The Computer-Science Department
The Technion – Israel Institute of Technology

Improved Image Style Transfer via CNN

This is an undergraduate Project in
the Geometric Processing Laboratory

By: Mark Erlich & Adar Elad
Supervisor: Alona Goltz

Background

Image style transfer refers to an artistic process in which a content image is modified to include the style taken from a second image. Early signs of this idea appeared in the early 2000's, but the real breakthrough came with the impressive work of Leon Gatys and his co-authors [1]. Their method relies on pre-learned neural networks, that has been trained for image recognition.

Before we proceed, let us show the results of their process, so as to give a clear context to the later discussion. Figure 1 shows two pairs of content + style images, and the fusion of the two as created by Gatys' algorithm. As can be seen, while the content image is greatly modified, the main essence of it is preserved, while borrowing artistic flavor from the style image. Observe that the texture created is faithful to the one in the style image in various scales, a fact that will be revisited in this research project.

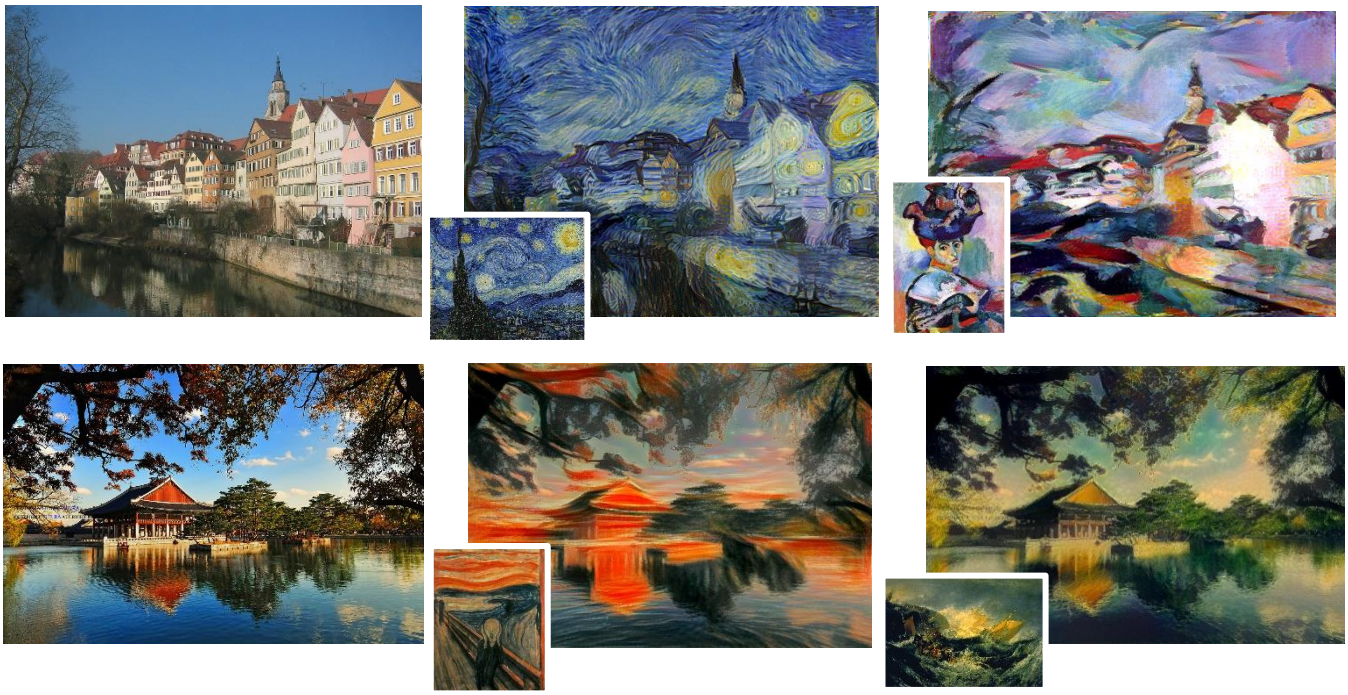

Figure 1 – Gatys et al. Results

And now let's dive into the finer details of this algorithm, which this project aims to further improve. The foundation of the fusion algorithm is the availability of a pre-trained CNN, which serves as a feature extractor. More specifically, VGG-19 has been proposed for this task. When fed with an image, the features of interest are taken in any desired inner layer, and the deeper we go, the more abstract the image description becomes. The style-transfer process starts by getting two input images – the content image denoted by C, and the style image denoted by S – and results with an output image X.

Gatys' approach towards the fusion task is to turn the mission into a minimization problem, where the unknown is the image to be generated. Two forces are operating in this context – the first forces the features of X to align with those of C, while the second force brings the effect of the style, by requiring proximity between the correlation features of S and X. Referring to the last point, the style is encapsulated in a series of cross-correlations between the various filters in any given layer of the VGG-19. As such, these lose relation to the spatial location, while focusing instead on brush-strokes, general repetitions, color gamut, and more. The following Equation is the energy function that this algorithm minimizes:

$$f(X) = \alpha \underbrace{\left\| F_k(X) - F_k(C) \right\|_2^2}_{L_{Content}(X,C)} + \beta \underbrace{\left\| CF_j(X) - CF_j(S) \right\|_2^2}_{L_{Style}(X,S)}, \quad (1)$$

and as can be seen, we have the two forces interacting with each other in order to result with X that has both the content and the style in it. The index k in the first term refers to the layer in which the features of the content are taken, and similarly, j stands for the layer to use for the correlation features of the style. The above is only a prototype form, and in fact the style and the content can be accumulated from several possible layers, as indeed practiced by Gatys.

The above minimization task requires to back-prop through the VGG-19 in order to calculate the gradient w.r.t. X. The overall algorithm starts with an initial X (typically chosen to either be C or random noise), which is updated by ~1000 iterations of the steepest-descent or close by numerical methods. Figure 2, taken from Gatys' paper, presents the overall block-diagram of this algorithm. We would like to draw the readers' attention to the fact that in this method there is no learning what so ever, as the CNN used is fixed.

The style-transfer paper by Gatys [1], which was published in CVPR June 2016, has already accumulated a huge impact, as reflected by the 600 citations in Google-Scholar. It is beyond the scope of this extended abstract to survey this vast work. Nevertheless, we would say the following. Various attempts were made to modify this algorithm and improve it, some emphasizing the idea to turn this into a feed-forward CNN [2,3], others attempting to suggest alternatives that avoid CNN altogether [4,5], several papers highlighted the desire for photo-realistic outcomes, leading sometimes to color palette transfer variations [6,7], and more. A video version of this idea has been explored as well [8,9], with varying degrees of success.

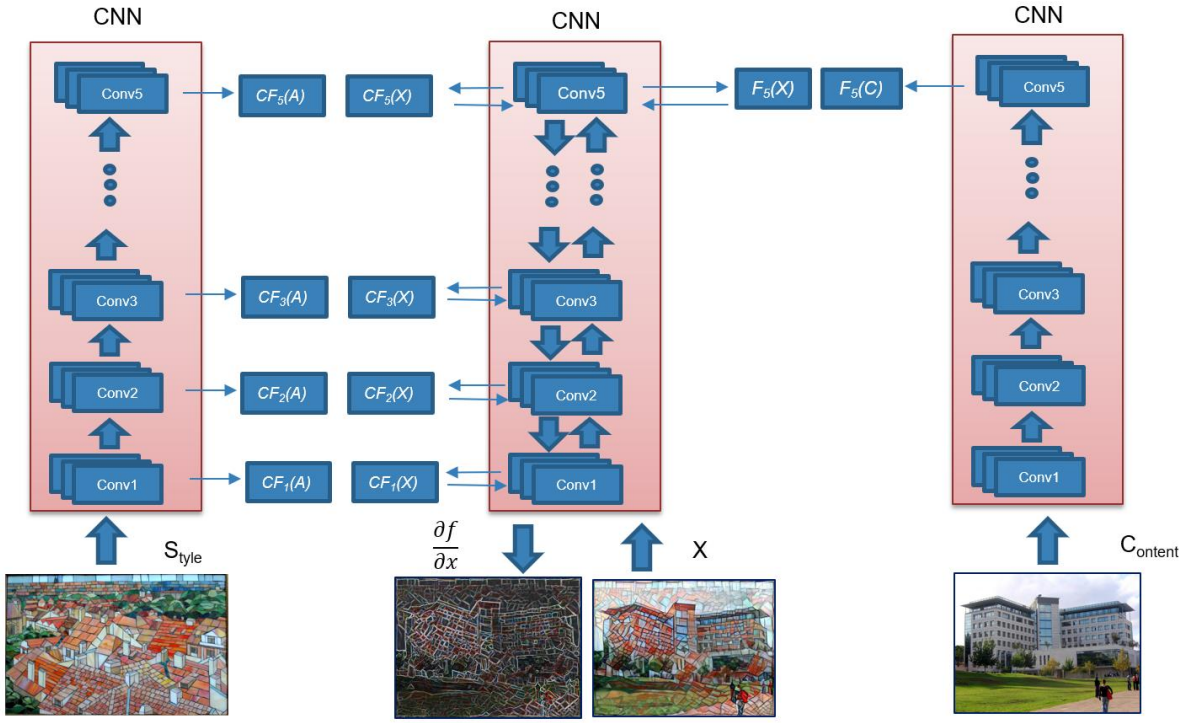

Figure 2 – The algorithm's scheme of [1]

This Project: Objectives, Contributions & Results

The results obtained by the Gatys' algorithm are often times very impressive and stimulating. The general feeling is that this machine is capable of creating automatic art out of the two images feeding the system. However, this method is not without shortcomings, which explain the many follow-up works. Our project focuses on the following prime weaknesses:

- Loss of edges from the content image in the resulting fusion,
- A limitation in the sizes of the images that can be treated, and
- The overall run-time of the transfer algorithm.

We turn to expand on each of these and present our proposed novel algorithmic remedies.

Better Treatment of Edges

Edges in the content image are not necessarily maintained, and when they are important for the outcome, Gatys' algorithm provides no direct control for their preservation. An indirect possibility is to weaken the style effect, either by reducing β (see Equation (1)), or by going into shallower layers. However, both these options ruin the overall stylization. This project presents a solution to this problem, in the form of a mask used within a modified version of the style penalty. More specifically, a mask E detecting the edge regions in the content image is created, and then plugged into the following alternative of the L_{Style} penalty:

$$L_{style}(X,S) = w_j^{(E)} \cdot \left\| CF_j(E \cdot X) - CF_j(S) \right\|_2^2 + w_j^{(1-E)} \cdot \left\| CF_j((1-E) \cdot X) - CF_j(S) \right\|_2^2 \qquad (2)$$

This enables to simultaneously transfer the style to both regions (edges and the rest of the domain), while separating their treatment. As can be seen in Figure 3, this approach leads to preservation of the edges without any harm to the overall stylization. Indeed, by replacing the

mask E with any desired alternative, we generalize Equation (2), to enable a merge of several styles into one image. Figure 4 presents few such examples.

Figure 3 – The first column presents the content images; the second displays Gatys' results for the given style image (bottom left); the third column shows the obtained results with edge preservation; and the fourth zooms in on portions to show that indeed the edges in the content image are maintained much better.

Figure 4 – Two example results of merging two styles in the Parrot image.

Style Transfer for Large-Scale Images

Gatys' solution is specifically tailored for mid-size images, and collapses when treating very large images (having thousands of pixels in rows and columns). This is a critical shortcoming in cases where the art created is to be printed in large-scale, or when shown on large screens. There are two sources for this limitation – a technical hardware memory issue, and a more fundamental algorithmic flaw. We start by addressing the later, more pressing, problem of these two.

Recall that the style-transfer algorithm relies on the VGG-19. As such, this architecture has a limited depth, which may be found as insufficient when the input images are very large. As we go deeper and deeper into this network, we get more and more abstract description of the image content. However, if the image is too large, even these abstract descriptions are of small scale, relatively to the overall image size. In such cases, the style transfer would emphasize small texture elements, and lose the bigger effects and the global view of the style image. As an example, Figure 6 presents two style-transfer results for the same two images, but held in different spatial resolutions, exposing the severity of this problem.

The solution we have developed relies on a multi-scale reformulation of Gatys' algorithm. The two original (and large) images, C and S, are converted into two corresponding Gaussian pyramids. The overall algorithm starts at the coarsest scale, feeding the scaled-down C and S to the original algorithm. The result is then scaled-up, and serves as an initialization to the next scale fusion. This proceeds until the bottom of the pyramids, in which the native scale of the images is handled (shown in Figure 5). In this process, large scale correlations are well-treated in the coarse levels of the algorithm, and inherited to their predecessors. Figure 7 is a follow up to the results shown in Figure 6, where this algorithm is deployed, showing the improved stylization effect.

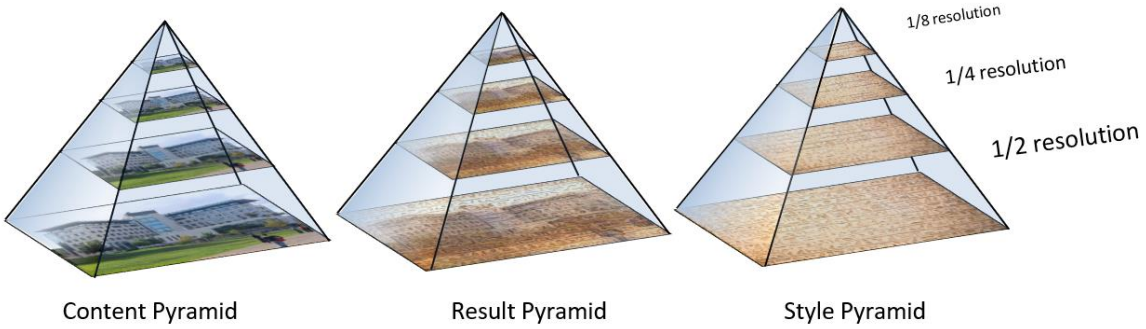

Figure 5 - Multi-Scale Style-Transfer scheme

We now return to the other source of limitation on the image sizes, mentioned above. The overall algorithm runs on GPU, as any other alternative is simply too slow. Therefore, transferring large-scale images to operate upon meets a natural hardware limitation in the GPU memory. Our solution to this problem relies on a split of the optimization posed in Equation (1) into a series of agents that serve the same overall goal. Relying on the ADMM (Alternating Direction Method of Multipliers) concept, a given global optimization problem can be cast as a series of local problems that are communicating with each other. In this spirit,

we break the image domain into a set of weakly overlapping patches, and each of these is treated separately by the Gatys' algorithm – See Algorithm 1 below. The solution for each patch is then forced to have a global agreement, leading eventually to a visually pleasing global outcome. As such, our style-transfer algorithm can operate virtually on ANY SIZE Image, employing a black-box engine of the Gatys' algorithm always running on mid-size images. Obviously, this idea is merged within the multi-scale scheme described above. We shall present results referring to this process in the Figure 8.

Run-Time Improvement

A by-product of the above two modifications, we obtain a substantial speed-up in the overall run-time of the fusion process. This is due to two main reasons: (1) the much fewer necessary iterations in each scale in the pyramid due to the warm-start idea; and (2) the overall complexity of the Gatys' algorithm is quadratic with respect to the image size, which means that operating on small patches is far more efficient.

Algorithm 1 – ADMM Style Transfer Algorithm

Input - S - Style image, C - Content image, R – Patch Extractor

Init $(\forall i)\ u_i = 0,\ X = 0$

For $t = 1 \rightarrow T$

1. $(\forall i)\quad z_i \leftarrow \underset{z}{\text{ArgMin}}\left[\alpha\left\|F_k(R_i C) - F_k(z)\right\|_2^2 + \beta\left\|CF_j(S) - CF_j(z)\right\|_2^2 + \rho\left\|z - R_i X + u_i\right\|_2^2\right]$
2. $X \leftarrow \underset{X}{\text{ArgMin}}\left[\sum_i \left\|z_i - R_i X + u_i\right\|\right]$
3. $(\forall i)\quad u_i \leftarrow u_i + z_i - R_i X$

Return X

All the results shown in this brief report were created in the course of this project by the Tensor-Flow environment, ran on Windows-10 64-bit 16GB RAM machine with a Nvidia GTX-1080 GPU.

Figure 6 – Gatys' results of the same content and style images **in different resolutions**. In each of the pairs, the left image refers to a size of 750x1000 and the right refers to a size of 384x512. Observe that an increased resolution hampers the stylization effect.

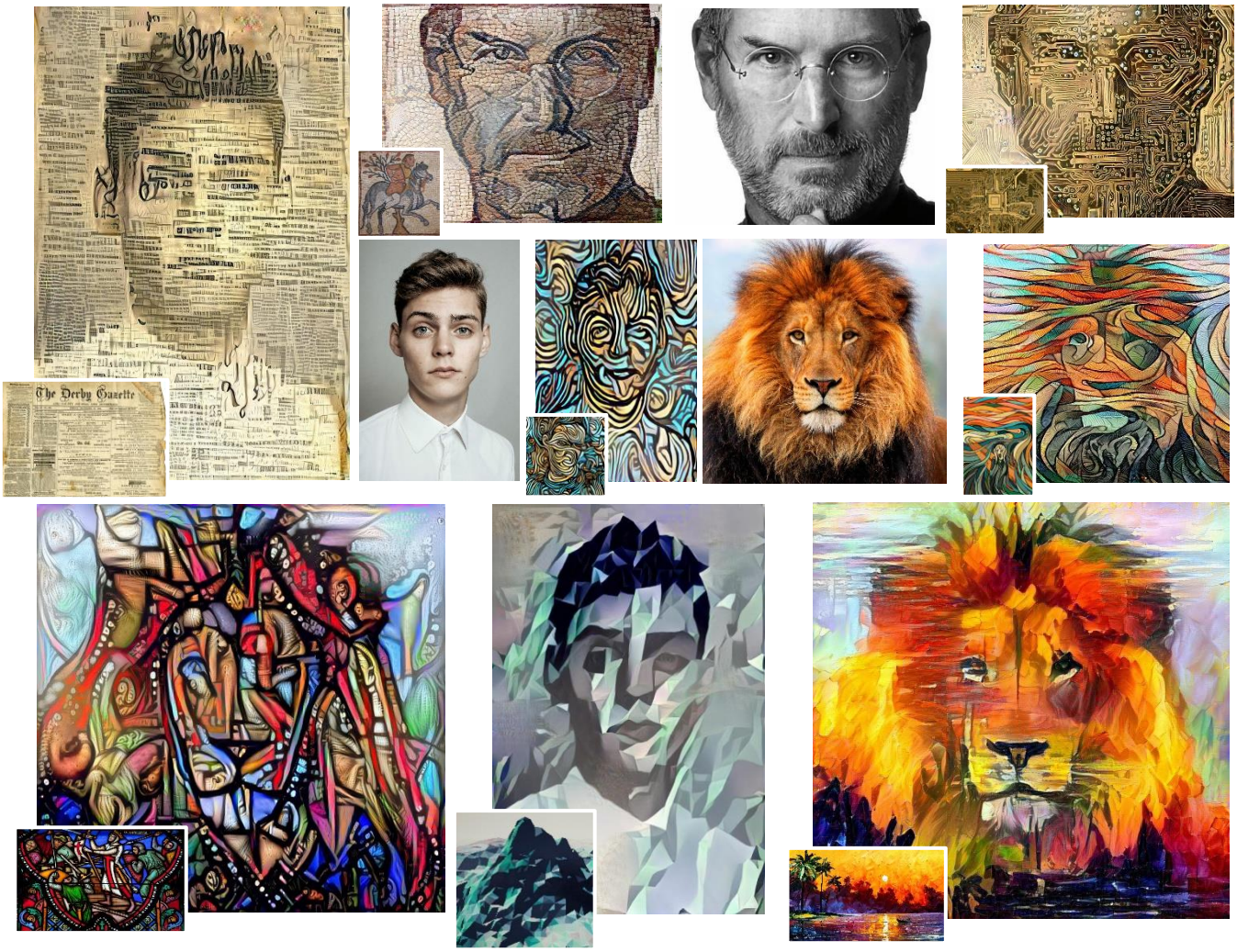

Figure 7 – Results of the proposed Multi-scale algorithm. Note that in these experiments we have chosen not to enforce the edge preservation (a matter of decision)

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Image Style Transfer Using Convolutional Neural Networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2414–2423, 2016.

[2] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super- resolution. In European Conference on Computer Vision (ECCV), pages 694–711. Springer, 2016.

[3] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In ICML, 2016.

[4] M. Elad and P. Milanfar. Style-transfer via texture-synthesis. arXiv preprint arXiv:1609.03057, 2016.

[5] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337, 2016.

[6] R. Mechrez, E. Shechtman, L. Z. Manor Photorealistic Style Transfer with Screened Poisson Equation. arXiv preprint arXiv:1709.09828, 2017.

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

[8] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. pages 1–14, 2016.

[9] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos and spherical images. arXiv preprint arXiv:1708.04538, 2017.

Figure 8 – The patch-based results

Style

Content