Style transfer with GANs

Clément Acher
Hugo Cisneros

Advisor: Thomas Eboli
Overseen by: Josef Sivic

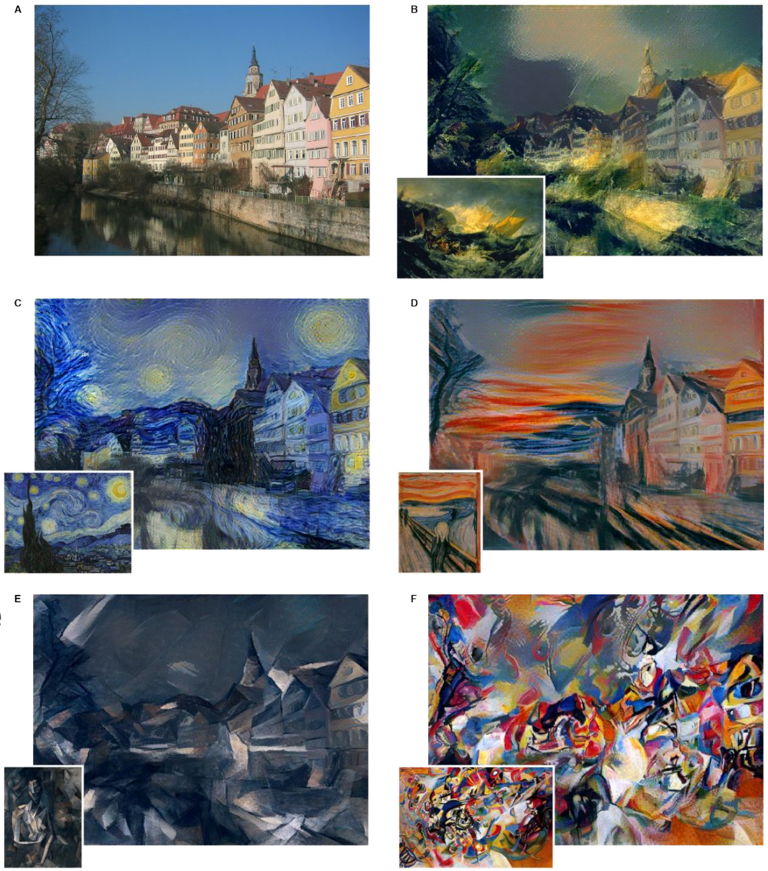
January 15th, 2019

Style transfer

Transfer the visual characteristic from a source image to a target content image.

Preserve main features of the content image.

- Tool for artistic creation
- Human perception of art and beauty
- A way of understanding how CNNs capture visual features of natural images



Gatys, L. A., Ecker, A. S. & Bethge, M. Image Style Transfer Using Convolutional Neural Networks. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2414–2423 (IEEE, 2016). doi:10.1109/CVPR.2016.265

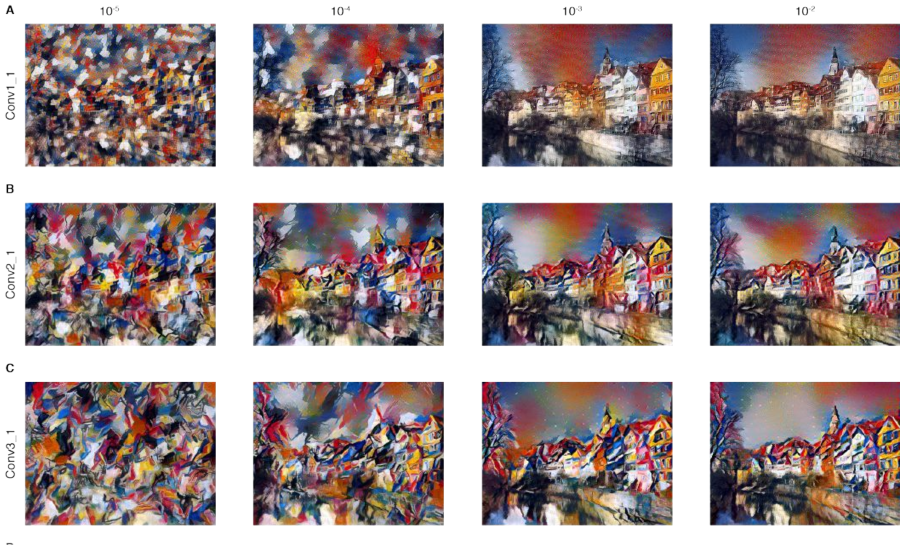
Challenges

- **What is a good style transfer?**
  - Most papers evaluate their proposed approach with side-by-side <u>subjective</u> visual comparaisons

  - No benchmark dataset
  - Quantitative evaluation : stylisation speed, training time, loss comparaison

- **Texture is different from style**

- **Balance between style and content**



Work done

Try out GANs on difficult style transfer tasks and compare the results to Neural Style Transfer methods

Datasets :

Pointillism paintings

Cubism paintings

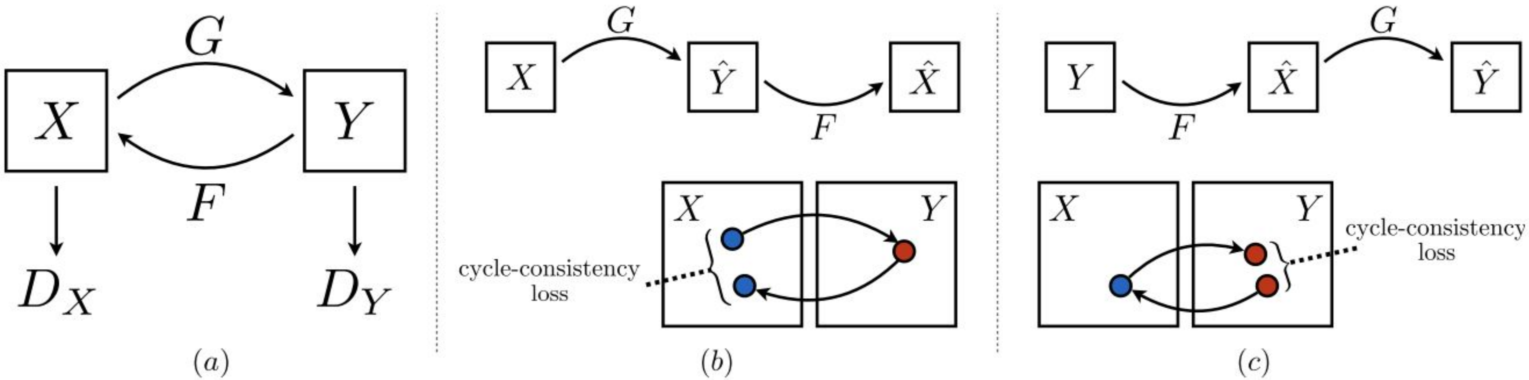
Anime movies

Flickr landscapes

Models used

- A Neural Algorithm of Artistic style by Gatys et al.
    - Composite content and style loss from internal features of a CNN
- CNNMRF by Li and Wand
    - Reconstruct the content image by using patches of features of the style image extracted with a CNN
- CycleGAN Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv:1703.10593 [cs]* (2017).
- CartoonGAN
    - A GAN architecture for mapping images to cartoons
    - Edge promoting loss to force the generator to create sharp edges

CycleGANs for style transfer



Can transfer style by learning a mapping between the two distributions (content and style)

Pros
- Style from a distribution (not single style image) → Captures a “global” style
- Reverse transformation

Cons
- Need for two corresponding datasets with a (likely) shared latent space
- Doesn’t work for a specific target style

Pointillism dataset

Style transfer is mostly about texture transfer.

This seems like an “easy” task - how do GANs perform?

+

Gatys et al.

CNNMRF

Pointillism dataset - CycleGAN

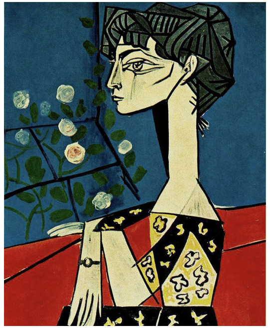
Photography to Pointillism

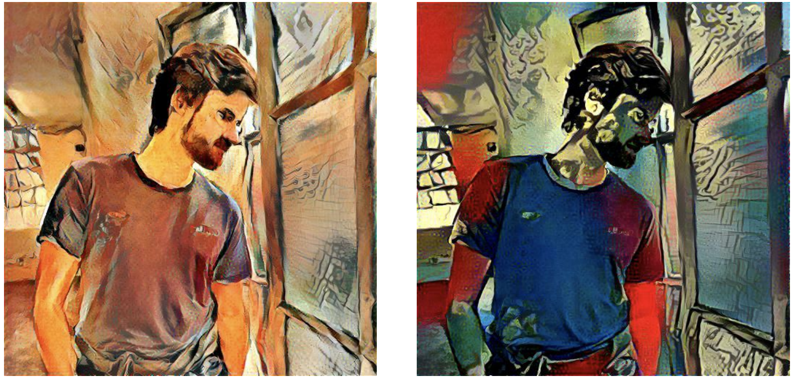
Cubism dataset

Interesting as :

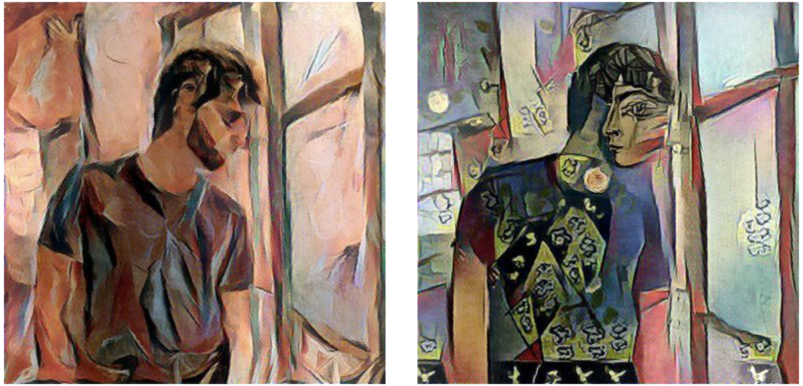
- Style is not only about texture
- Shapes are also modified

However, it was too tough to make a big enough and consistent dataset to train a GAN.





Gatys et al.

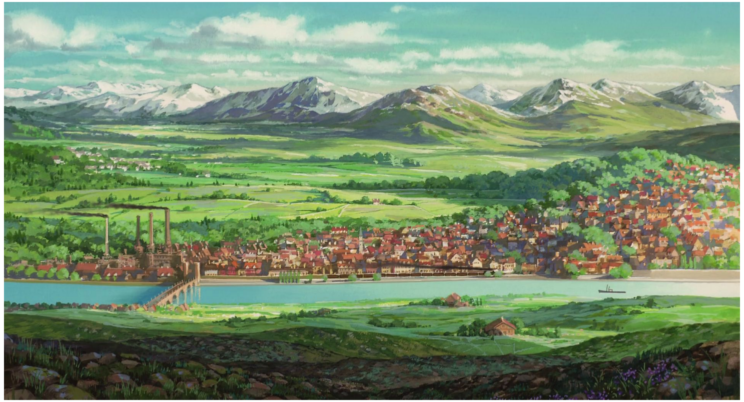


CNNMRF

Anime dataset

- Texture and color are still important
- Highly simplified and abstracted images from real world
- The style is very consistent across a movie - GANs seem more suited to this task

<u>Movies used :</u> “Spirited away” + “Howl’s moving castle”



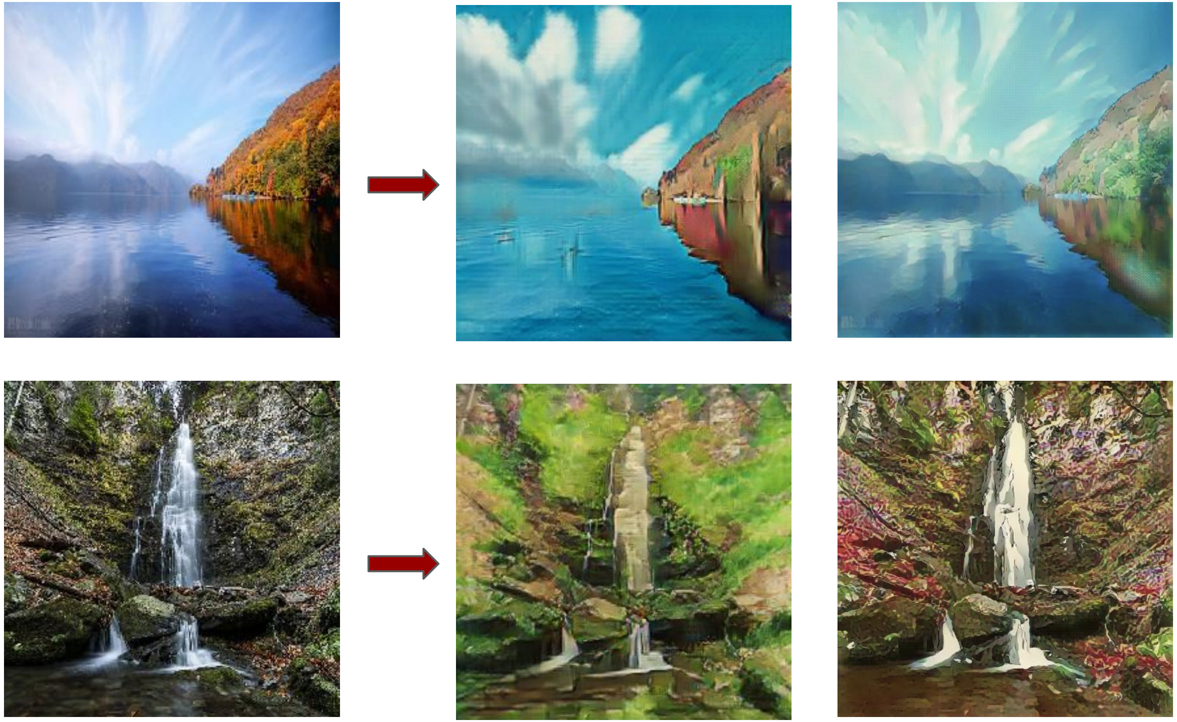
Anime dataset - CycleGAN and CartoonGAN

- Semantic characteristics are captured (grass, sky, water)
- Essential features of animated movies can be observed (sharp edges)

Limitations

- Results not consistent across the whole dataset
- Training time, computational cost & stability

Content | CycleGAN | CartoonGAN (pretrained - Hayao)



Thank you for your attention

Results

Pointillism to Photography

Cartoon dataset

Anime (“Spirited away” + “Howl’s moving castle”) ↔ Photography

CartoonGAN - Supplementary

Adversarial (edge promoting) loss:

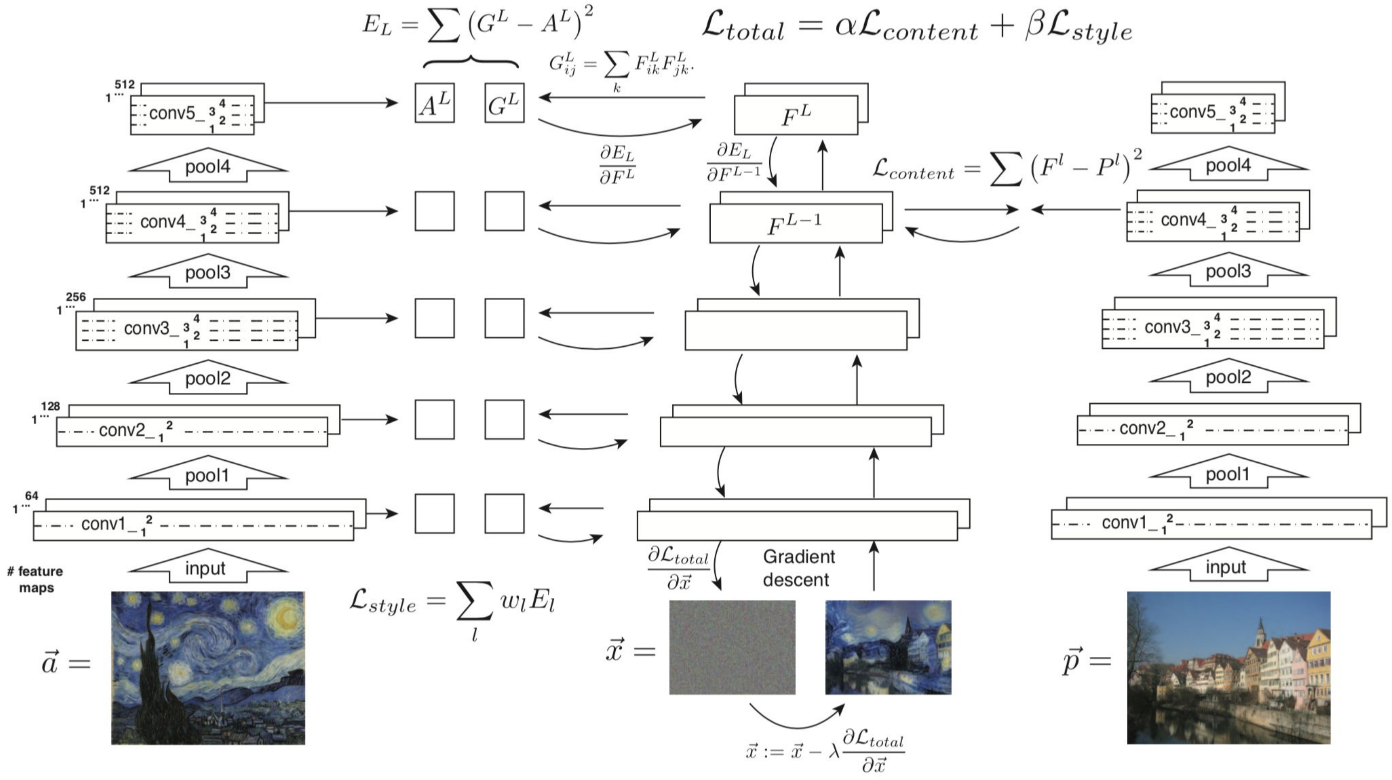
$$\mathcal{L}_{adv}(G, D) = \mathbb{E}_{c_i \sim S_{data}(c)}[\log D(c_i)] + \mathbb{E}_{e_j \sim S_{data}(e)}[\log(1 - D(e_j))] + \mathbb{E}_{p_k \sim S_{data}(p)}[\log(1 - D(G(p_k)))]$$

c are cartoon images, e are cartoon images with blurred edges,

p are content images.

Complete Loss: $\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D) + \omega \mathcal{L}_{con}(G, D)$

Gatys et al. - Supplementary



CNNMRF - Supplementary

Loss:

$$
\begin{aligned}
\mathbf{x} \quad = \quad & \arg\min_{x} E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) + \\
& \alpha_1 E_c(\Phi(\mathbf{x}), \Phi(\mathbf{x}_c)) + \alpha_2 \Upsilon(\mathbf{x})
\end{aligned}
$$

Patch reconstruction

$$
E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2
$$