Less is More, Faithful Style Transfer without Content Loss

Nicholas Kolkin[1] Eli Shechtman[2] Sylvain Paris[2] Greg Shakhnarovich[1]

Toyota Technological Institute at Chicago[1] Adobe Research[2]

Figure 1: Results generated by our algorithm using drawn, painterly, and texture-based styles at 1k resolution. We use the high stylization setting of our algorithm for the texture-based styles, and default settings for the other styles. We encourage using the electronic version of the document to zoom in and observe the details.

Abstract

The dominant style transfer framework is based on separately defining 'style loss' and 'content loss', then finding an image that trades off between minimizing both. The challenge of operating in this regime is that formulations proposed so far for the 'content loss' and 'style loss' are fundamentally at odds, and generally impossible to simultaneously drive to zero. In this work we show that an explicit content loss is unnecessary. We propose Neural Neighbor Style Transfer (NNST)—a straightforward approach based on nearest-neighbors that achieves higher quality stylization than prior work, without sacrificing content preservation.

1. Introduction

The style of an artist manifests both globally, through composition and choice of subject matter, and locally, through technique and choice of media. Like other style transfer algorithms we focus on the latter, local, aspects of style. One proxy for matching the technique and medium of an artwork is to synthesize a new image using only patches taken from the artwork. Selecting which patches to use, and blending them into a natural looking output is a challenging problem in pixel space; however, like other recent work [12, 25, 29, 3, 13, 23], we tackle these challenges in the feature space of a pre-trained neural network. However, our work fundamentally differs from the currently dominant paradigm of Neural Style Transfer (NST) algorithms [12, 18, 2, 25, 37, 34, 13, 23], in that our algorithm's output is not constrained to simultaneously satisfy a 'content loss' and a 'style loss'. Instead we explicitly construct a 'target feature tensor' consisting entirely of spatially rearranged features from the style image.

We are not the first to propose a style transfer algorithm based on explicit feature matching [15, 33, 17, 10, 3, 25, 29, 13, 41]. But to the best of our knowledge we are the first general purpose style transfer method to outperform the NST state-of-the-art without using any content loss.

The first challenge is finding a good mapping between features of the content image and style image. A common failure mode of feature matching methods is all content features mapping to a few style features. This leads to an output with two major flaws. First, many details of the content image are replaced by flat regions or repetitive artifacts. Second, because only a few features are used, the output fails to mimic the distribution of style features. In this work, we evaluate several potential solutions to this problem, including: feature pre-processing, choice of distance metric, and mapping features using nearest-neighbors (NN) or optimal transport (OT). We find that a good balance between stylistic fidelity, content preservation, and computational efficiency can be achieved by: (1) pre-processing the mean of the content features, (2) measuring similarity between neural activations using cosine distance instead of ℓ_2, and (3) using nearest-neighbors. Notably, none of these design decisions add additional constraints to the final optimization

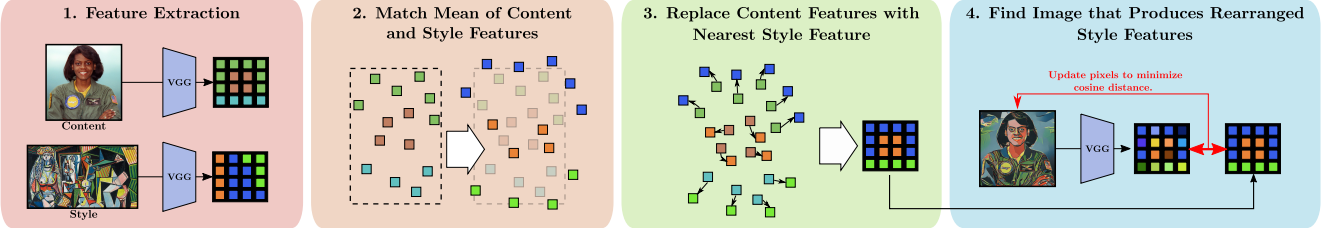


Figure 2: Overview of our method, see Section 3.2 for details.

of the output image.

Once the content features have all been replaced by style features, we must confront the second challenge, synthesizing an aesthetically pleasing image based on these features. Prior work in style transfer [13] notes that the outputs of methods using explicit feature correspondences often produce 'washed out' or desaturated results. Similar observations have been made in other patch-based synthesis work [21, 17, 10]. We identify two regimes of neural feature matching with complementary properties for style transfer, the second of which addresses this issue.

In the hypercolumn matching regime (HM) features are matched jointly at all layers using hypercolumns [35, 14]. Outputs of this approach capture many aspects of the style image while maintaining good content preservation. However, these outputs suffer from the washed out quality noted in prior work. In particular they are desaturated and do not match the style image's high frequencies.

In the second regime, which we call 'feature splitting' (FS) we take inspiration from recent work on painterly harmonization [30], and relax the constraint that entire hypercolumns be matched. Instead, features are matched separately for every layer. In contrast to [30] these matches are recomputed every iteration relative to the current output image, rather than the initial content image. This results in a final output with sharp high-frequencies and vivid colors that match the style image. However, because the matching procedure of the second regime is more flexible, jumping directly to it results in over-stylization. Our final method achieves the best of both worlds by synthesizing the final output using the second regime, but initializing with the output of the first.

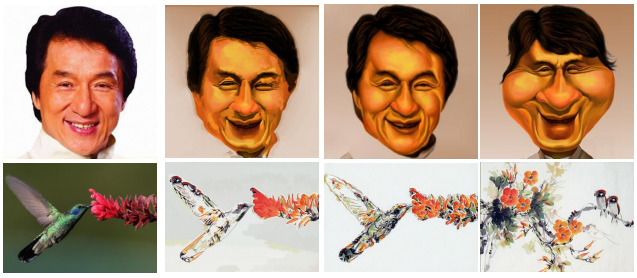
While we propose a specific algorithm that produces higher-quality style transfer results than prior work, our primary motivation is demonstrating that an explicit 'content loss' is unnecessary for high-quality style transfer, even between images with very different semantics and layout. We believe that our lack of a content loss, along with the sharper and more vivid results obtained in the FS regime relative to HM regime, provides valuable evidence that removing constraints is an important tool for improving image synthesis.

2. Related Work

Example-based style transfer, non-photorealistic rendering guided by a single piece of artwork, is a widely studied image synthesis task. Approaches like our own, in which image synthesis is guided by explicit matches between spatially localized content and style features, can be traced to [15] and related work on texture synthesis [7, 8, 42] from the late 1990s and early 2000s. Recent patch-based style transfer algorithms have focused on producing extremely high quality outputs when additional human guidance is available, either in the form of edge annotations [32, 31], or specially created style images [9, 1]. These methods are unified by their use of hand-crafted feature representations and use of nearest neighbors to find content-style correspondences.

Recent years have marked a significant departure from this line of work, building off techniques introduced by Gatys et al. in 'A Neural Algorithm of Artistic Style' [12]. This work had two major impacts. The first is leveraging features extracted by a convolutional neural network pretrained for image classification (typically VGG16 [39]). The second is directly optimizing the output's pixels using gradient descent to simultaneously minimize a 'style loss' (typically based on matching statistics derived from features of the target artwork), and a 'content loss' (typically based on minimizing deviation from the content image's features). Many works [12, 2, 25, 37, 34, 13, 23] have proposed alternate style loss formulations, and [23] proposes a relaxed content loss invariant to translations and rotoreflections in feature space.

In the example-based framework, where each style image can be arbitrary and is assumed to be previously unseen, these approaches produce the highest quality output. However, these methods face a fundamental challenge. In general it is impossible to match the statistics of the style features (satisfying the style loss) without modifying the content features (satisfying the content loss). Even if the content loss is invariant to rotoreflections [23], matching the second order statistics captured by the simplest original style loss [12, 28] generally requires an affine transformation (of which rotoreflections are a subset, and therefore do not provide enough invariance).



Content DIA [29] Ours Style

Figure 3: Deep Image Analogies (DIA) [29], another content loss free method, is designed specifically for style/content pairs with well-matched semantic, and works best when the layout and pose of the two inputs is similar. Our method, along with other generic style transfer methods, is intended to work well even for completely unrelated content/style pairs. We demonstrate our method's greater robustness, applying it to two failure cases from their paper.

Content Chen [3] Ours Style

Figure 4: In [3] Chen et al. also propose a content loss free style transfer method. However, a key difference relative to our method is that they average feature vectors when construction their 'target feature tensor'. This helps them preserve content well, but results in failure to capture distinctive elements of the style. We compare our results with theirs on two pairs of inputs taken from their paper.

Motivated by this tension, our proposed algorithm returns to the framework of guiding image synthesis with explicitly matched content and style features, although like other NST algorithms we leverage pretrained VGG16 for feature extraction.

We are not the first to revisit this framework using neural features. CNNMRF [25] replaces the style loss of [12] with minimizing each patch of content feature's distance from its nearest neighbor patch of style features under the cosine distance. In [13] Gu et al. constrain the matches found by nearest neighbors to only use each style vector at most k times. However, both of these works regularize their outputs using the content loss proposed in [12], leading to the fundamental tension outlined above.

We are aware of two similar style transfer works that also explicitly construct a set of target features and do not use a content loss. Liao et al. [29] propose a coarse-to-fine algorithm for finding feature correspondences between the content and style image. Their algorithm produces excellent results on style-content pairs with matching semantics and similar poses, but does not work for more disparate style-content pairs (see Figure 3). In [3] Chen et al. take a similar approach to [25], but average overlapping feature patches from the style image. This preserves content well, but at the cost of fidelity to the target style (see Figure 4).

Texler et al. [41] also propose a method using explicit neural feature matching, but to guide an upsampling procedure using patch-based synthesis. Their goal is to increase the resolution of a NST output (such as our own) to 4k resolution and beyond, and their method is complementary to ours.

Other recent work has used optimal transport to find mappings from content features to 'stylized' features. In [26, 36] the authors find the optimal transport map for the content features under the assumption that the true distribution of content and style features is Gaussian. This is efficient, but limits their stylization to being an affine map applied to the content features. In [23], Kolkin et al. formulate their style loss based on an efficiently computed lower bound on the earth-movers distance. However, like other methods descended from [12], their style loss is fundamentally at odds with the content loss.

Feed-Forward Style Transfer A common scenario, which we do not address in this work, is when the styles of interest are known beforehand, and a neural network can be pretrained to produce stylizations of the predetermined type(s) [19, 46, 20, 38, 24]. These methods are very fast (performing inference with the forward pass of a CNN), and can produce high quality results. However, they require enough training data, and must be retrained for any new style.

Another line of feed-forward methods is 'universal style transfer' [16, 27, 4, 45, 5, 40]. These are parametric methods which, based on statistics of the style features, derive closed form modifications of the content features, then decode the result into an image with a feed-forward network. These methods are able to offer speed and generalization to unseen styles, but at the cost of stylization quality.

3. Neural Neighbor Style Transfer

3.1. Feature Extraction

We first describe our feature extractor $\Phi(x)$, where x is an RGB image. Throughout the paper, except in Figure 1, we resize x to be 512 pixels on the long side. $\Phi(x)$ extracts the hyper-columns [35, 14] formed from the activations produced for all convolutional layers of pre-trained

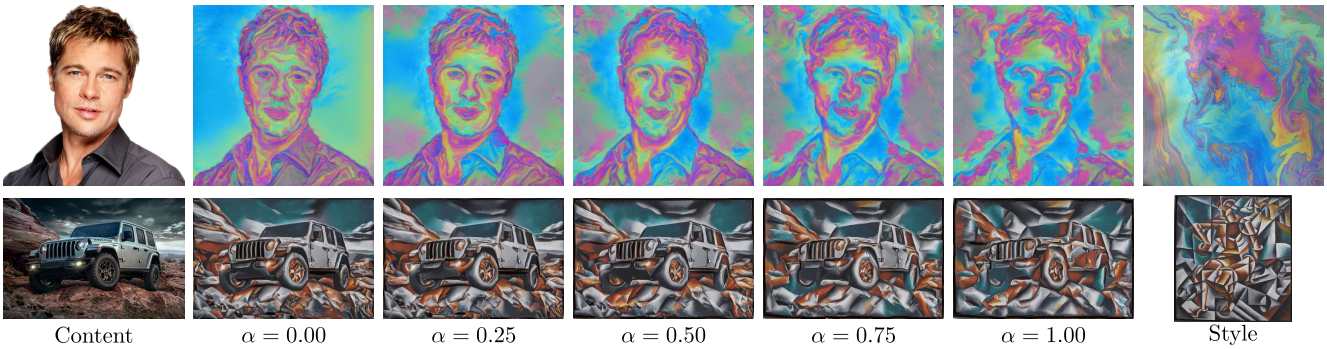


Figure 5: visualization of the method's content/style tradeoff caused by varying α. Because our final output is synthesized entirely based on features from the style image, there is non-trivial stylization even when $\alpha = 0$

VGG16 [39] when x is passed in. We use bilinear interpolation on activations from all layers to give them spatial resolution equal to one quarter of the original image. For an image with height H, and width W, this yields an image representation $\Phi(x) \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 4224}$. Generally we consider style to be rotation invariant, and to reflect this we extract features from the style image rotated at $0°, 90°, 180°$ and $270°$ in all experiments.

3.2. Image Synthesis from Hypercolumns (HM)

The core of our algorithm is a simple procedure outlined in Figure 2. We extract features from the style image and content image (1) and shift content features to match their mean with that of the style features (2). Then we use nearest-neighbors matching (3) to replace each content style feature (hypercolumn) with the closest style feature. If the content image is of size $H_c \times W_c$, and the style image is of size $H_s \times W_s$, this yields a new target representation for our stylized output $T \in \mathbb{R}^{\frac{H_c}{4} \times \frac{W_c}{4} \times 4224}$ where the feature vector $T_i \in \mathbb{R}^{4224}$ at each spatial location is taken from the style image, or a rotated copy of the style image. Finally we optimize the pixels of our output image x (4) to minimize the cosine distance loss:

$$\min_x \quad -\frac{1}{P} \sum_{i=0}^{P-1} \cos\left(\Phi_i(x), T_i\right) \tag{1}$$

where $P = W_c H_c / 16$, the number spatial locations in $\Phi(x)$ and T.

We minimize Equation 1 via 200 updates of x using Adam [22] with parameters $\eta = 2e^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. To allow the average color of large regions to quickly change within 200 updates, we parameterize x as a Laplacian pyramid with 8 levels.

We run this procedure at eighth, quarter, half, and full resolution. The upsampled output of the previous scale serves as initialization for the next. We initialize the coarsest scale with a downsampled version of the content image.

Control of stylization level We use this multi-scale procedure to control the stylization level of our final output in the following manner. Let O_s be the output of our algorithm at scale s. Let C_{s+1}, S_{s+1} be the content and style images at finer scale $s+1$. Let $O_s^{\uparrow}$ be O_s upsampled to be the same resolution as C_{s+1}. Instead of constructing T by finding matches between $\Phi(C_{s+1})$ and $\Phi(S_{s+1})$, we instead find matches between $\Phi(\alpha O_s^{\uparrow} + (1-\alpha) C_{s+1})$ and $\Phi(S_{s+1})$. The parameter α controls stylization level, with $\alpha = 0$ corresponding to the lowest stylization level, and $\alpha = 1$ the highest. We demonstrate the effect of varying α in Figure 5. By default, we set $\alpha = 0.25$, as this generally produces a visually pleasing balance between stylization and content preservation.

3.3. Feature Splitting (FS) regime

We find that replacing entire hypercolumns of content features with entire hypercolumns of style features produces results that are desaturated and fail to capture the high-frequencies of the style image (see Figure 6). We believe that this effect is due to incompatible hypercolumns, which were not adjacent in the original style image, being placed next to each other in T. Because these features have overlapping receptive fields, the output is optimized to produce the average of several features from different regions of the style in one region of the output. This manifests visually as a 'washed out' quality. We find this problem can be solved by making the feature matching less constrained, a similar observation to one made in the field of image compositing [30]. Matches are computed for each layer of VGG separately, resulting in the T consisting of novel hypercolumns where features at different layers are mixed and matched from different locations/rotations of the style image. Unlike [30] we do not compute matches only once, we recompute them after every update to the output image. Updating the output procedes as in Section 3.2, except features are matched relative to the current output, rather than the initial content. Adding this step results in vivid outputs with better

Figure 6: Ablation demonstrating the complementary roles of hypercolumn matching (HM), and feature splitting (FS). Outputs using FS only are over-stylized, while images produced by HM only are desaturated and lack the distinctive high frequencies of the style image. Initializing FS with HM (Ours) produces the best results.

stylized high frequencies (see Figure 6).

3.4. Design Decisions

When mapping from content features to style features, there are several important decisions: (1) what preprocessing to use, (2) what metric used to compare features with, and (3) the matching algorithm to use. It is worth reiterating that while these design decisions must be made carefully, they do not add additional constraints to the final optimization of the output image.

Qualitatively we find the best generic settings to be: (1) matching the means of the content and style features, (2) comparing them using cosine distance, and (3) mapping between them using nearest neighbors. In the following sections we discuss these choices.

3.4.1 Ablation: Feature Pre-Processing

Our primary goal in pre-processing the features is to prevent nearest-neighbors from overusing the same style features. We approach this as a question of how to align the distributions of content features and style features. This immediately leads to three simple options. First, we can do nothing. Second, we can align μ (replace the content features' mean with the style features' mean). Third, we can match μ and also match Σ using the WCT as suggested elsewhere in the style transfer literature [4].

We find that some pre-processing is vital, and that matching the μ only produces the best results overall. See Figure 7 for details.

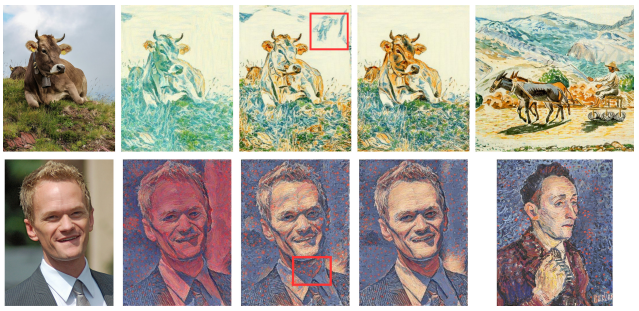


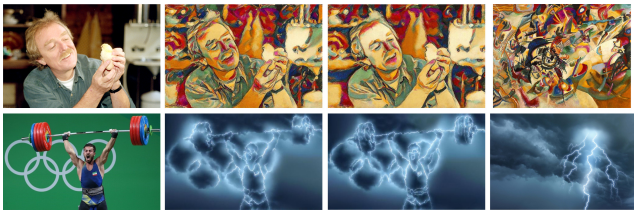
Figure 7: Without any pre-processing (N/A), a small subset of of style features is overused, resulting in homogeneous outputs with lower stylization quality and content preservation. Matching μ and Σ works reasonably, but is more computationally expensive (increasing the runtime of our algorithm by around 20%), and often introduces artifacts (e.g. adding blue splotches to the sky behind the cow, and polka dots inside the man's face). We find that matching μ only (Ours) produces images that utilize many features from the style image without overly distorting content.

3.4.2 Ablation: Optimal Transport

Other works [26, 36, 23] have used methods inspired by optimal transport (OT) to map content features to style features. This approach elegantly solves the problem of ensuring that the distribution of features in the output image matches the distribution of features in the style image. However, computing an OT plan (even approximately) is far more expensive than finding nearest neighbors. During HM we only computer feature matches once per scale, so it is expensive but feasible to replace nearest neighbors with the sinkhorn algorithm [6], a close approximation of OT. However, even using the approximate sinkhorn algorithm is impractical during FS, and would increase the runtime of our method to over an hour. While using OT during the HM phase often increases the output's visual diversity, the effect is not dramatic enough to justify the increased computational cost (See Figure 8).

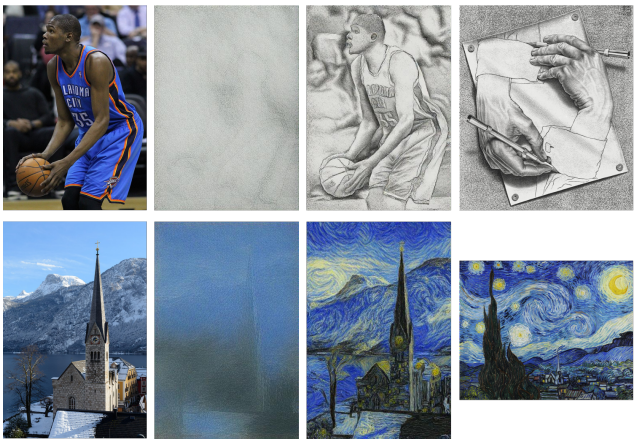
3.4.3 Ablation: Ground Metric

Both Nearest-Neighbors and Optimal Transport rely on choosing a good metric to compare features. Like other other style transfer works [13, 25, 23], we find that the cosine distance seems better suited to comparing the activations of pre-trained neural networks than the ℓ_2 distance. Using the cosine distance leads to dramatically better content preservation **and** stylization quality (See Figure 9).



Content | HM w/ OT ≈ 65 Secs. | Ours ≈ 35 Secs. | Style

Figure 8: Replacing nearest neighbors (NN) with OT during the HM stage is feasible, but significantly increases runtime and produces similar results. Replacing NN with OT during the FS stage is infeasible, and would increase runtime to over an hour (See Section 3.4.2)



Content | ℓ_2 | Cosine (Ours) | Style

Figure 9: Ablation demonstrating the importance of comparing neural activations using the cosine distance rather than ℓ_2.

4. Evaluation

4.1. User Study

Evaluating and comparing style transfer algorithms quantitatively is challenging. One reason for this is that we have yet to develop automatic metrics for robust content recognition, let alone judgment of stylistic similarity, that can truly compete with humans. Another challenge is that there is an inherent tension in style transfer between content preservation and stylization. Inspired by the evaluation proposed proposed in [30, 43, 23, 44], we conducted a human study using Amazon Mechanical Turk (AMT), evaluating content preservation and stylization quality separately.

As a measure of stylization quality, users are asked the question "Which of image A or image B better matches the style of the reference", where A and B are the outputs two different algorithms produced for the same content/style pair, and the reference is the input style image. Users are forced to choose between 'A', 'B'. For each algorithm we report a *style preference rate* between 0 and 1, which is the fraction of times an algorithm won such comparisons. For each algorithm this score includes comparisons with all other algorithms/hyperparameter combinations (but exclude comparisons to the same algorithm with different settings). This study is conducted over over 90 input pairs, each of which is shown to on average 3.9 AMT workers. This leads to the style preference rate being averaged over at least 3500 responses for each algorithm/hyperparameter combination.

In addition we can isolate head-to-head preference rates between our algorithm and others for default settings. Our output is judged to better match the style 72%, 67%, 69%, and 62% of the time relative to CNNMRF [23], Gatys [11], Reshuffle [13], and STROTSS[23] respectively. Each of these isolated head-to-head rates is based on slightly under 360 responses.

Content preservation is measured in absolute terms. AMT workers are shown an algorithm's output, along with the input content image, and asked to rate their agreement with "I perceive the same content in both images.", the options ranging from "Strongly Disagree" to "Strongly Agree" on a 7 point scale (4 being "Neutral"). We find that in general AMT workers tend to feel neutral about this statement for all algorithms tested. Each content preservation score is computed over 90 input pairs, shown to on average 4.7 unique AMT workers. This leads to a content preservation score averaged over slightly under 450 responses for each algorithm/hyperparameter combination.

We perform these studies using the set of 90 content/style pairs proposed in [23]; 30 pairs with semantically paired style and content, 30 with semantically different style and content, and 30 where the content is a face and the style is a texture. We compare our algorithm with leading optimization-based style transfer methods [25, 13, 12, 23]. For these methods we use the procedure described in [23], when possible obtaining results for each algorithm when the default hyperparameter controlling stylization level is doubled and halved. To obtain our high, default, and low stylization results we set α to be 0.75, 0.25, and 0.00 respectively. Qualitative examples from this study are shown in Figure 10, and quantitative results are summarized in Figure 11.

4.2. Limitations and Failure Cases

Qualitatively our method reliably produces images composed of small details from the style. However, the style features we can robustly capture tend to have small spatial extent. Values of $\alpha > 0.5$ can capture aspects of the style image with larger spatial extent. However, this often dis-

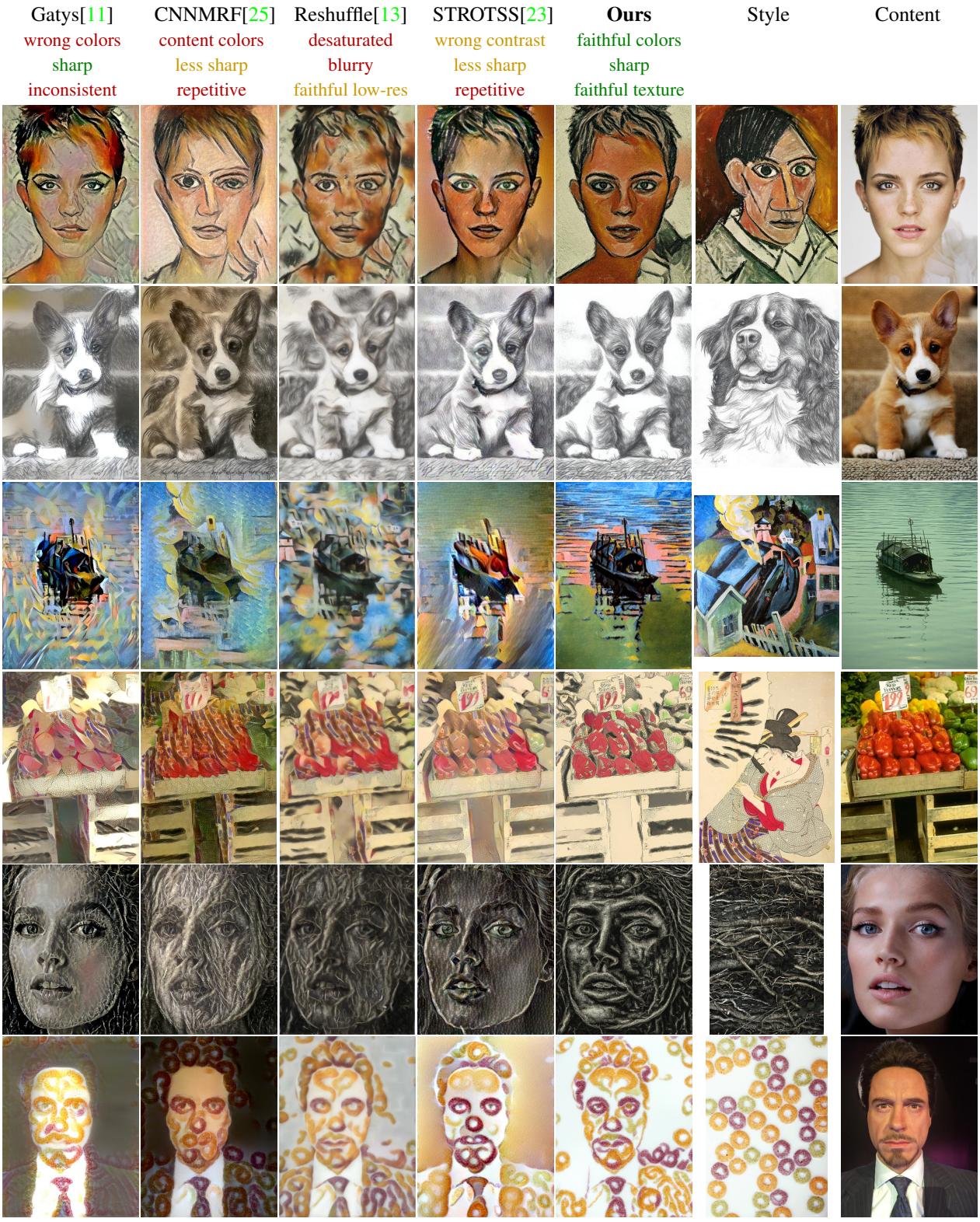


Figure 10: Qualitative comparison between outputs used in AMT study (default hyperparameters). Below each algorithm's column is the runtime when producing a 512×512 pixel output on our hardware. **We project the speed of Reshuffle [13] using the relative speed to Gatys reported in [23], as running Reshuffle required a different machine running Windows**

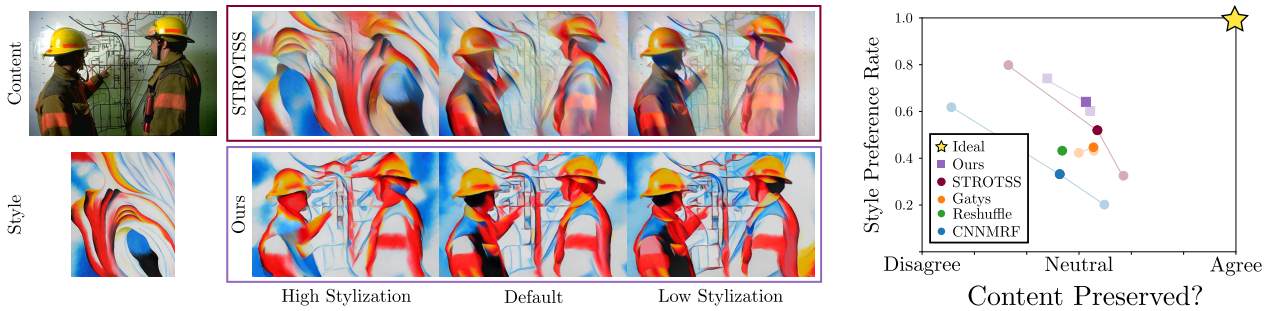


Figure 11: On the left we give an example of our method and STROTSS [23] trading off between stylization and content preservation. While STROTSS is able to achieve a wider range of visual effects, their high stylization setting often destroys content preservation by cloning large sections of the style image, and their low stylization setting fails to alter the high frequencies of the content image. If higher stylization levels are desired for our method, users can either increase α from 0.75 to 1.00 (Figure 5) or use FS only (Figure 6). On the right we plot the average ratings each algorithm (with varying hyperparameter settings when possible) achieved in our two AMT studies, one evaluating content preservation, the other fidelity to the style image. Fully saturated points indicate default hyperparameter settings for each method, and different settings of the same method are connected by lines

torts the content an unacceptable amount, and works best when the style is a homogeneous texture, or highly abstract (e.g. a cubist or abstract expressionist style).

A second limitation is that our method sometimes introduces a slight palette shift relative to the style image. This can manifest as outputs that are slightly over-saturated; or as slight hue shifts for drawn styles, where the color of the 'paper' and 'ink' will be a subtly different than in the style.

The main failure mode of our method is when jarring color shifts are introduced within a homogeneous region of the content during the FS regime. While our AMT study indicates this does not occur frequently enough to dramatically impact our content preservation relative to other methods, it is a problem when it occurs. This is an unfortunate side-effect of using no content loss. We can re-introduce a content loss into our framework easily, and in many cases this resolves the color-shifting problem. However this can also introduce content features to the output that are inconsistent with the style (See Figure 12).

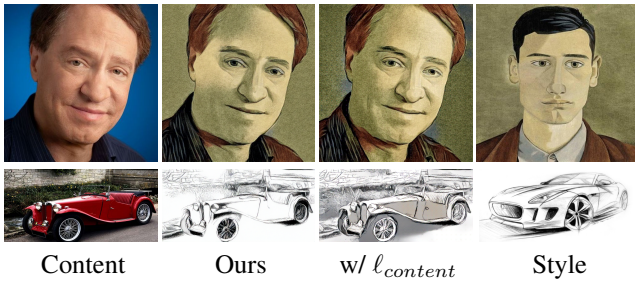


Figure 12: Because our method lacks a content loss ($\ell_{content}$), jarring color shifts occasionally occur within homogeneous objects. These can typically be corrected by adding a content loss to our optimization (top row), however adding a content loss can result distinctive aspects of the style being ignored, for example introducing shading to a black and white style (bottom row).

5. Conclusion

In this work we demonstrate that the layout and perceptual semantics of a content image can be recreated by rearranging features extracted from a very different style image, without the need for any additional 'content loss'. We propose Neural Neighbor Style Transfer (NNST), a straightforward procedure for appropriately rearranging the style features, and then using them to synthesize an aesthetically pleasing image that faithfully capture small details of the target style.

We show that nearest neighbors, despite its simplicity, is an effective tool for mapping from the features of one image to another. However we suspect significant improvements could be made by incorporating more sophisticated techniques that have been leveraged in other style transfer work. One example is optimal transport [26, 36, 23]. While it did not benefit our proposed method, it is a more principled tool for matching distributions of features than nearest neighbors. Another example is graphical models [2, 25], which have been previously used to model long range spatial dependencies between style features.

A further area for improvement is our algorithm's reliance on gradient descent, which dramatically limits its efficiency. Using our 'target feature tensor' as the input to a neural network, rather than as the optimization targets for gradient descent, seems like a viable and important path to explore. However, we leave this to future work.

References

[1] Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. Stylizing animation by example. In *TOG*, volume 32, pages 1–12. ACM New York, NY, USA, 2013. 2

[2] Guillaume Berger and Roland Memisevic. Incorporating long-range consistency in cnn-based texture generation. *arXiv preprint 1606.01286*, 2016. 1, 2, 8

[3] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint 1612.04337*, 2016. 1, 3

[4] Tai-Yin Chiu. Understanding generalized whitening and coloring transform for universal style transfer. In *ICCV*, pages 4452–4460, 2019. 3, 5

[5] Tai-Yin Chiu and Danna Gurari. Iterative feature transformation for fast and versatile universal style transfer. In *ECCV*, 2020. 3

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013. 5

[7] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346. ACM, 2001. 2

[8] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *CVPR*, volume 2, pages 1033–1038. IEEE, 1999. 2

[9] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sỳkora. Stylit: illumination-guided example-based stylization of 3d renderings. In *TOG*, volume 35, pages 1–11. ACM New York, NY, USA, 2016. 2

[10] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. StyLit: Illumination-guided example-based stylization of 3D renderings. *ACM Trans. Gr.*, 35(4):92, 2016. 1, 2

[11] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. In *CoRR*, 2015. 6, 7

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 1, 2, 3, 6

[13] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *CVPR*, pages 8222–8231, 2018. 1, 2, 3, 5, 6, 7

[14] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015. 2, 3

[15] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *SIGGRAPH*, pages 327–340. ACM, 2001. 1, 2

[16] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *ICCV*, pages 1510–1519, 2017. 3

[17] Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. LazyFluids: Appearance transfer for fluid animations. In *ACM Trans. Gr.*, volume 34, page 92, 2015. 1, 2

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016. 1

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 3

[20] Andrej Junginger, Markus Hanselmann, Thilo Strauss, Sebastian Boblest, Jens Buchner, and Holger Ulmer. Unpaired high-resolution and scalable style transfer using generative adversarial networks. *arXiv preprint 1810.05724*, 2018. 3

[21] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. In *CGF*, volume 34, pages 349–359. Wiley Online Library, 2015. 2

[22] Diederik P Kingma and J Adam Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 4

[23] Nicholas Kolkin, Jason Salavon, and Greg Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, pages 10051–10060, 2019. 1, 2, 3, 5, 6, 7, 8

[24] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *ICCV*, pages 4422–4431, 2019. 3

[25] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, pages 2479–2486, 2016. 1, 2, 3, 5, 6, 7, 8

[26] Pan Li, Lei Zhao, Duanqing Xu, and Dongming Lu. Optimal transport of deep feature for image style transfer. In *the 2019 4th International Conference on Multimedia Systems and Signal Processing*, pages 167–171, 2019. 3, 5, 8

[27] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NIPS*, pages 385–395, 2017. 3

[28] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *IJCAI*, 2017. 2

[29] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *SIGGRAPH*, 2017. 1, 3

[30] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *CVPR*, 2017. 2, 4, 6

[31] Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Brushables: Example-based edge-aware directional texture painting. In *CGF*, volume 34, pages 257–267. Wiley Online Library, 2015. 2

[32] Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sỳkora. Painting by feature: texture boundaries for example-based image creation. In *TOG*, volume 32, pages 1–8. ACM New York, NY, USA, 2013. 2

[33] Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sýkora. Painting by feature: Texture boundaries for example-based image creation. *ACM Transaction on Graphics*, 32(4):116, 2013. 1

[34] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, pages 768–783, 2018. 1, 2

[35] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, pages 3376–3385, 2015. 2, 3

[36] Youssef Mroueh. Wasserstein style transfer. *arXiv preprint 1905.12828*, 2019. 3, 5, 8

[37] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint 1701.08893*, 2017. 1, 2

[38] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *ECCV*, pages 698–714, 2018. 3

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014. 2, 4

[40] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *CVPR*, pages 13816–13825, 2020. 3

[41] Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Arbitrary style transfer using neurally-guided patch-based synthesis. *CAG*, 2020. 1, 3

[42] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488, 2000. 2

[43] Mao-Chuang Yeh, Shuai Tang, Anand Bhattad, and David A Forsyth. Quantitative evaluation of style transfer. *arXiv preprint 1804.00118*, 2018. 6

[44] Mao-Chuang Yeh, Shuai Tang, Anand Bhattad, Chuhang Zou, and David Forsyth. Improving style transfer with calibrated metrics. In *WACV*, pages 3160–3168, 2020. 6

[45] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via graph cuts. In *ICCV*, pages 5943–5951, 2019. 3

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, pages 2242–2251, 2017. 3