Deep Style Transfer

Andrea Battistello
Politecnico di Milano
andrea.battistello@mail.polimi.it

Fabio Chiusano
Politecnico di Milano
fabio.chiusano@mail.polimi.it

Samuele Conti
Politecnico di Milano
samuele.conti@mail.polimi.it

Abstract

Transferring the style of an image onto another image is a challenging task. In recent years, Gatys et al. demonstrated the power of Convolutional Neural Networks (CNN) in combining the content of an image using the style of another image. Since then, several approaches to style transfer have been proposed to either extend or improve this original work. This review aims to provide an overview of the current progress towards style transfer and a comparative analysis of state-of-the-art algorithms.

1. Introduction

The problem of synthesizing content by example is a classic problem in computer vision and graphics. It is of fundamental importance to many applications, including creative tools such as high-level interactive photo editing [[1], [2], [8]], as well as scientific applications, such as generating stimuli in psycho-physical experiments [6]. While this task can be performed by skilled humans with advanced image manipulation tools, it is not easy to find an algorithmic counterpart of it. Before the advent of CNNs, researchers tried to attempt this task with several approaches:

Stroke-based Rendering: This approach consists of incrementally composing virtual strokes (e.g. brush strokes, tiles, stripples) upon a digital canvas until they match the content image, thus producing artistic images that resemble the content. However, it's not possible to transfer style in a photorealistic setting. [7]

Image analogies: Image analogy aims to learn a mapping between the content image and the target in a supervised manner. This requires a dataset with both stylized and unstylized images, which is usually unavailable in practice. [8]

Image filtering: The iterative application of several filters can achieve the result of a fully stylized image, like in [21] where this approach is used to obtain cartoon-like effect. Although straightforward and efficient, this approach suffer from limited style diversity.

Texture Synthesis: In this approach, the style is considered as a kind of texture. By constraining the semantic of the texture to the original content image, we can obtain the stylized image by texture transferring. This approach is used in [6], but suffer from low-level image features, which limits their performance.

These approaches lack of a higher-level representation of the image and this is the main reason why CNNs greatly improved the performance of style transfer algorithms. When CNNs are trained on object recognition, they develop a high-level representation of an image leveraging on simple pixel values [5]. This can be seen by visualizing the information at each layer via reconstruction from feature maps [18]. This high-level representation of images is the key observation that marked the beginning of modern neural style transfer.

2. Related work

2.1. Style transfer with CNN

Recently, inspired by the power of Convolutional Neural Network (CNN), Gatys et al. [5] first studied how to use CNN to reproduce famous painting styles on natural images. They obtained the image representations derived from CNN and found that the representations of image content and style were separable. Based on this finding, Gatys et al. proposed a Neural Style Transfer algorithm to recombine the content of a given photograph and the style of well-known artworks.

$$Image = Content + Style$$

The key idea behind this algorithm is to start from random noise as the initial result and then change the values of pixels iteratively until the desired statistical feature distribution is satisfied. The CNN provides a good representation

Figure 1: Image as content + style.

of the image at different layers of abstraction. In order to transfer the style from an image onto another, we can define two constraints: the first one is modeled as content loss (the final image must resemble the content image) and the second one is the style loss (the final image must have the style of the style image). Gatys defined style transfer task as to minimize:

$$\mathcal{L}_{Gatys} = \alpha\mathcal{L}_c + \beta\mathcal{L}_s$$

Where:

$$\mathcal{L}_c = \sum_{\ell=1}^{L} \frac{w_{c,\ell}}{2N_\ell D_\ell} \sum_{i=1}^{N_\ell}\sum_{p=1}^{D_\ell} (F_\ell[O] - F_\ell[I])^2_{ip}$$

$$\mathcal{L}_s = \sum_{\ell=1}^{L} \frac{w_{s,\ell}}{2N_\ell^2} \sum_{i=1}^{N_\ell}\sum_{j=1}^{N_\ell} (G_\ell[O] - G_\ell[S])^2_{ij}$$

Where L is the total number of convolutional layers of the used CNN, N_ℓ the number of filters in layer ℓ, D_ℓ the number of activation values of each filter at layer ℓ. $F_\ell[\cdot] \in \mathbb{R}^{N_\ell \times D_\ell}$ is a matrix of activation coefficients for each filter. $G_\ell[\cdot] = F_\ell[\cdot]F_\ell[\cdot]^\intercal \in \mathbb{R}^{N_\ell \times N_\ell}$ is the corresponding Gram matrix, $w_{c,\ell}$ and $w_{s,\ell}$ are weights controlling the influence of each layer for respectively content and style. α and β control the tradeoff between the content and the style. A graphical representation of the algorithm of Gatys et al. is in figure 2.

2.1.1 Advanced loss-based refinements

Johnson et al. [10] showed that adding a total variation loss produces smoother outputs thus improving style transfer results:

$$\mathcal{L}_{tv}(O) = \sum_{x,y} (O_{x,y} - O_{x,y-1})^2 + (O_{x,y} - O_{x-1,y})^2$$

where the sum is over all the (x, y) pixels of the output image O.

Another addition in the style loss function has been done by Wilmot et al. [20] who showed that $\mathcal{L}_{Gatys}$ is unstable and produce output images where the brightness and contrast vary significantly throughout the image. This is due to not providing guarantees that the mean or variance of the texture is preserved. To reduce this instability, they proposed to add an histogram loss:

$$\mathcal{L}_{hist} = \sum_{\ell=1}^{L} \gamma_\ell \sum_{i=1}^{N_\ell}\sum_{p=1}^{D_\ell} (F_\ell[O] - R_\ell[O])^2_{ip}$$

where

$$R_\ell[O] = \texttt{histmatch}(F_\ell[O], F_\ell[S])$$

A different approach was taken by Luan et al. [15] that tries to consider style transfer for photorealistic images. In fact, the approach from Gatys et al. does not work on photos, because it produces effects on the output image that are not adequate in photorealistic settings because of spatial inconsistency, which may be acceptable in paintings. To increase photorealism in style transfer, they constrain the transformation from the input to the output to be locally affine in colorspace and express this constraint with a differentiable energy term that acts as a regularization term.

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^\intercal \mathcal{M}_I V_c[O]$$

Where V_c is the vectorized output image in channel c and $\mathcal{M}_I$ is a matrix that represents the set of linear standard equations that minimize the least-squares penalty of the Matting Laplacian (see [15], [12] for reference).

2.2. Transfer style from semantically similar patches

The Gram matrix encodes the style of the whole style image: better results can be obtained by taking the style from the patch in the style image, whose content is the most similar to the one in the current patch in the content image. In this way, perturbations from other regions of the style image can be avoided. In this paper we decided to focus on this problem since, in our opinion, it's the one providing the most sensible results on the output image. As regards the algorithms presented in the following paragraphs, we will explain only their contributions related to the problem of finding semantically similar patches, neglecting other modifications because of space limitations.

2.2.1 Deep image analogies with Markov Random Fields

Li and Wand [13] built a nearest neighbors Markov Random Field (MRF) to match content patches to the most semantically similar style patch. In this way we avoid artifacts due to inconsistent style matches. The matching of pixels from I to S at layer ℓ is found with normalized cross-correlation distance:

$$\boldsymbol{\psi}^\ell_{I\to S}(p) = \arg\min_q \frac{N^\ell_I(p) N^\ell_S(q)}{|N^\ell_I(p)||N^\ell_S(q)|}$$

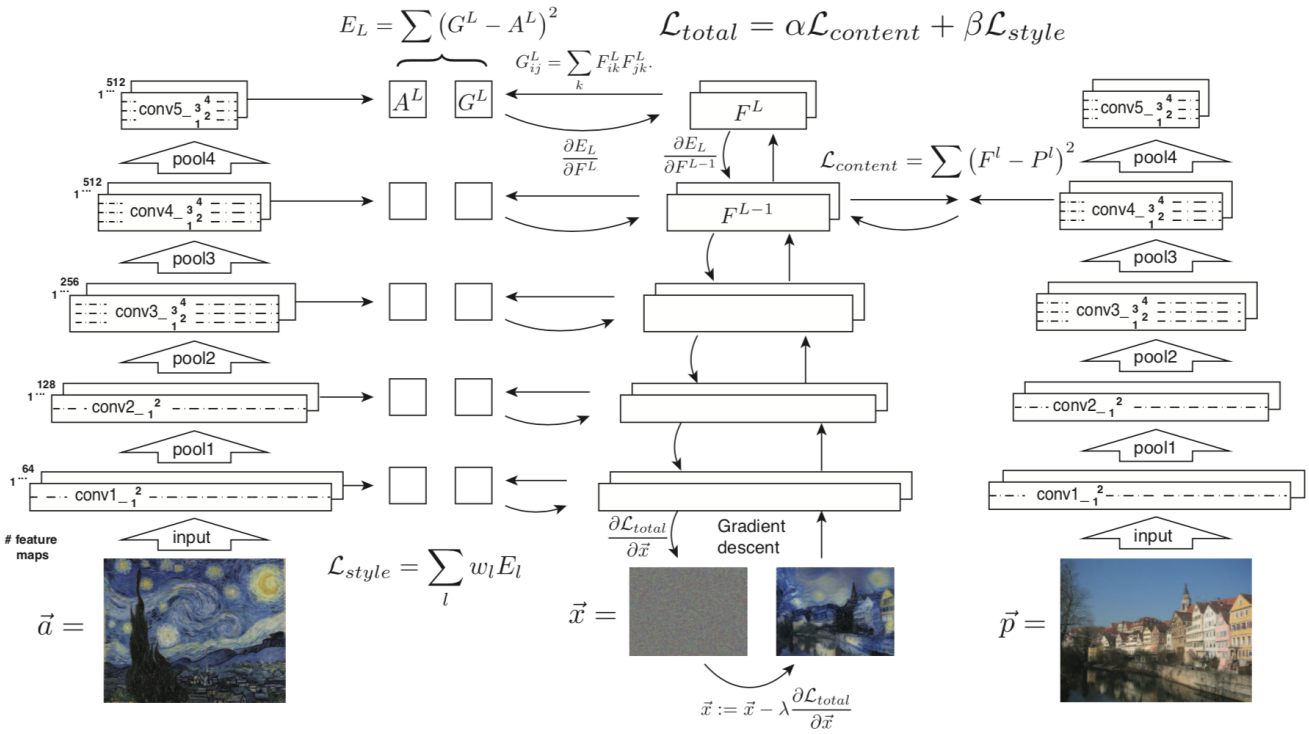


Figure 2: How the algorithm of Gatys at al. works, taken from [5]. $\vec{p}$ is the content image, $\vec{a}$ is the style image. Both content, style and output images pass through a CNN pre-trained for image classification (e.g. VGG16). At each layer, the activations of the output image are directly compared to the ones of the content image, while they are indirectly compared to the ones of the style image through their Gram matrices.

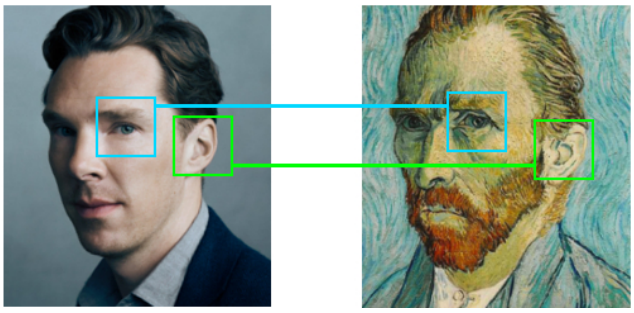


Figure 3: Example of Nearest Neighbour Field for finding similar patches.

where $N(p)$ is the patch around pixel p, i.e. the matrix of $k \times k$ pixels centered in p, being k is the size of the patch. The matching is done with PatchMatch algorithm [2], which is a randomized algorithm faster than brute-force search but empirically proven to provide good results, and the following loss substitutes the style loss:

$$\mathcal{L}_{MRF} = \sum_{\ell}^{L} \gamma_\ell \sum_{p=1}^{P_\ell} \left\| N_O^\ell(p) - N_S^\ell(\psi^\ell_{I \to S}(p)) \right\|^2$$

Where P_ℓ is the number of pixels in layer ℓ.

2.2.2 Bidirectional constraint

Liao et al. improve the nearest neighbor field NNF [14] by considering bidirectional correspondence between the content and the style image. The algorithm produces two images from the content image I and the style image S':

- I': content of I, style of S'
- S: content of S', style of I

In this way it's possible to build a better NNF by considering a double mapping of the patches between the images.

$$\psi^\ell_{I \to S}(p) = \arg\min_q \left(\left\| N_I^\ell(p) - N_S^\ell(q) \right\|^2 + \left\| N_{I'}^\ell(p) - N_{S'}^\ell(q) \right\|^2 \right)$$

2.2.3 Enforce spatial consistency

Another approach in improving the patch mapping is the one found in [16]. At first, an initial guess of the mapping is built according to [13]. Then, it is improved by adding

Figure 4: Nearest Neighbour Field built upon the bidirectional constraint defined in [14], explained in 2.2.2.

a bonus whenever contiguous patches are mapped to contiguous patches. For example, starting from pixels p in the content image I, a candidate pixel q in the style image S that matches p is built considering $down(\boldsymbol{\psi}^{\ell}_{I\rightarrow S}(up(p)))$, where the $down(.)$ and $up(.)$ functions return respectively the first patch above and under the passed patch.

Figure 5: Nearest neighbor field built according to [16], explained in 2.2.3. Similar neighboring patches help in finding a better match.

2.3. Style transfer related challenges

Several modifications of style transfer algorithms have been devised to address different challenges. **Fast style transfer**: running the algorithms from scratch to transfer the same style on different images seems a waste of resources. Leveraging on this idea, [10] train a neural network on a specific style, so that it transforms the input image to a new image with same content but with the style the network was trained on. Therefore, time is spent on training the neural network so that style transfer can be applied with a single forward pass. As described by [9], another possible approach is provided by using Conditional Adversarial Networks (cGAN) as a general-purpose solution to image-to-image translation problems: these networks are not only able to learn a mapping between input and output images, but also a loss function to train this mapping. This approach, due to its generality, can be applied in the same way to many different tasks that would required very different loss formulations, avoiding the need of hand-engineering both the mapping and the loss function for the task at hand. **Semantic augmentations**: [4] semantic annotations are proposed to augment the CNNs architecture for style transfer, in order to obtain more control over the final outcome and increase its quality. [3] builds on a patch-based approach, where the information of the semantic map is used to compute the nearest-neighbour patches and contributes to the loss. It also shows that existing patch-based algorithms require minor adjustments and perform very well using this additional information. **Video**: [19] style transfer can be applied to video too, but special attention must be taken in enforcing consistency between adjacent frames.

3. Experiments

3.1. Datasets, Experiments setup

We tested state-of-the-art algorithms for style transfer on a set of common used images for content and style. We ran two types of experiment to investigate the algorithms. The first experiment uses an implementation of Gatys et al. and investigates the parameter space that motivated the choices made by the authors of the paper. The second experiment, instead, aims to compare the results of several algorithms on several content and style choices.

3.2. Experiment 1: Gatys parameters comparison

This experiment aims at exploring the meaning of the parameters involved in style transfer in the simple and more intuitive implementation of Gatys et al. [5] We decided to investigate along the following dimensions: **Content layer**: the layer to use as a representation of the content. In figure 6 we notice that choosing layer Conv5_2 gives a more abstract representation of the content, becoming sometimes blurry and less detailed, while Conv3_2 bounds the output to be more similar to the content. Layer Conv4_2 is a good compromise. **Style layer**: the layers used to represent the style of the image. We compared the choice of Gatys et al (column 1) with deeper convolutional layers in the network. This resulted in more noisy results, meaning that the style representation in deeper layers is too specific. **Image initialization**: the initial value of the output image. We can see that with random initialization we have possibly more artistic results, but sometimes too noisy because of the lack of a strong constraint on the content representation. **Optimization algorithm**: we used Adam [11] and L-BFGS [22] as optimization algorithms. We can see that L-BFGS is more suitable to optimize from random initialization than Adam and possibly gives more artistic results in general. However, such observations are subject to personal taste. As a general perspective, we can say that this approach is quite robust to the choice of hyper-parameters, because almost all the images (with the exception of the ones started from random) are very similar to each other.

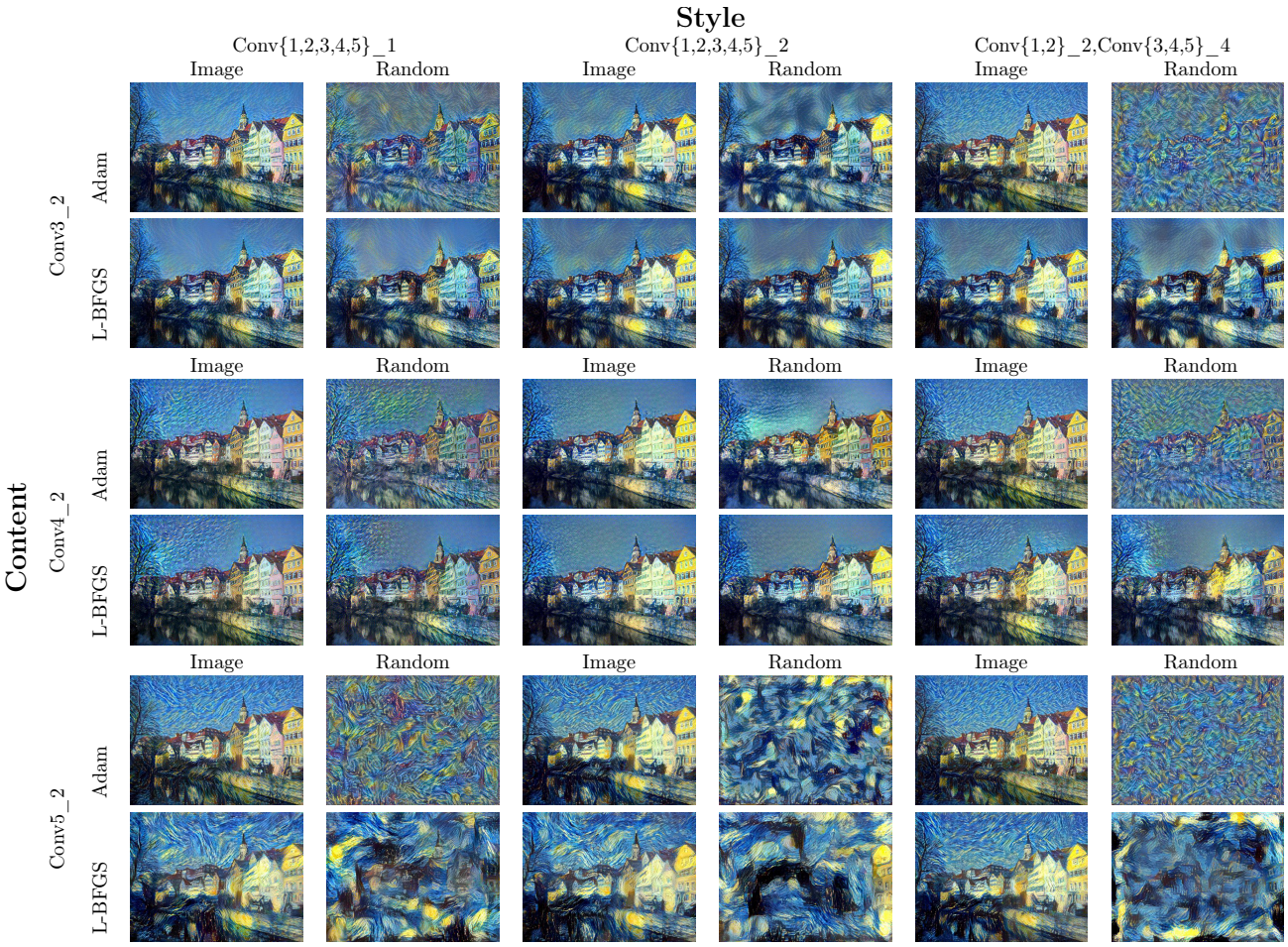


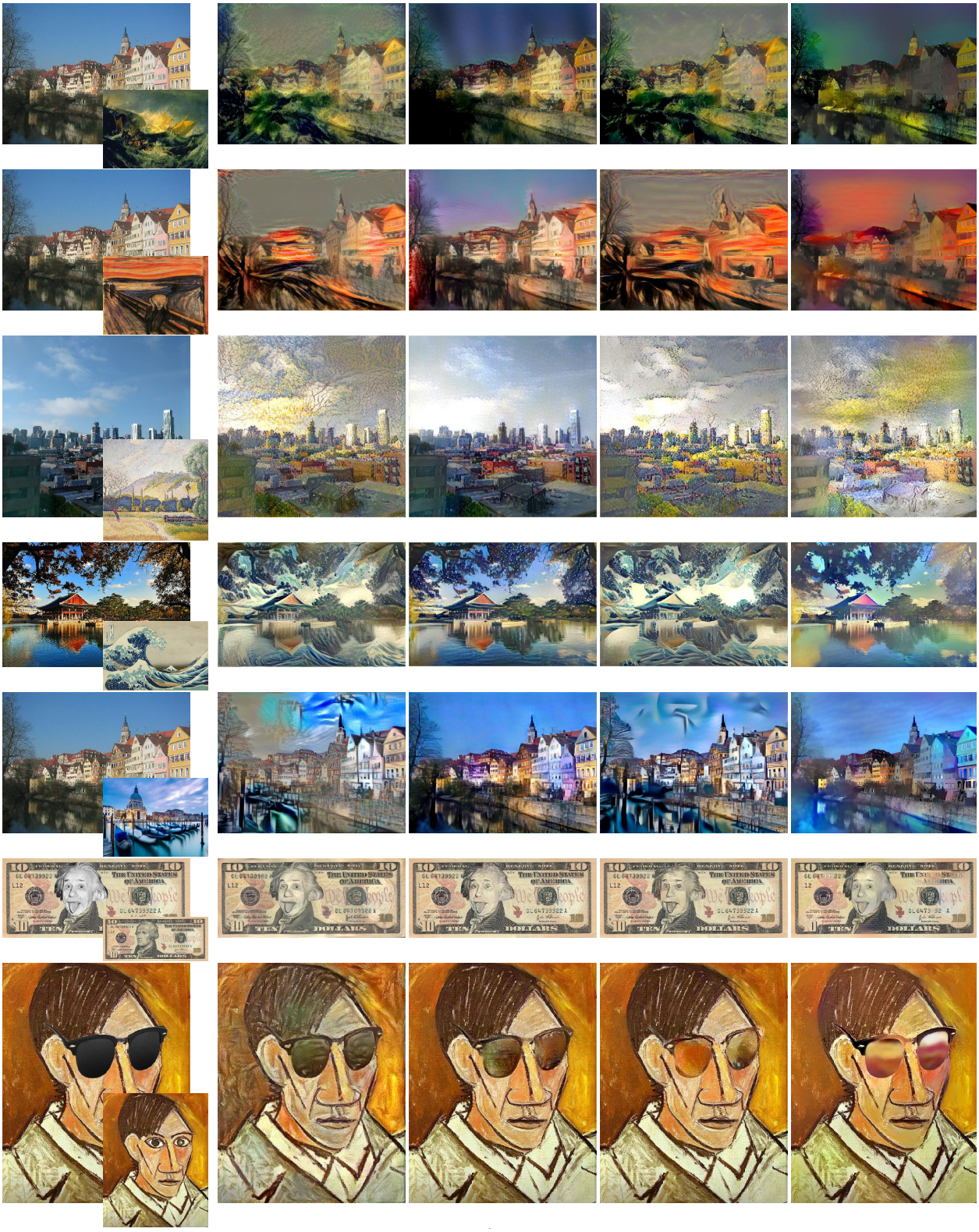
Figure 6: Output of Gatys for different content layer, style layer, optimizer and initialization

3.3. Experiment 2: Algorithms output comparison

This second experiment aims at comparing different state-of-the-art algorithms. We have chosen to compare the naìve but powerful algorithm from [5], described in 2.1, with other – more complicated – algorithms: MRF [13], described in 2.2.1, performs an additional matching by building a Markov Random Field on the activation patches when building the Gram matrix. DPA [17], described in 2.2.3, builds upon this image and adds another pass with a more constrained MRF patch matching, plus a histogram and total variation loss. PHOTO [15] constrains the transformation to lie in color space using Matting Laplacian as described in section 2.1.1. We tested the behaviour of the algorithms in different and realistic scenarios: paint-to-photo, photo-to-photo and painterly harmonization. In the first scenario, we transfer the style of a paint onto a realistic photo. The second scenario involves the transfer of the style of a photo onto another photo. The third scenario adds an extra step of harmonization, where the content is similar to the style with the except of an extraneous object with a complete different style. The goal is to smoothly transform the added object to match the style of the painting. As a result, we can see that the simpler algorithm form Gatys performs remarkably well on every test with few exceptions. More complex approaches (PHOTO, MRF and DPA) work really well on a limited number of images, but fail to generalize on every domain. Results can be seen in figure 10

4. Conclusion

Style transfer is a really hot topic in machine learning because of its many applications in graphics and photo manipulation. Simple approaches like Gatys et al. work remarkably well on various domains and require little to no hyperparameter tuning. Although more advanced approaches are available and give really good results on some domains, they can also perform very bad on other domains, questioning whether the more complex model was necessary in the first place.



(a) Content and Style (b) Gatys (c) MRF (d) DPA (e) PHOTO

Figure 7: Comparison of Gatys, MRF DPA and PHOTO algorithms on different contents and styles

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 294–302, New York, NY, USA, 2004. ACM. 1

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 24:1–24:11, New York, NY, USA, 2009. ACM. 1, 3

[3] A. J. Champandard. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *ArXiv e-prints*, Mar. 2016. 4

[4] Y.-L. Chen and C.-T. Hsu. Towards deep style transfer: A content-aware perspective. pages 8.1–8.11, 01 2016. 4

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 1, 3, 4, 5

[6] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR*, abs/1505.07376, 2015. 1

[7] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 453–460, New York, NY, USA, 1998. ACM. 1

[8] 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*, SIGGRAPH '01, pages 327–340, New York, NY, USA, 2001. ACM. 1

[9] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 4

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 2, 4

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. 12 2014. 4

[12] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 61–68, Washington, DC, USA, 2006. IEEE Computer Society. 2

[13] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. *CoRR*, abs/1601.04589, 2016. 2, 3, 5

[14] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *CoRR*, abs/1705.01088, 2017. 3, 4

[15] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. *CoRR*, abs/1703.07511, 2017. 2, 5

[16] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep painterly harmonization. *arXiv preprint arXiv:1804.03189*, 2018. 3, 4

[17] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep painterly harmonization. *CoRR*, abs/1804.03189, 2018. 5

[18] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. *CoRR*, abs/1412.0035, 2014. 1

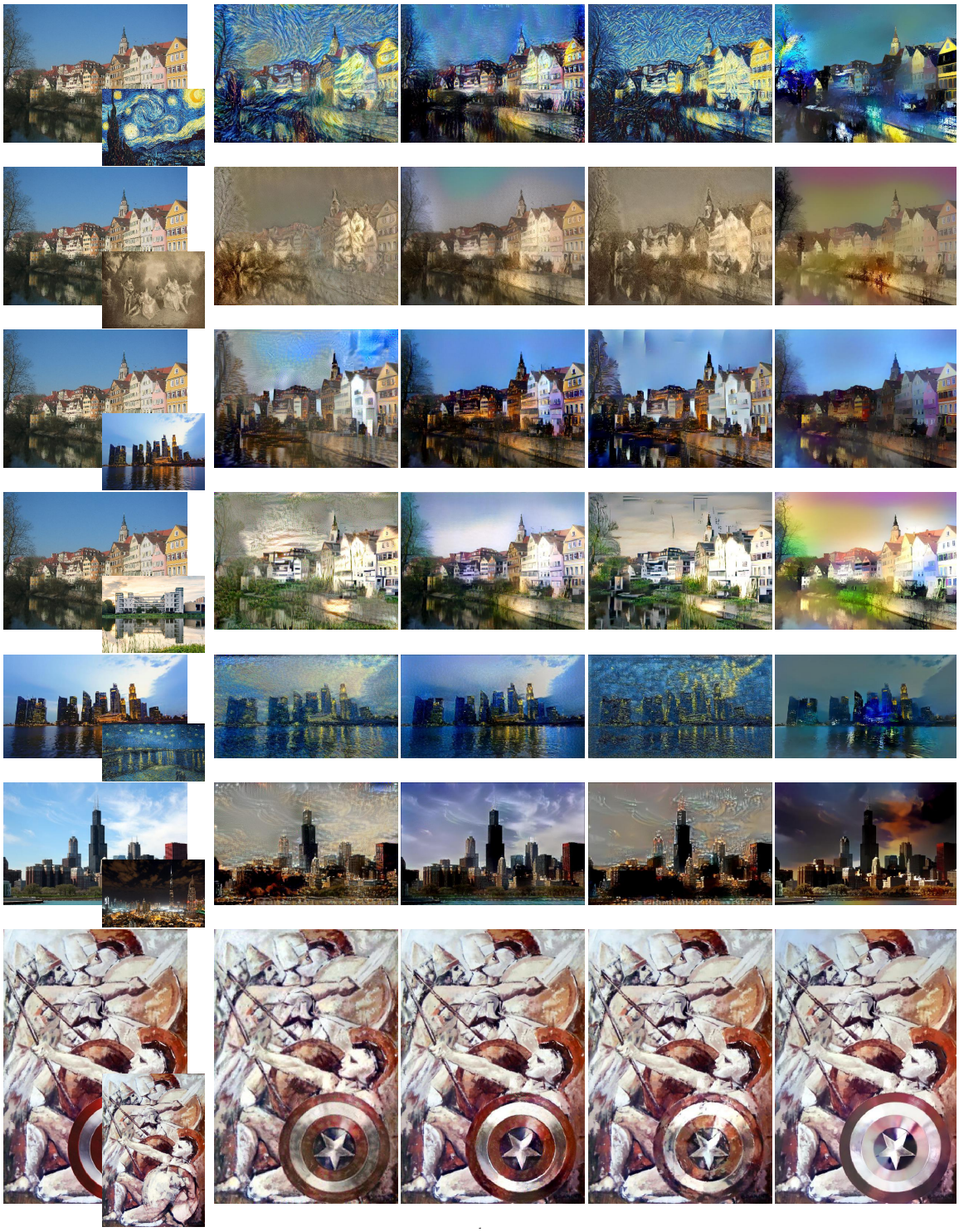
[19] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. In B. Rosenhahn and B. Andres, editors, *Pattern Recognition*, pages 26–36, Cham, 2016. Springer International Publishing. 4

[20] P. Wilmot, E. Risser, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *CoRR*, abs/1701.08893, 2017. 2

[21] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1221–1226, New York, NY, USA, 2006. ACM. 1

[22] Y. Xiao, Z. Wei, and Z. Wang. A limited memory bfgs-type method for large-scale unconstrained optimization. *Computers & Mathematics with Applications*, 56(4):1001 – 1009, 2008. 4

A. More comparisons



(a) Content and Style (b) Gatys (c) MRF (d) DPA (e) PHOTO

Figure 8: More Comparisons of Gatys, MRF DPA and PHOTO algorithms on different contents and styles

B. Histogram loss

(a) Content and Style (b) $w_{HIST} = 0$ (c) $w_{HIST} = 100$ (d) $w_{HIST} = 500$ (e) $w_{HIST} = 1000$

Figure 9: Effect of histogram loss using Gatys

Histogram loss limits the difference between the histogram of the output image and the histogram of the output image reconstructed through histogram matching with the style image. Thus, it forces the output image to have a color distribution similar to the one of the style image.

C. Total variation loss

(a) Content and Style (b) $w_{TV} = 0$ (c) $w_{TV} = 0.001$ (d) $w_{TV} = 0.01$

Figure 10: Effect of total variation loss using Gatys

Total variation loss limits the difference of activations of adjacent pixels and thus produces smoother outputs.