Combining Multiple Styles using Neural Style Transfer

Master Thesis Colloquium

Felix Munzlinger

May 12, 2022

Outline

Figure: "Annunciation"(1475) by Leonardo da Vinci und Andrea del Verrocchio (left) and Fra Angelico (1430)(right)

Key questions

Is it technically possible to intersect artistic styles using neural style transfer?

To what extent do style intersections create a new artistic style?

Neural style transfer (NST)

- introduced by Gatys et al. in 2016 [2]
- task: transfer texture of one image to another
- transformation can be performed by AdaIN [7, 8]

Generative Adversarial Networks (GAN)

- introduced by Goodfellow et al. in 2014 [5]
- two competing neural networks based on *Game Theory* [14]
- generator $\mathcal{G}$: generates fake samples
- discriminator $\mathcal{D}$: distinguishes real and fake samples

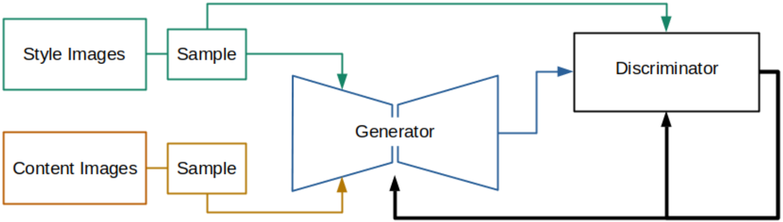

Figure: Interaction between generator and discriminator

COCO-FUNIT architecture

- COCO-FUNIT architecture by Saito et al. (2020) [15] based on Liu et al. (2019) [11]
- $\mathcal{G}$: few-shot image generator with content conditioned style encoder
- $\mathcal{D}$: patch GAN discriminator

Generator $\mathcal{G}$ [1/2]

- E_c encodes content image
- E_s encodes style image
- D conditions the representation of E_s with E_c to determine AdaIN parameters
- D applies AdaIN parameters in a normalization layer

Generator $\mathcal{G}$ [2/2]

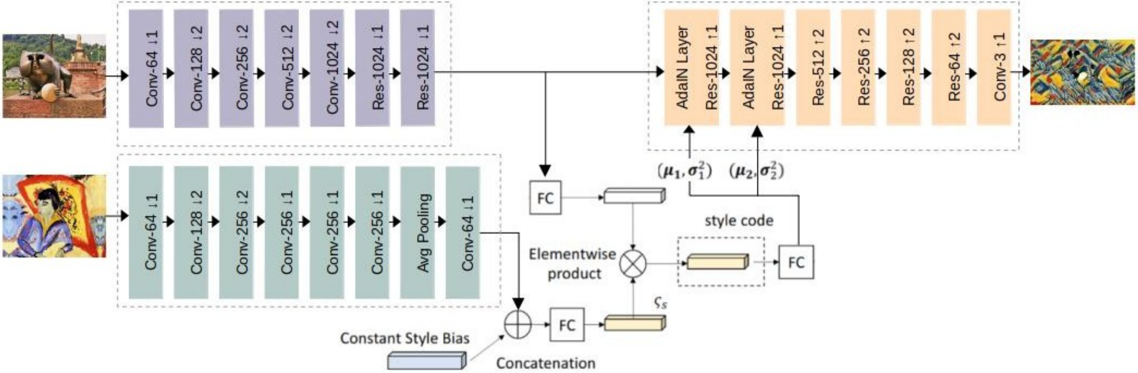

Figure: Architecture of $\mathcal{G}$. Modified from Saito et al. [15] and Liu et al. [11]

Discriminator $\mathcal{D}$

- $\mathcal{D}_m$ encodes samples
- $\mathcal{D}_c$ classifies the encoded sample by a data class

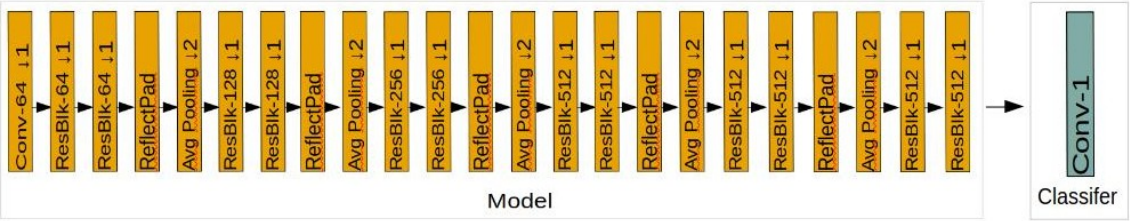

Figure: Architecture of $\mathcal{D}$

Loss function

- $\mathcal{G}$ and $\mathcal{D}$ are trained separately [4]
- hinge GAN loss [10, 18, 15]
- feature matching loss [16, 15]
- reconstruction loss [15]
- no classification of reconstructed images

Training approaches

1. single style sample: $(y_1, l_1) \sim \mathcal{S}$ [15]
2. two style samples from the same artists: $(y_1, l_1), (y_2, l_2) \sim \mathcal{S}$, with $l_1 = l_2$ [15]
3. two style samples from the different artists: $(y_1, l_1), (y_2, l_2) \sim \mathcal{S}$, with $l_1 \neq l_2$
4. randomly use mode 1-3

Experiments

- training instabilities occurred using (hyper-)parameters by Saito et al. [15]
- determine stability measures in a separate optimization problem:
  - spectral normalization in $\mathcal{G}$ [12]
  - TTUR-learning rate [6, 18]
  - differentiable augmentations [19]
- best stylization results when randomly sampling the training mode

Results

Figure: Style intersection between Berthe Morisot and Pablo Picasso (top) vs. style intersection between two paintings by Pablo Picasso (bottom).

t-distributed stochastic neighbor embedding (t-SNE) [1/3]

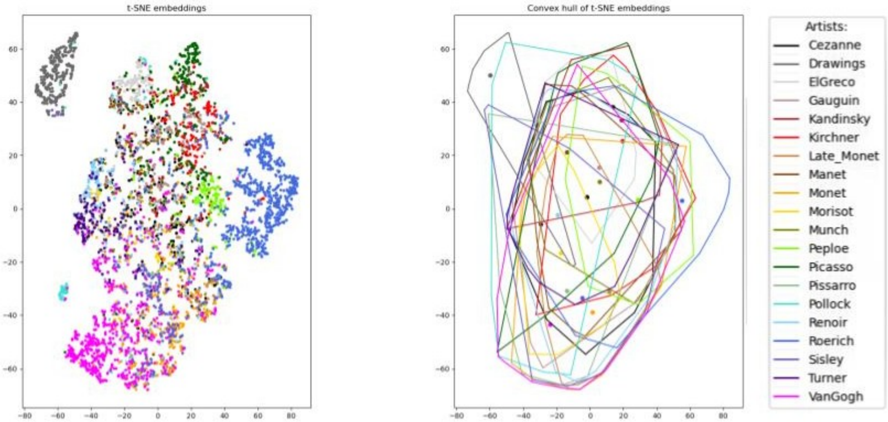

Figure: t-SNE embedding for model D (left) vs. convex hull of t-SNE embedding with centroids (right)

t-distributed stochastic neighbor embedding (t-SNE) [2/3]

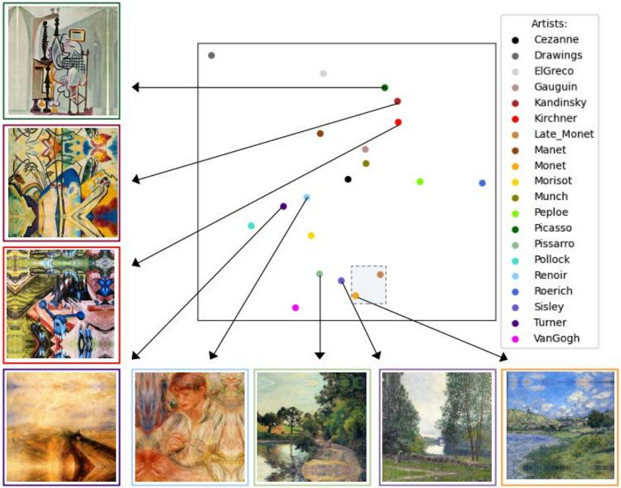

Figure: Comparison of style and centroids in t-SNE embedding for model D

t-distributed stochastic neighbor embedding (t-SNE) [3/3]

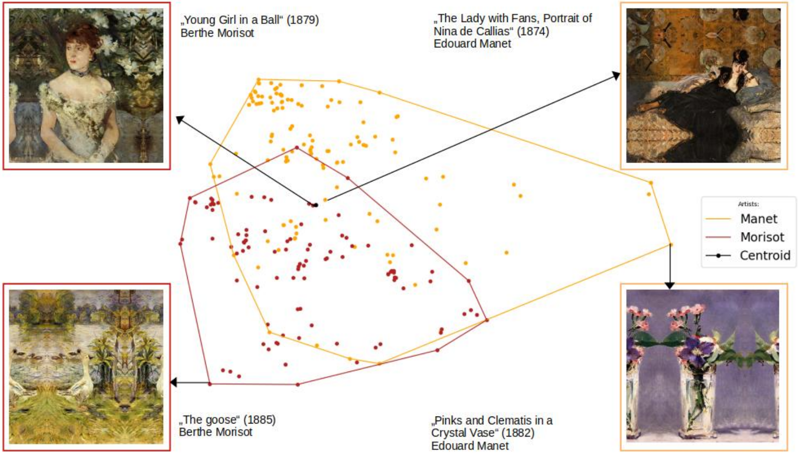

Figure: Comparing the style embeddings in t-SNE of Berthe Morisot (firebrick red) and Edouard Manet (orange)

Comparison to other approaches

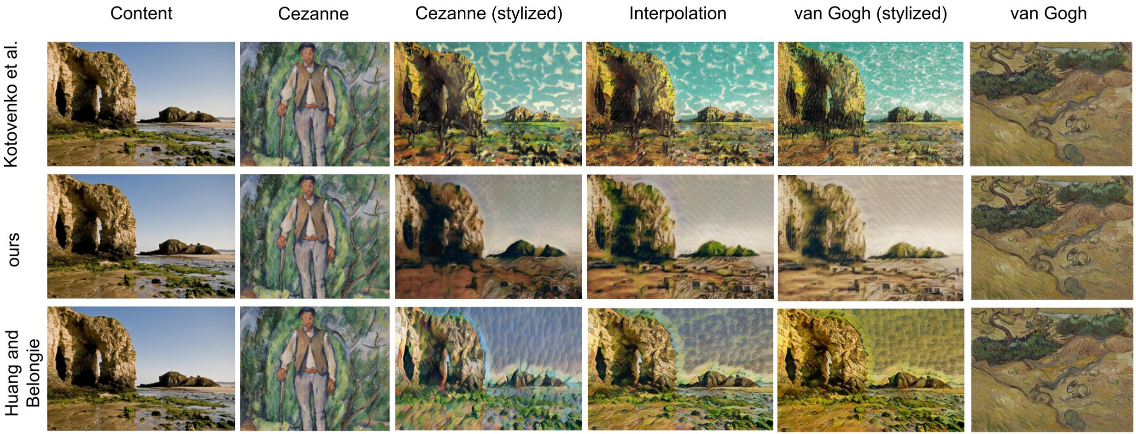

Figure: Comparing our approach to Kotovenko et al. [9] Huang and Belongie [7]. Adopted from Kotovenko et al. [9].

Improvements to this approach

- add noise by a linearly function to $\mathcal{D}$ [3]
- random game participants
- balance style dataset
- combination of styles from the same epoch

Future research with neural style transfer

- 3D style transfer using NeRF by Müller et al. [13]
- sim2real: scene adjustment for scene recognition [17]

Reflections on the nature of artistic styles [1/4]

- Fernie defines style as "[...] an expression of a collective spirit". [1]
- style includes what is troubling the minds of a society
- style also includes an iconography by which we attribute meaning
- machine learning cannot (yet) represent the iconographic level of styles

Reflections on the nature of artistic styles [2/4]

- interpreting art is difficult because of our cultural bias
- machines are less biased in this regard
- similarities can provide valuable insights

Reflections on the nature of artistic styles [3/4]

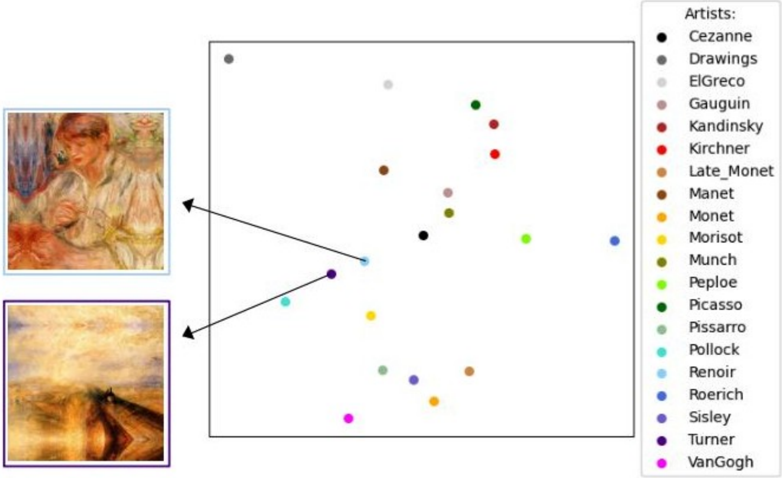

Figure: Style embedding showed a similarity between William Turner and Pierre-Auguste Renoir

Reflections on the nature of artistic styles [4/4]

- machine learning can intersect styles to the same extent that it captures the complexity of style
- visually observable that they created something new.

Stylization samples [1/5]

Figure: Model D: Image stylized in the style of Pablo Picasso.

Stylization samples [2/5]

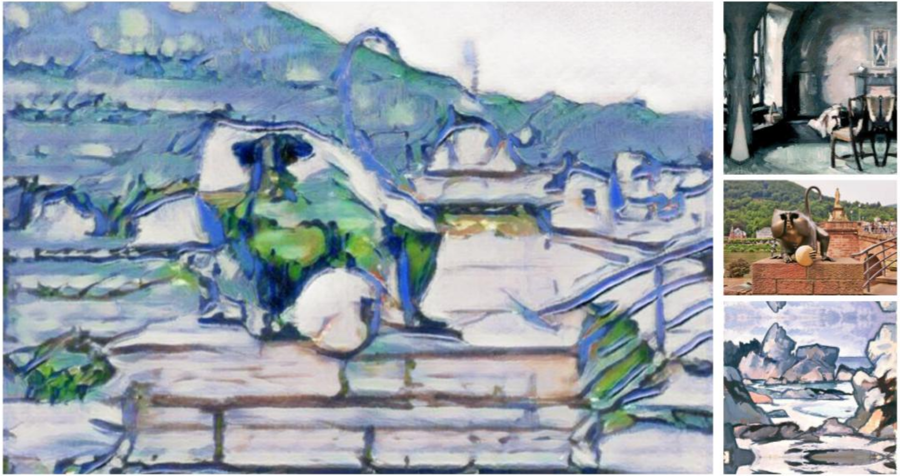

Figure: Model D: Image stylized in the style of Samuel Peploe.

Stylization samples [3/5]

Figure: Model D: Image stylized in the style of Pablo Picasso and Édouard Manet.

Stylization samples [4/5]

Figure: Model D: Image stylized in the style of Jackson Pollock and Ernst Ludwig Kirchner.

Stylization samples [5/5]

Figure: Model D: Image stylized in the style of Camille Pissarro and Pablo Picasso.

[1] E. Fernie et al. *Art History and Its Methods.* Phaidon Press, 1995. ISBN: 9780714829913. URL: https://books.google.de/books?id=rZkYAQAAIAAJ.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265.

[3] Ian Goodfellow. *NIPS 2016 Tutorial: Generative Adversarial Networks.* 2017. arXiv: 1701.00160 [cs.LG].

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[5] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27.* Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[6] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* 2018. arXiv: 1706.08500 [cs.LG].

[7] Xun Huang and Serge J. Belongie. "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: *CoRR* abs/1703.06868 (2017). arXiv: 1703.06868. URL: http://arxiv.org/abs/1703.06868.

[8] Xun Huang et al. *Multimodal Unsupervised Image-to-Image Translation.* 2018. arXiv: 1804.04732 [cs.CV].

[9] Dmytro Kotovenko et al. "Content and Style Disentanglement for Artistic Style Transfer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* Oct. 2019.

[10] Jae Hyun Lim and Jong Chul Ye. *Geometric GAN.* 2017. arXiv: 1705.02894 [stat.ML].

[11] Ming-Yu Liu et al. "Few-Shot Unsupervised Image-to-Image Translation". In: *IEEE International Conference on Computer Vision (ICCV).* 2019.

[12] Takeru Miyato et al. *Spectral Normalization for Generative Adversarial Networks.* 2018. arXiv: 1802.05957 [cs.LG].

[13] Thomas Müller et al. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *arXiv:2201.05989* (Jan. 2022).

[14] John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition).* Princeton University Press, 1944. ISBN: 9780691130613. URL: http://www.jstor.org/stable/j.ctt1r2gkx.

[15] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. "COCO-FUNIT: Few-Shot Unsupervised Image Translation with a Content Conditioned Style Encoder". In: *arXiv preprint arXiv:2007.07431* (2020).

[16] Tim Salimans et al. *Improved Techniques for Training GANs.* 2016. arXiv: 1606.03498 [cs.LG].

[17] Matthew Tancik et al. “Block-NeRF: Scalable Large Scene Neural View Synthesis”. In: *arXiv* (2022).

[18] Han Zhang et al. *Self-Attention Generative Adversarial Networks*. 2019. arXiv: 1805.08318 [stat.ML].

[19] Shengyu Zhao et al. *Differentiable Augmentation for Data-Efficient GAN Training*. 2020. arXiv: 2006.10738 [cs.CV].

Appendix

Loss function

- $\mathcal{G}$ and $\mathcal{D}$ are trained separately
- sample from a content dataset $\mathcal{C}$ and style dataset $\mathcal{S}$
- play a *minimax* game with loss function $\mathcal{L}(\mathcal{G}, \mathcal{D})$:

$$\mathcal{L}(\mathcal{G}, \mathcal{D}) = \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{\mathcal{G}}(\mathcal{G}, \mathcal{D}) + \mathcal{L}_{\mathcal{D}}(\mathcal{G}, \mathcal{D}) \tag{1}$$

Loss function $\mathcal{G}$

- $\mathcal{L}^{\mathcal{G}}_{GAN} = \underset{\substack{(y,l)\in\mathcal{S}\\ x\in\mathcal{C}}}{\mathbb{E}}[-log\ \mathcal{D}(\mathcal{G}(x,y),l)]$ [3]
- $\mathcal{L}_F(\mathcal{G}) = \underset{\substack{(y,l)\in\mathcal{S}\\ x\in\mathcal{C}}}{\mathbb{E}}\ [||\mathcal{D}_m(y) - \mathcal{D}_m(\mathcal{G}(x,y)||_1]$ [16]
- $\mathcal{L}_R(\mathcal{G}) = \underset{x\in\mathcal{C}}{\mathbb{E}}\ [||x - \mathcal{G}(x,x)||_1]$ [11]

- $\mathcal{G}$ optimizes these losses with respective weights in λ:

$$\mathcal{L}_{\mathcal{G}}(\mathcal{G},\mathcal{D}) = \lambda_{GAN} * \mathcal{L}^{\mathcal{G}}_{GAN}(\mathcal{G},\mathcal{D}) + \lambda_R * \mathcal{L}_R(\mathcal{G}) + \lambda_F * \mathcal{L}_F(\mathcal{G}) \quad (2)$$

Loss function $\mathcal{D}$

- $\mathcal{L}^{\mathcal{D}}_{GAN}(\mathcal{G}, \mathcal{D}) = \underset{(y,l)\in\mathcal{S}}{\mathbb{E}} \left[-\min(0,\ \mathcal{D}(y,l) - 1)\right] + \underset{\substack{(y,l)\in\mathcal{S}\\ x\in\mathcal{C}}}{\mathbb{E}} \left[-\min(0, -\mathcal{D}(\mathcal{G}(x,y),l) - 1)\right]$

- $\mathcal{D}$ optimizes this loss with a respective weight in λ:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{G}, \mathcal{D}) = \lambda_{GAN} * \mathcal{L}^{\mathcal{D}}_{GAN}(\mathcal{G}, \mathcal{D}) \tag{3}$$

Generative model

- generates samples in image space
- attempts to sample from $\mathcal{D}$'s probability distribution of real data
- maximizes the probability of $\mathcal{D}$ making a mistake
- at convergence: samples are indistinguishable from real data

Discriminative model

- determines a scalar value for each sample
- distinguishes samples drawn from real data and $\mathcal{G}$
- maximize the probability of assigning the correct label to its input
- at convergence: same scalar value for all samples