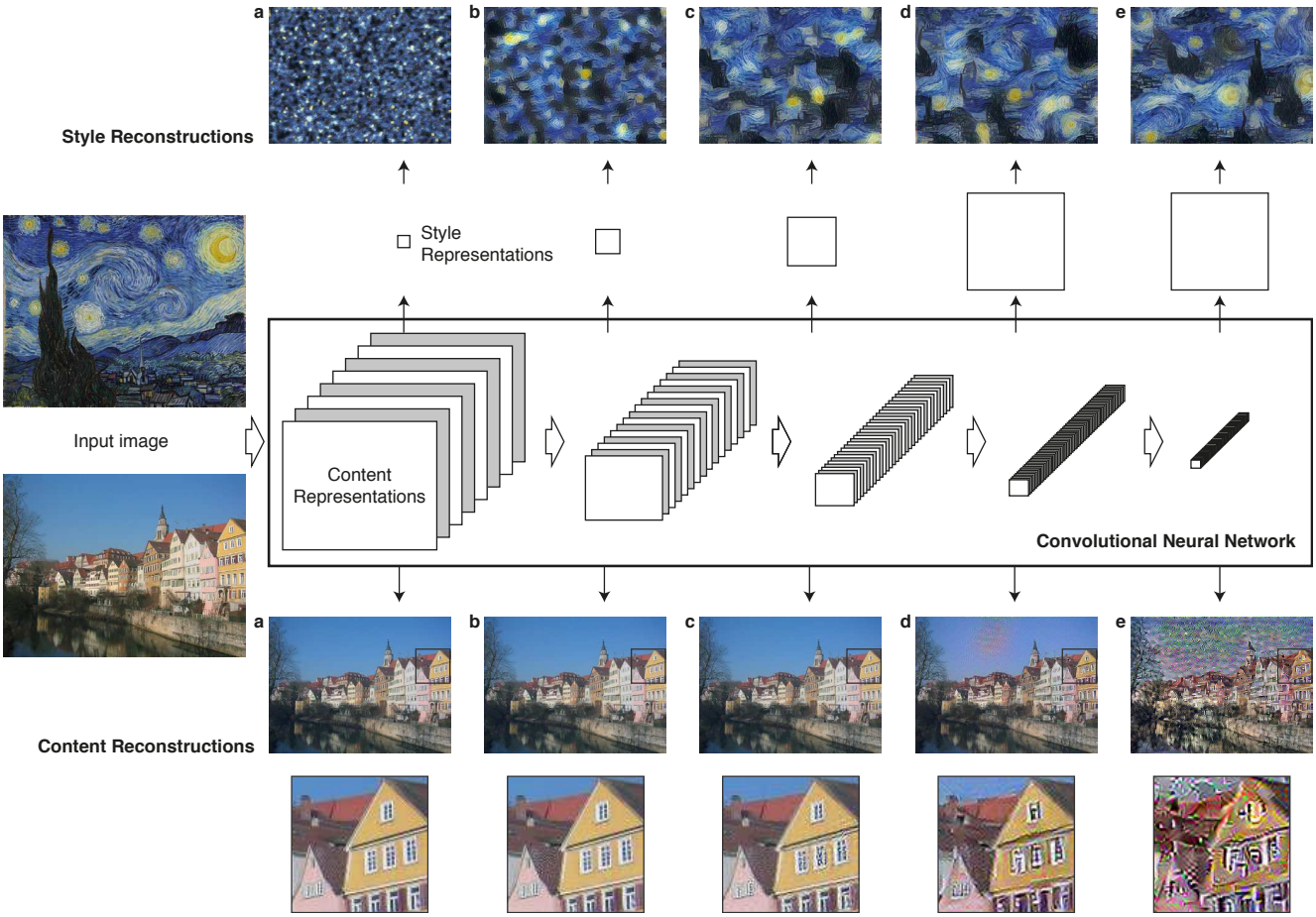

Figure 1. Image representations in a Convolutional Neural Network (CNN). A given input image is represented as a set of filtered images at each processing stage in the CNN. While the number of different filters increases along the processing hierarchy, the size of the filtered images is reduced by some 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_2' (a), 'conv2_2' (b), 'conv3_2' (c), 'conv4_2' (d) and 'conv5_2' (e) of the original VGG-Network. We find that reconstruction from lower layers is almost perfect (a–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 activations we use a feature space that captures the texture information 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 a style representation 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.

it is presented. Such factorised representations were previously achieved only for controlled subsets of natural images such as faces under different illumination conditions and characters in different font styles [29] or handwritten digits and house numbers [17].

To generally separate content from style in natural images is still an extremely difficult problem. However, the recent advance of Deep Convolutional Neural Networks [18] has produced powerful computer vision systems that learn to extract high-level semantic information from natural images. It was shown that Convolutional Neural Networks trained with sufficient labeled data on specific tasks such as object recognition learn to extract high-level image content in generic feature representations that generalise across datasets [6] and even to other visual information processing tasks [19, 4, 2, 9, 23], including texture recognition [5] and artistic style classification [15].

In this work we show how the generic feature representations learned by high-performing Convolutional Neural Networks can be used to independently process and manipulate the content and the style of natural images. We introduce *A Neural Algorithm of Artistic Style*, a new algo-

rithm to perform image style transfer. Conceptually, it is a texture transfer algorithm that constrains a texture synthesis method by feature representations from state-of-the-art Convolutional Neural Networks. Since the texture model is also based on deep image representations, the style transfer method elegantly reduces to an optimisation problem within a single neural network. New images are generated by performing a pre-image search to match feature representations of example images. This general approach has been used before in the context of texture synthesis [12, 25, 10] and to improve the understanding of deep image representations [27, 24]. In fact, our style transfer algorithm combines a parametric texture model based on Convolutional Neural Networks [10] with a method to invert their image representations [24].

2. Deep image representations

The results presented below were generated on the basis of the VGG network [28], which was trained to perform object recognition and localisation [26] and is described extensively in the original work [28]. We used the feature space provided by a normalised version of the 16 convolutional and 5 pooling layers of the 19-layer VGG network. We normalized the network by scaling the weights such that the mean activation of each convolutional filter over images and positions is equal to one. Such re-scaling can be done for the VGG network without changing its output, because it contains only rectifying linear activation functions and no normalization or pooling over feature maps. We do not use any of the fully connected layers. The model is publicly available and can be explored in the caffe-framework [14]. For image synthesis we found that replacing the maximum pooling operation by average pooling yields slightly more appealing results, which is why the images shown were generated with average pooling.

2.1. Content representation

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 Convolutional Neural Network 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 one can perform gradient descent on a white noise image to find another image that matches the feature responses of the original image (Fig 1, content reconstructions) [24]. 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

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

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

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

from which the gradient with respect to the image $\vec{x}$ can be computed using standard error back-propagation (Fig 2, right). Thus we can change the initially random image $\vec{x}$ until it generates the same response in a certain layer of the Convolutional Neural Network as the original image $\vec{p}$.

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 [10]. Therefore, along the processing hierarchy of the network, the input image is transformed into representations that are increasingly sensitive to the actual *content* of the image, but become relatively invariant to its precise appearance. Thus, 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 very much (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–c). We therefore refer to the feature responses in higher layers of the network as the *content representation*.

2.2. Style representation

To obtain a representation of the *style* of an input image, we use a feature space designed to capture texture information [10]. This feature space can be built on top of the filter responses in any layer of the network. It consists of the correlations between the different filter responses, where the expectation is taken over the spatial extent of the feature maps. 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 maps i and j in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}. \qquad (3)$$

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. Again, we can visualise the information captured by these style feature spaces built on different

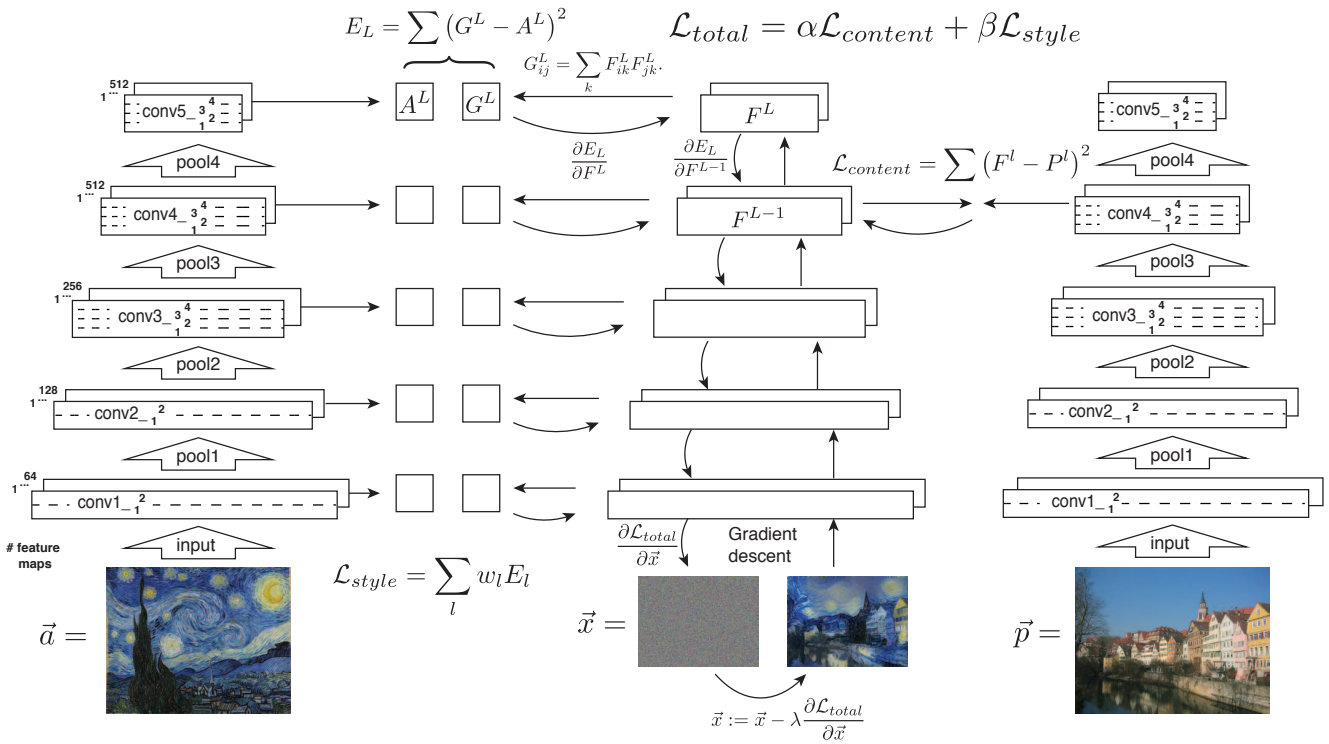

Figure 2. Style transfer algorithm. First content and style features are extracted and stored. The style image $\vec{a}$ is passed through the network and its style representation A^l on all layers included are computed and stored (left). The content image $\vec{p}$ is passed through the network and the content representation P^l in one layer is stored (right). Then a random white noise image $\vec{x}$ is passed through the network and its style features G^l and content features F^l are computed. On each layer included in the style representation, the element-wise mean squared difference between G^l and A^l is computed to give the style loss $\mathcal{L}_{style}$ (left). Also the mean squared difference between F^l and P^l is computed to give the content loss $\mathcal{L}_{content}$ (right). The total loss $\mathcal{L}_{total}$ is then a linear combination between the content and the style loss. Its derivative with respect to the pixel values can be computed using error back-propagation (middle). This gradient is used to iteratively update the image $\vec{x}$ until it simultaneously matches the style features of the style image $\vec{a}$ and the content features of the content image $\vec{p}$ (middle, bottom).

layers of the network by constructing an image that matches the style representation of a given input image (Fig 1, style reconstructions). This is done by using gradient descent from a white noise image to minimise the mean-squared distance between the entries of the Gram matrices from the original image and the Gram matrices of the image to be generated [10, 25].

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 representation in layer l. The contribution of layer l to the total loss is then

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

and the total style loss is

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

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:

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

The gradients of E_l with respect to the pixel values $\vec{x}$ can be readily computed using standard error back-propagation (Fig 2, left).

2.3. Style transfer

To transfer the style of an artwork $\vec{a}$ onto a photograph $\vec{p}$ we synthesise a new image that simultaneously matches the content representation of $\vec{p}$ and the style representation of $\vec{a}$ (Fig 2). Thus we jointly minimise the distance of the feature representations of a white noise image from the content representation of the photograph in one layer and the style representation of the painting defined on a number of layers of the Convolutional Neural Network. The loss function we minimise is

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.

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

where α and β are the weighting factors for content and style reconstruction, respectively. The gradient with respect to the pixel values $\frac{\partial \mathcal{L}_{\text{total}}}{\partial \vec{x}}$ can be used as input for some numerical optimisation strategy. Here we use L-BFGS [32], which we found to work best for image synthesis. To extract image information on comparable scales, we always resized the style image to the same size as the content image before computing its feature representations. Finally, note that in difference to [24] we do not regularise our synthesis results with image priors. It could be argued, though, that the texture features from lower layers in the network act as a specific image prior for the style image. Additionally some differences in the image synthesis are expected due to the different network architecture and optimisation algorithm we use.

3. Results

The key finding of this paper is that the representations of content and style in the Convolutional Neural Network are well 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 riverfront of the Neckar river in Tübingen, Germany and the style representations of several well-known artworks taken from different periods of art (Fig 3). The images shown in Fig 3 were synthesised by matching the content representation on layer 'conv4_2' and the style representation 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 3 B), 8×10^{-4} (Fig 3 C), 5×10^{-3} (Fig 3 D), or 5×10^{-4} (Fig 3 E, F).

3.1. Trade-off between content and style matching

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, since the loss function we minimise during image synthesis is a linear combination between the loss functions for content and style respectively, we can smoothly regulate the emphasis on either reconstructing the content or the style (Fig 4). A strong emphasis on style will result in images that match the appearance of the artwork, effectively giving a texturised version of it, but show hardly any of the photograph's content ($\alpha/\beta = 1 \times 10^{-4}$, Fig 4, top left). When placing strong emphasis on content, one can clearly identify the photograph, but the style of the painting is not as well-matched ($\alpha/\beta = 1 \times 10^{-1}$, Fig 4, bottom right). For a specific pair of content and style images one can adjust the trade-off between content and style to create visually appealing images.

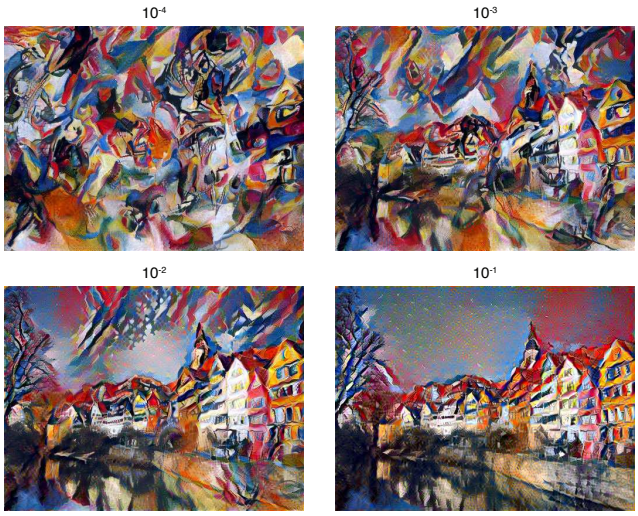

Figure 4. Relative weighting of matching content and style of the respective source images. The ratio α/β between matching the content and matching the style increases from top left to bottom right. A high emphasis on the style effectively produces a texturised version of the style image (top left). 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.

3.2. Effect of different layers of the Convolutional Neural Network

Another important factor in the image synthesis process is the choice of layers to match the content and style representation on. As outlined above, the style representation is a multi-scale representation that includes multiple layers of the neural network. The number and position of these layers determines the local scale on which the style is matched, leading to different visual experiences (Fig 1, style reconstructions). We find that matching the style representations up to higher layers in the network preserves local images structures 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 high layers in the network, which is why for all images shown we match the style features in layers 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' of the network.

To analyse the effect of using different layers to match the content features, we present a style transfer result obtained by stylising a photograph with the same artwork and parameter configuration ($\alpha/\beta = 1 \times 10^{-3}$), but in one

Figure 5. The effect of matching the content representation in different layers of the network. Matching the content on layer 'conv2_2' preserves much of the fine structure of the original photograph and the synthesised image looks as if the texture of the painting is simply blended over the photograph (middle). When matching the content on layer 'conv4_2' the texture of the painting and the content of the photograph merge together such that the content of the photograph is displayed in the style of the painting (bottom). Both images were generated with the same choice of parameters ($\alpha/\beta = 1 \times 10^{-3}$). The painting that served as the style image is shown in the bottom left corner and is named *Jesuiten III* by Lyonel Feininger, 1915.

matching the content features on layer 'conv2_2' and in the other on layer 'conv4_2' (Fig 5). When matching the content on a lower layer of the network, the algorithm matches much of the detailed pixel information in the photograph and the generated image appears as if the texture of the artwork is merely blended over the photograph (Fig 5, middle). In contrast, when matching the content features on a higher layer of the network, detailed pixel information of the photograph is not as strongly constraint and the texture of the artwork and the content of the photograph are properly merged. That is, the fine structure of the image, for example the edges and colour map, is altered such that it agrees with the style of the artwork while displaying the content of the photograph (Fig 5, bottom).

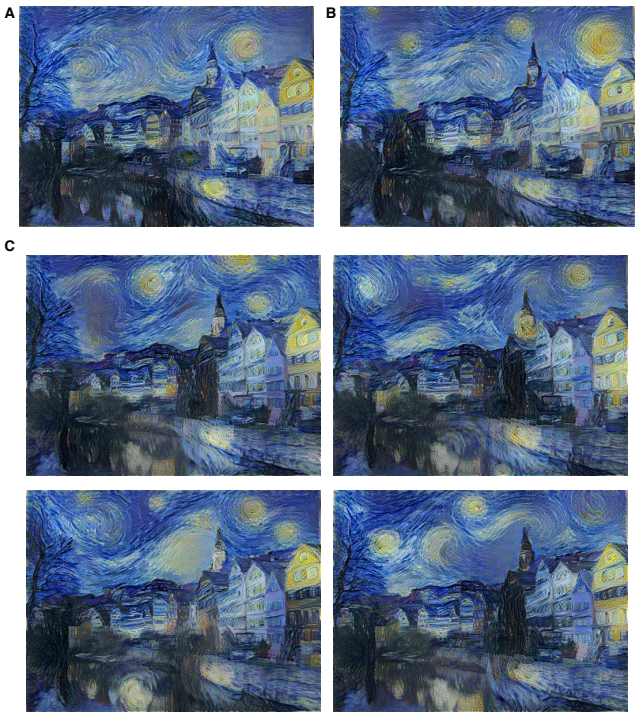

Figure 6. Initialisation of the gradient descent. **A** Initialised from the content image. **B** Initialised from the style image. **C** Four samples of images initialised from different white noise images. For all images the ratio α/β was equal to 1×10^{-3}

3.3. Initialisation of gradient descent

We have initialised all images shown so far with white noise. However, one could also initialise the image synthesis with either the content image or the style image. We explored these two alternatives (Fig 6 A, B): although they bias the final image somewhat towards the spatial structure of the initialisation, the different initialisations do not seem to have a strong effect on the outcome of the synthesis procedure. It should be noted that only initialising with noise allows to generate an arbitrary number of new images (Fig 6 C). Initialising with a fixed image always deterministically leads to the same outcome (up to stochasticity in the gradient descent procedure).

3.4. Photorealistic style transfer

Thus far the focus of this paper was on artistic style transfer. In general though, the algorithm can transfer the style between arbitrary images. As an example we transfer the style of a photograph of New York by night onto an image of London in daytime (Fig 7). Although photorealism is not fully preserved, the synthesised image resembles much of the colours and lightning of the style image and to some extent displays an image of London by night.

4. Discussion

Here we demonstrate how to use feature representations from high-performing Convolutional Neural Networks to transfer image style between arbitrary images. While we are able to show results of high perceptual quality, there are still some technical limitations to the algorithm.

Probably the most limiting factor is the resolution of the synthesised images. Both, the dimensionality of the optimisation problem as well as the number of units in the Convolutional Neural Network grow linearly with the number of pixels. Therefore the speed of the synthesis procedure depends heavily on image resolution. The images presented in this paper were synthesised in a resolution of about 512×512 pixels and the synthesis procedure could take up to an hour on a Nvidia K40 GPU (depending on the exact image size and the stopping criteria for the gradient descent). While this performance currently prohibits online and interactive applications of our style transfer algorithm, it is likely that future improvements in deep learning will also increase the performance of this method.

Another issue is that synthesised images are sometimes subject to some low-level noise. While this is less of an issue in the artistic style transfer, the problem becomes more apparent when both, content and style images, are photographs and the photorealism of the synthesised image is affected. However, the noise is very characteristic and appears to resemble the filters of units in the network. Thus it could be possible to construct efficient de-noising techniques to post-process the images after the optimisation procedure.

Artistic stylisation of images is traditionally studied in computer graphics under the label of non-photorealistic rendering. Apart from the work on texture transfer, common approaches are conceptually quite different to our work in that they give specialised algorithms to render a source image in one specific style. For a recent review of the field we refer the reader to [21].

The separation of image content from style is not necessarily a well defined problem. This is mostly because it is not clear what exactly defines the style of an image. It might be the brush strokes in a painting, the colour map, certain dominant forms and shapes, but also the composition of a scene and the choice of the subject of the image – and probably it is a mixture of all of them and many more. Therefore, it is generally not clear if image content and style can be completely separated at all – and if so, how. For example it is not possible to render an image in the style of van Gogh's "Starry Night" without having image structures that resemble the stars. In our work we consider style transfer to be successful if the generated image 'looks like' the style image but shows the objects and scenery of the content image. We are fully aware though that this evaluation criterion is neither mathematically precise nor universally agreed upon.

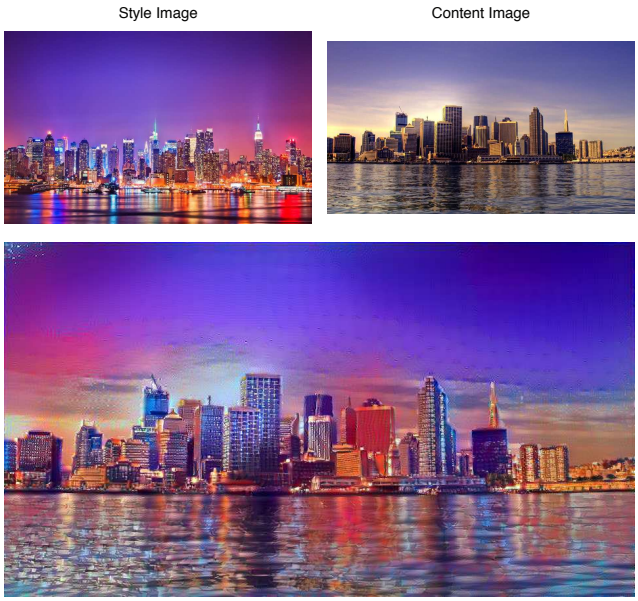

Figure 7. Photorealistic style transfer. The style is transferred from a photograph showing New York by night onto a picture showing London by day. The image synthesis was initialised from the content image and the ratio α/β was equal to 1×10^{-2}

Nevertheless, we find it 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 – at least to some extent – the separation of image content from style. One explanation may 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. In light of the striking similarities between performance-optimised artificial neural networks and biological vision [11, 31, 3, 19, 16], it is thus tempting to speculate that the human ability to abstract content from style – and therefore our ability to create and enjoy art – might also be primarily a preeminent signature of the powerful inference capabilities of our visual system.

References

[1] N. Ashikhmin. Fast texture transfer. *IEEE Computer Graphics and Applications*, 23(4):38–43, July 2003. 1

[2] M. Berning, K. M. Boergens, and M. Helmstaedter. SegEM: Efficient Image Analysis for High-Resolution Connectomics. *Neuron*, 87(6):1193–1206, Sept. 2015. 2

[3] C. F. Cadieu, H. Hong, D. L. K. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo. Deep Neural Networks Rival the Representation of Primate IT Cor-

tex for Core Visual Object Recognition. *PLoS Comput Biol*, 10(12):e1003963, Dec. 2014. 8

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv:1412.7062 [cs]*, Dec. 2014. arXiv: 1412.7062. 2

[5] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015. 2

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv:1310.1531 [cs]*, Oct. 2013. arXiv: 1310.1531. 2

[7] A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999. 1

[8] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 1

[9] D. Eigen and R. Fergus. Predicting Depth, Surface Normals and Semantic Labels With a Common Multi-Scale Convolutional Architecture. pages 2650–2658, 2015. 2

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture Synthesis Using Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 28*, 2015. 3, 4

[11] U. Güçlü and M. A. J. v. Gerven. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. *The Journal of Neuroscience*, 35(27):10005–10014, July 2015. 8

[12] D. J. Heeger and J. R. Bergen. Pyramid-based Texture Analysis/Synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 229–238, New York, NY, USA, 1995. ACM. 3

[13] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001. 1

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 3

[15] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 2

[16] S.-M. Khaligh-Razavi and N. Kriegeskorte. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Comput Biol*, 10(11):e1003915, Nov. 2014. 8

[17] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised Learning with Deep Generative Models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[19] M. Kümmerer, L. Theis, and M. Bethge. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet. In *ICLR Workshop*, 2015. 2, 8

[20] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003. 1

[21] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *Visualization and Computer Graphics, IEEE Transactions on*, 19(5):866–885, 2013. 8

[22] H. Lee, S. Seo, S. Ryoo, and K. Yoon. Directional Texture Transfer. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '10, pages 43–48, New York, NY, USA, 2010. ACM. 1

[23] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. pages 3431–3440, 2015. 2

[24] A. Mahendran and A. Vedaldi. Understanding Deep Image Representations by Inverting Them. *arXiv:1412.0035 [cs]*, Nov. 2014. arXiv: 1412.0035. 3, 6

[25] J. Portilla and E. P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, 40(1):49–70, Oct. 2000. 3, 4

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*, Sept. 2014. arXiv: 1409.0575. 3

[27] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*, Dec. 2013. 3

[28] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014. arXiv: 1409.1556. 3

[29] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000. 2

[30] L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000. 1

[31] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, page 201403112, May 2014. 8

[32] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997. 6