Exploring the Neural Algorithm of Artistic Style

Yaroslav Nikulin[*] and **Roman Novak**[*]
Department of Mathematics
École normale supérieure de Cachan
94230 Cachan, France
{yaroslav.nikulin, rnovak}@ens-cachan.fr

Abstract

In this work we explore the method of style transfer presented in [1]. We first demonstrate the power of the suggested style space on a few examples.

We then vary different hyper-parameters and program properties that were not discussed in [1], among which are the recognition network used, starting point of the gradient descent and different ways to partition style and content layers. We also give a brief comparison of some of the existing algorithm implementations and deep learning frameworks used.

To study the style space further, an idea similar to [2] is used to generate synthetic images by maximizing a single entry in one of the Gram matrices $\mathcal{G}_l$ and some interesting results are observed. Next, we try to mimic the sparsity and intensity distribution of Gram matrices obtained from a real painting and generate more complex textures.

Finally, we propose two new style representations built on top of network's features and discuss how one could be used to achieve local and potentially content-aware style transfer.

1. Introduction

The key idea behind style transfer is to perform a gradient descent from random noise minimizing the deviation from content (i.e. feature responses in the upper convolutional layers of a recognition network) of the target image and the deviation from the style representation of the style image. The latter is defined as a set of Gram correlation matrices $\{\mathcal{G}_l\}$ with the entries $\mathcal{G}_{ij}^l = \langle F_i^l, F_j^l \rangle$, where l is the network layer, i and j are two filters of the layer l and F_k^l is the array (indexed by spatial coordinates) of neuron responses of filter k in layer l. In other words, $\mathcal{G}_{ij}^l$ value says how often features i and j in the layer l appear together. Please refer to [1] for details.

On figure 1 we demonstrate the spectacular performance of the algorithm by running 500 iterations of the L-BFGS [3] optimization algorithm on a photo of a cat.

In section 8 we consider cases where the algorithm doesn't work and suggest possible solutions in section 9.

2. Frameworks and Implementations

We have tried running implementations on Caffe [4] (using CUDA and OpenCL [5] backend) by Frank Liu [6] and on Torch [7] by Kai Sheng Tai [8] and by Justin Johnson [9].

We have observed the OpenCL Caffe backend to be very promising but yet unstable compared to CUDA. Otherwise, both Torch implementations turned out to be significantly faster.

We have thus built our work on top of the Torch implementation by Kai Sheng Tai. Interestingly, the Torch cunn [10] backend performed slightly better than cuDNN [11] by NVIDIA.

3. Networks

In [1] the VGG-19 [12] recognition network is used to produce results. We compare the impact of using other networks (AlexNet [13], GoogLeNet [14], VGG-16 and VGG-19) in figure 2, with AlexNet performing similarly to GoogLeNet and VGG-16 similarly to VGG-19.

VGG networks perform much better at style transfer due to their architecture. For example, AlexNet and GoogLeNet strongly compress the input at the first convolutional layer using large kernels and stride (11×11 with stride 4 and 7×7 with stride 2 respectively) and thus a lot of fine detail is lost. VGG networks use 3×3 kernels with stride 1 at all convolutional layers and thus capture much more information.

We have therefore used the VGG-19 network for all our experiments.

---

[*]Authors contributed equally to this work.

4. Initialization

In [1] gradient descent is always performed from white noise. We try and compare two other initialization points: content and style. We demonstrate the impact of the initialization strategy in figure 3, starting the gradient descent from the content photo, the style artwork or from white noise.

The results highlight well the highly non-convex nature of our objective and that the starting point has a tremendous influence on the basin of attraction that the gradient descent will converge to. We naturally observe that starting from the content lets us preserve the most of it, while starting from the style image is prone to "leaking" the content of the artwork into the final result. This also serves to reinforce the observation made in [1] that style and content are not strictly separable.

We find that for most practical applications starting from the content image produces the best result. This corresponds well to the way most artworks are produced – starting from the artist observing the content and drawing a rough sketch (i.e. content reconstruction comes first), and only then applying paint (i.e. style) on top.

Note that noise initialization is still very useful for testing, benchmarking and hyper-parameter tuning. For example, starting from content and observing no change one might wonder whether the content weight is too high or the learning rate is too small. Such questions do not occur when starting from noise.

5. Partial Style Transfer

In [1] the impact of shrinking the style layer pool (from using the first 5 convolutional layers {conv1-1,...,conv5-1} down to using only the first convolutional layer {conv1-1}) is demonstrated. The authors observe how the style feature scale and complexity goes down as they consider using lower and lower convolutional layers to represent style. In general, one doesn't benefit from using only the lower layer style features (apart from the computation facilitation) and is better off simply reducing the style weight in the optimization objective.

In our work we consider keeping the upper convolutional layers and removing the bottom ones, while enforcing them as part of the content pool.

For example, instead of using {conv4-2} as the content layer and {conv1-1,...,conv5-1} as the style layers, we could try to set {conv1-1, conv2-1, conv4-1, conv4-2} as content and {conv3-1, conv5-1} as style layers (see figure 4). Notice how the partial style transfer managed to reshape the content and make it more rectangular (according to the style image) while preserving the original colors (which are mostly captured by the features in the bottom layers).

We thus conclude that relaxing the bottom layer style constraints and enforcing them instead as the content constrains allows us to retain the colors and low-level features of the content photo and only transfer mid- to high-level properties of the style, combining them into a visually appealing result.

6. Generating Styles

In order to better understand the style space constructed in [1] we draw inspiration from the Deep Visualization technique [2]. We disregard all the Gram matrices except for one with a single non-zero entry and descend from white noise without content constraints. The motivation is to understand what parts of style a single element can describe and verify if the Gram matrices present a basis in the style space qualitatively similar to the basis of VGG features in the space of natural images. By varying the non-zero element and its magnitude we can generate some curious textures with complexity increasing from the first to the fourth layer (see figure 5).

Next we attempt to generate some more sophisticated styles. For this purpose we consider Gram matrices of a single painting and visualize its histogram. We observe that sparsity and amplitude of the Gram matrix elements increase from the first to the fifth layer (of course, this doesn't necessary need to generalize, yet it seems to be in accordance with the CNN paradigm where more complex and more discriminative features are constructed from simpler ones). Although it is impossible to judge about the distribution of Gram matrices describing styles with such a limited sample, we can try to mimic the sparsity and amplitude of Gram matrices representing the "Starry Night" by van Gogh [15]. We generate a random matrix using absolute values of Gaussian distribution and apply a random sparse zero-one mask to it. We vary only two parameters: the variance of the Gaussian distribution and the sparsity of the mask. Interestingly enough, with such a simple construction, we were able to generate some intriguing textures (see figures 6 and 7) suggesting that style density estimation and generation might be a promising research direction.

7. Spatial Style Transfer

In [1] each style layer l with k features introduces a $k \times k$ Gram matrix $\mathcal{G}^l$ with feature correlation entries $\mathcal{G}^l_{ij} = \langle F^l_i, F^l_j \rangle$. This makes the style completely invariant to the spatial configuration of the style image (which is by design).

As an experiment, we suggest a new style representation designed to capture less of the artistic details and more of the spatial configuration of an image with roughly the same computational and storage complexity.

We construct $\mathcal{G}^l$ matrices of size $k \times (2X-1) \times (2Y-1)$ (where X and Y are the spatial dimensions of the layer l) with entries

$$\mathcal{G}_i^l(x,y) = (F_i^l * F_i^l)(x,y),$$

where $*$ stands for full 2D convolution. We thus impose a soft constraint on the feature distributions across the spatial coordinates (note that in principle we could store all pairwise convolutions of F_i^l and F_j^l, but such an experiment would be computationally infeasible).

In order for this objective to work we need to also rescale both content and style images to the same dimensions and modify the error derivative in [1] accordingly:

$$\frac{\partial E_l}{F_i^l(x,y)} = \frac{1}{N_l^2 M_l^2} \left[(\mathcal{G}^l - A^l)_i \circ F_i^l \right](x,y),$$

where $\circ$ stands for valid correlation.

We demonstrate this new approach on classic style transfer and on style reconstruction from noise (without content constraints) in figure 8.

Notice how differently the proposed algorithm scales: it is easy to optimize on top layers (high k, small X and Y) but is expensive on bottom layers (low k, huge X and Y). This is contrary to the algorithm in [1], where the computation cost mostly depends on k^2 and thus grows from bottom to top layers.

8. Illumination and Season Transfer

Not all artistic styles are transferred equally well by the algorithm in [1]. If the style is highly repetitive and homogeneous over the whole style image (see abstract art, textures and patterns), it can be transferred with remarkable quality. However, once it becomes more subtle and varied within the image (see Renaissance, Baroque, Realism etc), style transfer falters. It fails to capture properties like the dramatic use of lighting, exaggerated face features etc. These properties get averaged-out, as the style representation is computed over the whole image.

The same problem (i.e. global style representation) stands in the way of season and illumination transfer, as these properties change different elements of the scene differently.

We present some relatively successful examples of season and illumination transfer on images that are well aligned and are fairly repetitive in figures 9, 10, 11, 12.

9. Towards Content-Aware Style Transfer

The examples presented in figures 9, 10, 11, 12 are quite bad, but they aren't *too bad*. Indeed, one can easily spot a lot of regions where the texture was luckily transferred

correctly. This serves to indicate that in principle the style representation developed in [1] is capable of capturing photorealistic properties.

What remains is developing a content-aware style transfer, i.e. transferring style according to the content matches between the image to be repainted and the style image. Below we discuss some possible leads towards implementing such a task.

A direct approach could be to replace the global style values as defined in [1]

$$\mathcal{G}_{ij}^l = \langle F_i^l, F_j^l \rangle$$

with localized values of

$$\mathcal{G}_{ij}^l \begin{pmatrix} x \\ y \end{pmatrix} = \sum_{-s^l \leqslant dx, dy \leqslant s^l} w \begin{pmatrix} dx \\ dy \end{pmatrix} F_i^l \begin{pmatrix} x+dx \\ y+dy \end{pmatrix} F_j^l \begin{pmatrix} x+dx \\ y+dy \end{pmatrix}$$

where we capture only feature correlations in a small (s^l) region around a point (x,y) and we make the contributions decay as they get more distant from the point of interest:

$$w \begin{pmatrix} dx \\ dy \end{pmatrix} = \frac{1}{1+dx^2+dy^2}.$$

We can then replace the global style loss

$$E_l = \frac{1}{4N_l^2 M_l^2} \left\| \mathcal{G}^l - A^l \right\|_2^2$$

as defined in [1] with a global style-content covariation loss

$$E_l \sim \left\| \sum_{x,y} \left(\mathcal{F}_k^{c,l} \begin{pmatrix} x \\ y \end{pmatrix} \mathcal{G}_{ij}^l \begin{pmatrix} x \\ y \end{pmatrix} - \mathcal{P}_k^{c,l} \begin{pmatrix} x \\ y \end{pmatrix} A_{ij}^l \begin{pmatrix} x \\ y \end{pmatrix} \right) \right\|_2^2$$

where $\mathcal{F}_k^{c,l}(x,y)$ is the weighted content response of neurons of filter k in layer c reachable from (x,y,l) and the norm is that of a 3-dimensional tensor indexed with i,j and k.

Of course such an objective dramatically increases the computational cost of our problems and while could be efficiently parallelized, would probably still remain unfeasible.

In our implementation we make some very rough assumptions to test it: we consider $\mathcal{F}^c \equiv \mathcal{P}^c$ (content of images is aligned; we thus perform a locality-sensitive style transfer), $s = 0$ (pixel-wise style) and $w \equiv 1$ (no distance decay). This leads to a simplified expression of

$$\mathcal{G}_{ij}^l = F_i^l \odot F_j^l$$

and, consequently,

$$\frac{\partial E_l}{\partial F_i^l(x,y)} = \frac{1}{N_l^2 M_l^2} \left[(\mathcal{G}_i^l - A_i^l) \begin{pmatrix} x \\ y \end{pmatrix} \right] \left[F_i^l \begin{pmatrix} x \\ y \end{pmatrix} \right]^T.$$

We were only able to test this approach on small images (see figure 13 and 14). Note that due to very rigid constraints the style picture basically gets painted over the content image.

The next step within this approach would be to expand the style window s^l and see whether a locality-sensitive style transfer is feasible. If it yields good results, the content-aware transfer could be investigated.

We believe that if implemented well, such an algorithm could tackle more exquisite artistic styles, season transfer, illumination transfer, super-resolution and possibly many other applications.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015. [Online]. Available: http://arxiv.org/abs/1508.06576.

[2] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015. [Online]. Available: http://arxiv.org/abs/1506.06579.

[3] W. Commons. (2015). Limited-memory bfgs, [Online]. Available: https://en.wikipedia.org/wiki/Limited-memory_BFGS.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014. [Online]. Available: http://caffe.berkeleyvision.org/.

[5] AMD. (2016). Opencl-caffe, [Online]. Available: https://github.com/amd/OpenCL-caffe.

[6] F. Liu. (2015). Style-transfer, [Online]. Available: https://github.com/fzliu/style-transfer.

[7] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: a matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011. [Online]. Available: http://torch.ch.

[8] K. S. Tai. (2015). Style-transfer, [Online]. Available: https://github.com/kaishengtai/neuralart.

[9] J. Johnson. (2015). Neural-style, [Online]. Available: https://github.com/jcjohnson/neural-style.

[10] Torch. (2015). Cunn, [Online]. Available: https://github.com/torch/cunn.

[11] NVIDIA. (2015). Cudnn, [Online]. Available: https://developer.nvidia.com/cudnn.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/pdf/1409.1556.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014. [Online]. Available: http://arxiv.org/abs/1409.4842.

[15] V. V. Gogh. (1889). Starry night, [Online]. Available: https://en.wikipedia.org/wiki/The_Starry_Night.

[16] R. Roberts. (2012). Fawkes, [Online]. Available: http://www.allofthisisforyou.com/gallery-2/fawkes2012/.

[17] A. Grey. (1996). Despair, [Online]. Available: http://alexgrey.com/art/paintings/soul/despair/.

[18] (), [Online]. Available: http://hqwallbase.pw/105645-strange-birds/.

[19] D. Kuzmenka. (2008). Hatosia, [Online]. Available: https://www.fractalus.com/dan/Galleries/Fractals/Fractal%20Gallery%208/slides/Hatosia.html.

[20] Pando. (2013). Square vp jared fliesler joins matrix partners as a general partner, [Online]. Available: https://pando.com/2013/03/07/square-vp-jared-fliesler-joins-matrix-partners-as-a-general-partner/.

[21] (), [Online]. Available: http://wallpapercave.com/wp/Gu6gpja.jpg.

[22] M. Apostolescu. (). Midnight cat, [Online]. Available: http://www.013a.com/html/cat.htm.

[23] —, (). Farewell mr. eldritch, [Online]. Available: http://www.013a.com/html/skull_color.htm.

[24] (). Using science fiction to teach creative thinking, part three: steampunk, [Online]. Available: http : / / creativendeavors . blogspot . com / 2014 / 01 / using- science- fiction- to- teach-creative_3779.html.

[25] (), [Online]. Available: http : / / wallpapercave.com/wp/1sGOysa.jpg.

[26] (), [Online]. Available: http : / / whvn . cc / 186977.

[27] PetFinder. (2015). The special grooming needs of a senior cat, [Online]. Available: https : / / www . petfinder . com / cats / cat - grooming / grooming-needs-senior-cat/.

[28] G. Balla. (1923). Pessimismo e optimismo, [Online]. Available: http : / / www . wikiart . org / en / giacomo - balla / pessimism - and - optimism-1923.

[29] T. C. Fedro. (1969). Cubist 9, [Online]. Available: http : / / www . ebsqart . com / Art - Galleries / Contemporary - Cubism / 43 / Cubist- 9/204218/.

Figure 1. Transferring different styles [16–26] to a photo of a cat [27] (top-left).

Figure 2. Style transfer using GoogLeNet (left) and VGG-19 (right).

Figure 3. Impact of the initialization point on the final result. Top row: the source photo (left), the style image (right, [28]). Bottom row: results of 500 iterations starting from style (left), content (center) and noise (right) images.
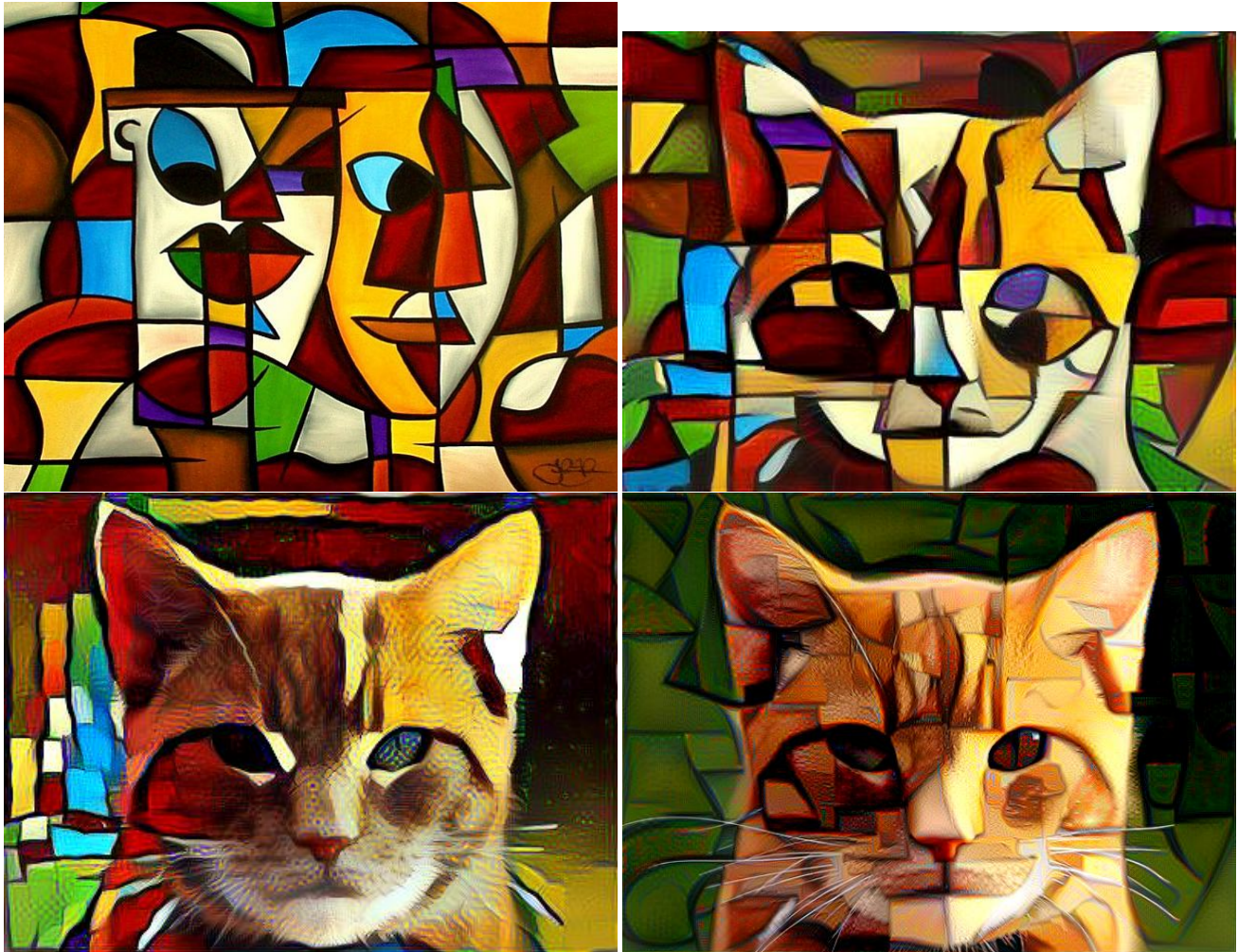
Figure 4. Using {conv1-1, conv2-1, conv4-1, conv4-2} as content and {conv3-1, conv5-1} as style layers to repaint a photo of a cat. Style image (top-left, [29]), full (top-right), low-weight (bottom-left) and partial style transfer (bottom-right). Notice that simply using a low style weight does not allow to reproduce the same result.

Figure 5. Styles generated by targeting one Gram matrix $\mathcal{G}^l$ having a single non-zero entry. Top to bottom: layer l from 1 to 4 of the target Gram matrix $\mathcal{G}^l$. Position of the non-zero element is either fixed (left) or generated randomly at each gradient descent iteration (right)
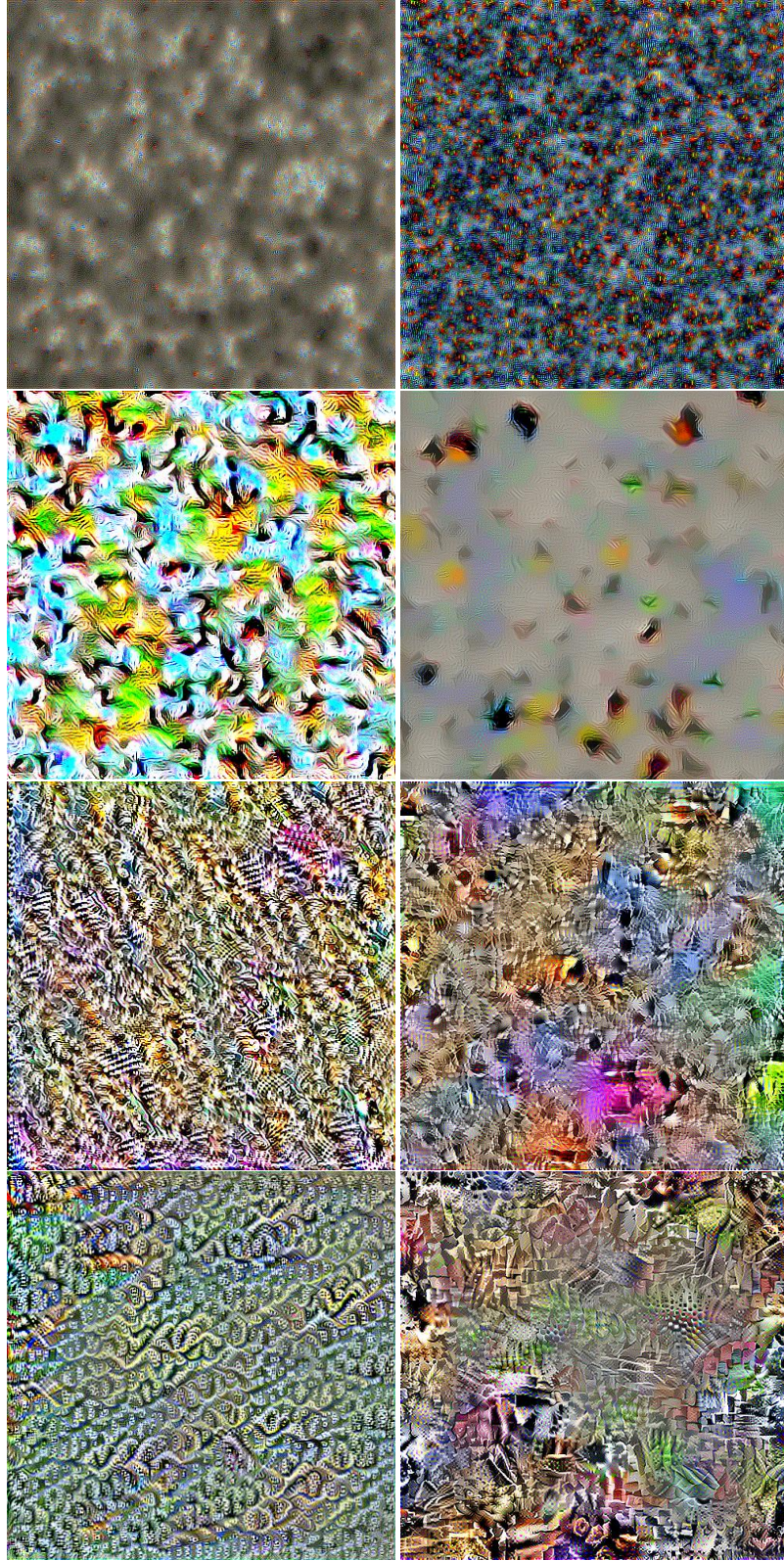
Figure 6. Styles generated by targeting one random sparse Gram matrix $\mathcal{G}^l$. Top to bottom: layer l from 1 to 4 of the target Gram matrix $\mathcal{G}^l$. The matrix is either fixed (left) or generated at each gradient descent iteration (right).
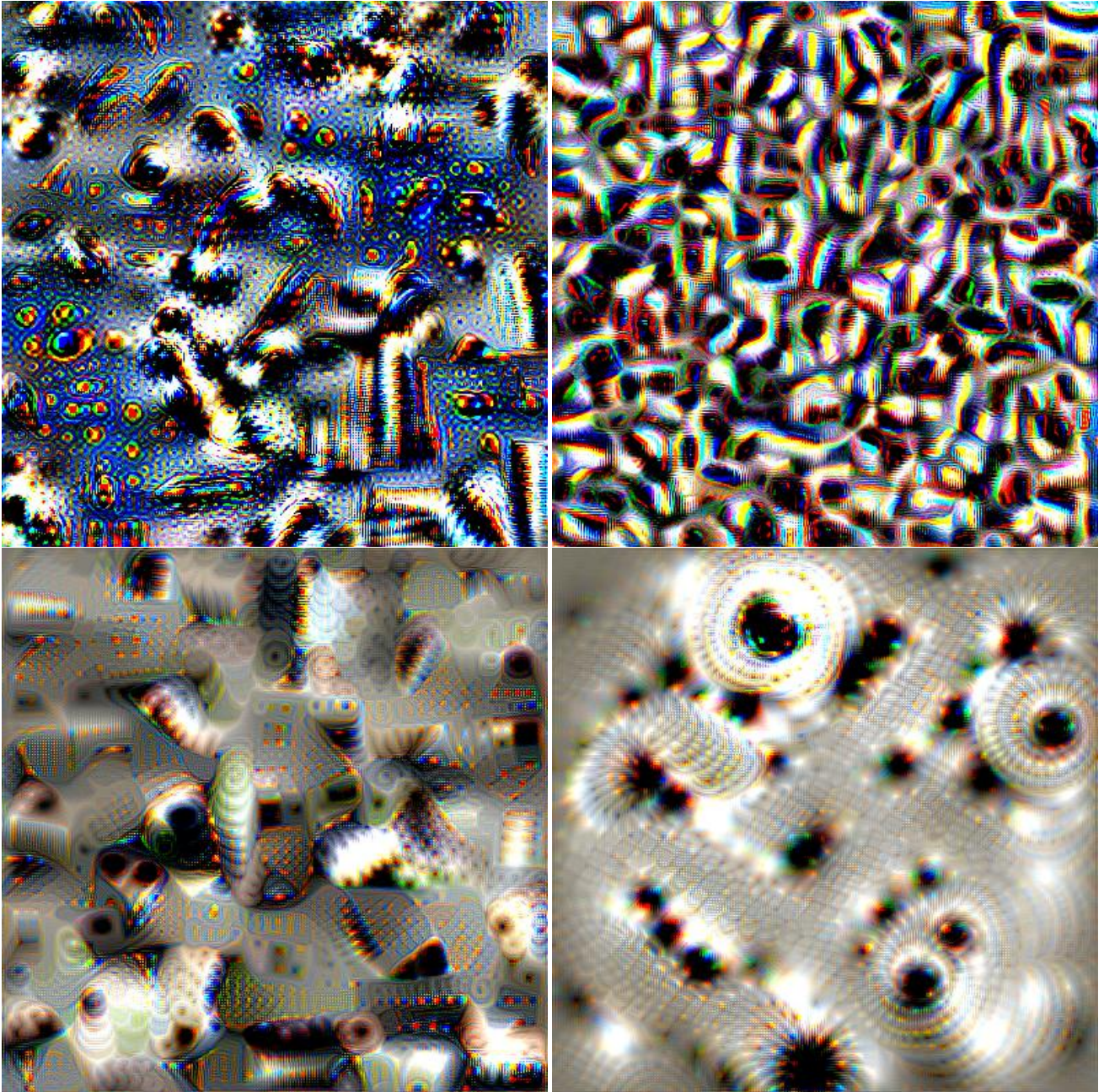
Figure 7. Styles generated targeting multiple random (but fixed for the whole optimization procedure) sparse Gram matrices. Target matrices are at convolutional layers $\{1, 2, 4\}$, $\{1, 2, 3\}$ (top row), $\{1, 4\}$, $\{1, 2, 5\}$ (bottom row).
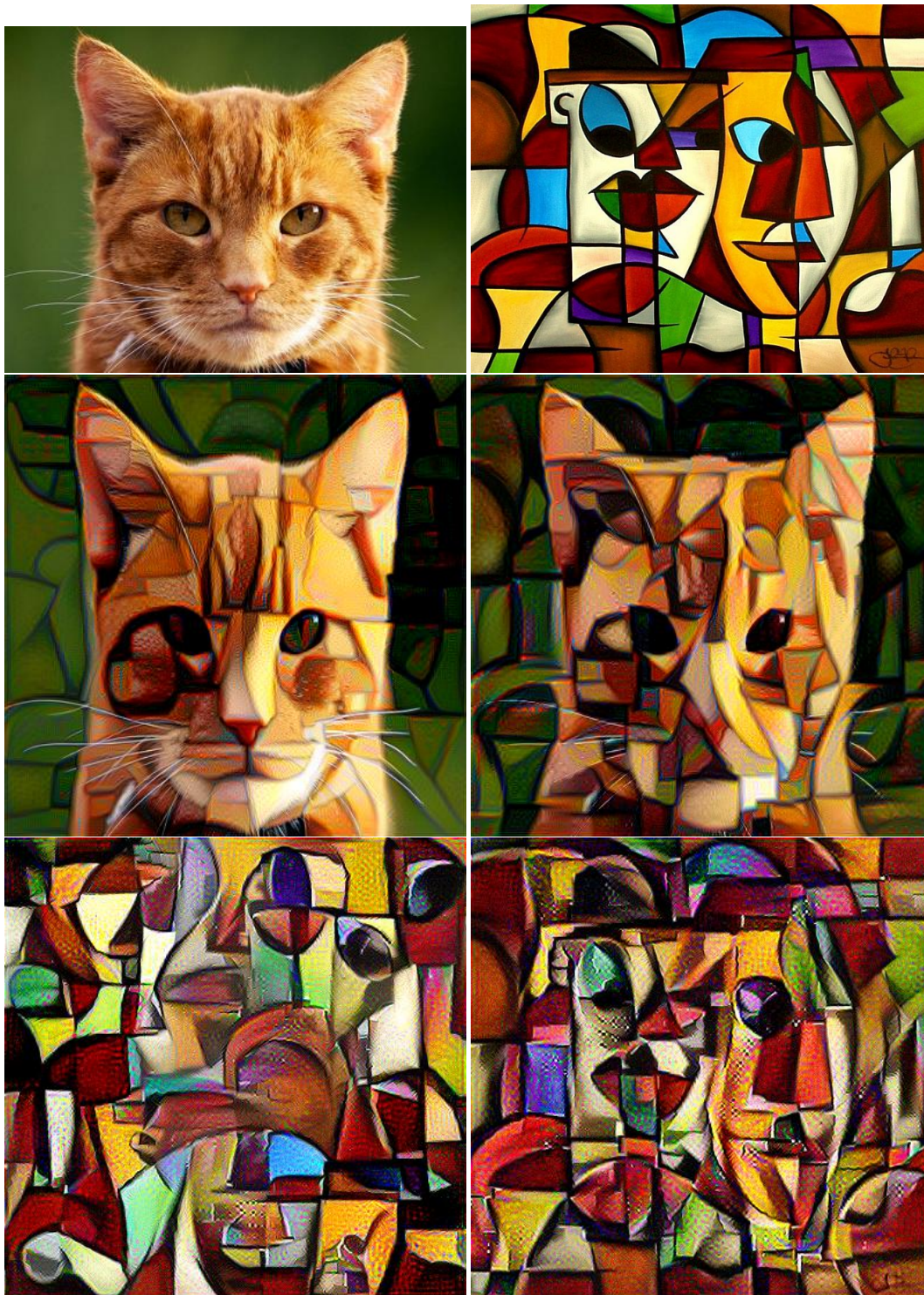
Figure 8. Comparing the conventional and the "spatial" style transfer. Top row: content (left) and style (right); middle row: conventional (left) and "spatial" (right) style transfer; bottom row: style reconstruction from noise with no content constraint using conventional (left) and "spatial" (right) representations. Notice how the spatial style transfer gently imprints the scene structure into the photo. Reconstruction from noise demonstrates how it arrives at a rough but not exact configuration of the style scene.

Figure 9. An attempt of season transfer. Top row: content (left) and style (right). Bottom row: reconstruction from content (left), style (center) and noise (right). Notice how the problem of global style is especially well observed when descending from white noise, because random textures are generated everywhere and partially stay on the sky / ground, which is undesirable.
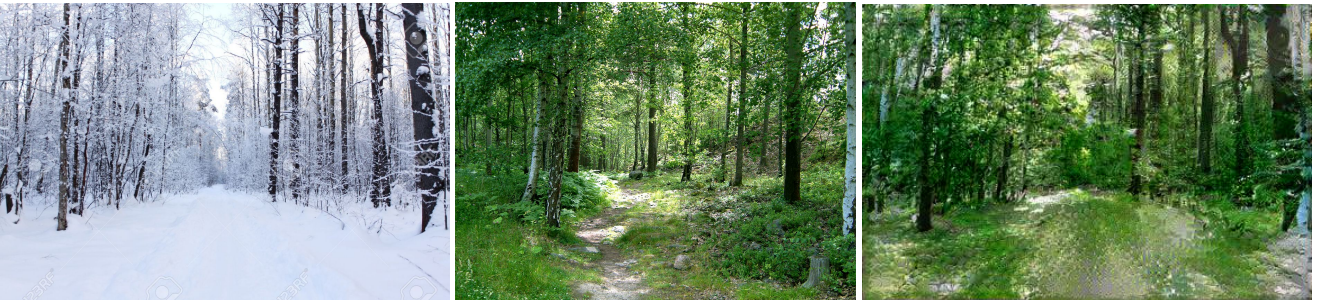

Figure 10. Another attempt of season transfer. Content (left), style (center) and the transfer result (right).

Figure 11. Yet another attempt of season transfer. Content (left), style (center) and the transfer result (right).

Figure 12. An attempt of illumination transfer. Content (left), style (center) and the transfer result (right).
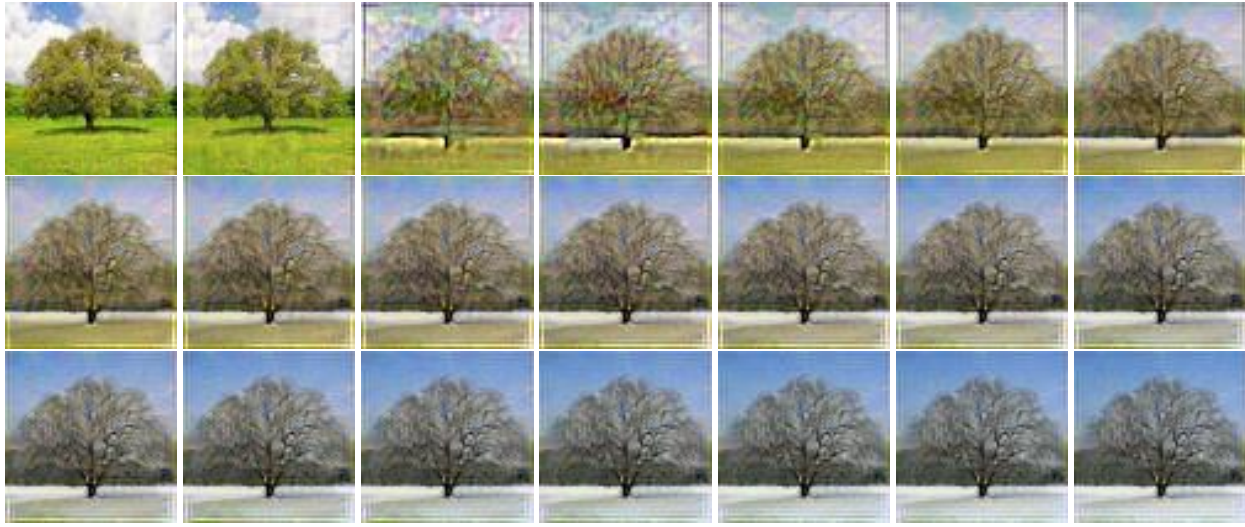
Figure 13. Summer to winter transition using a very rough approximation of a locality-sensitive style transfer. Due to very rigid constraints the style gets basically painted over the content.

Figure 14. Winter to summer transition using a very rough approximation of a locality-sensitive style transfer. As earlier, due to very rigid constraints the style gets basically painted over the content.