Automated Deep Photo Style Transfer

Nathan Lichtlé
Alain Riou

École Normale Supérieure de Cachan

Image style transfer

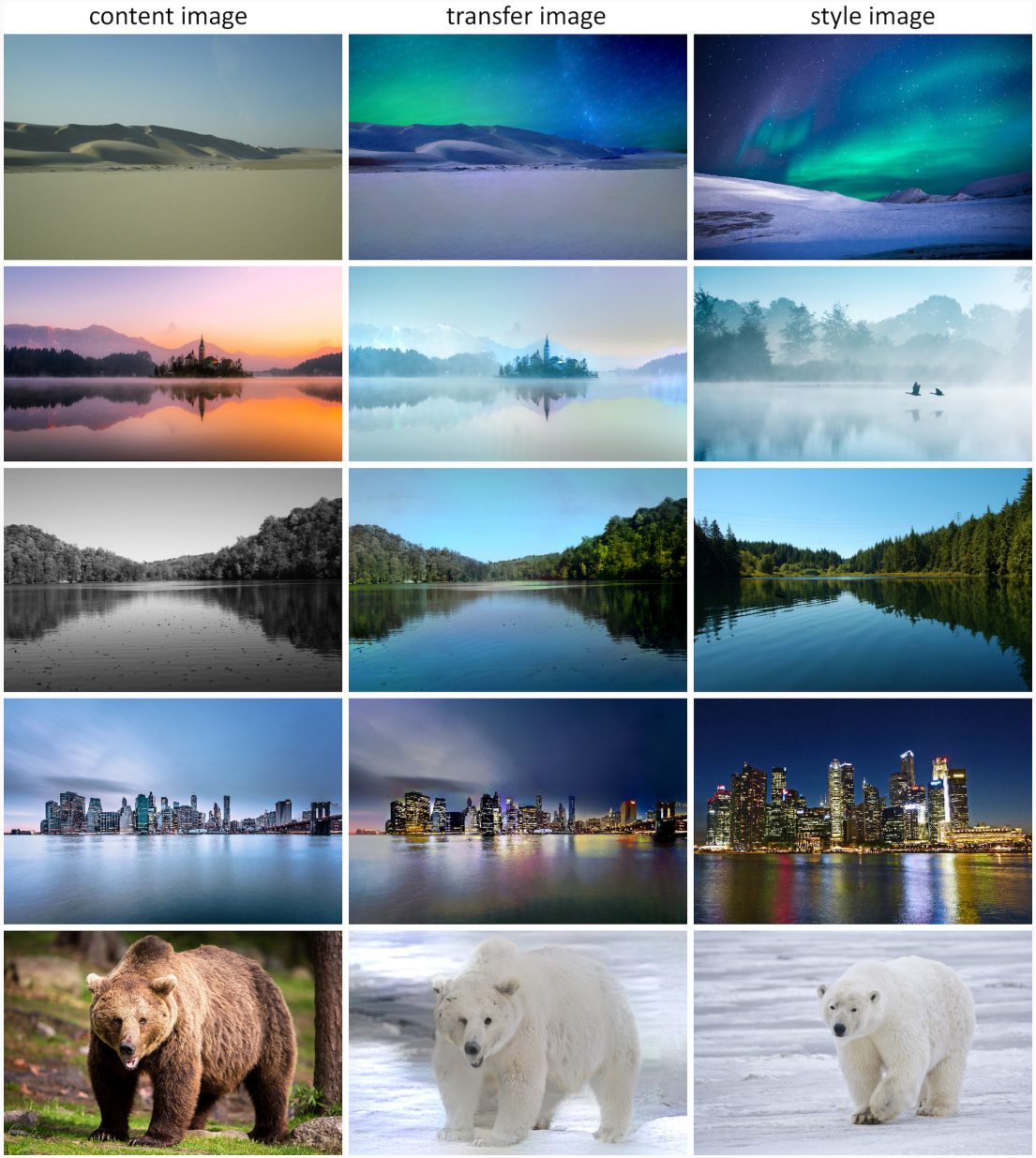


Image credits: https://github.com/Spenhouet/automated-deep-photo-style-transfer

Image style transfer

Painting: *La Muse* (Pablo Picasso)

Image style transfer using CNNs

$$\mathcal{L}_{\text{total}} = \underbrace{\sum_i \|F_i^\ell - P_i^\ell\|^2}_{\mathcal{L}_{\text{content}}} + \underbrace{\sum_\ell \sum_i (F_i^\ell \cdot F_j^\ell - A_i^\ell \cdot A_j^\ell)^2}_{\mathcal{L}_{\text{style}}}$$

where F_i^ℓ (resp. P_i^ℓ, resp. A_i^l) is the activations of the i^{th} filter in layer ℓ of the CNN when the input image is the transfer image (resp. the content image, resp. the style image).

Using this loss, perform gradient descent on the transfer image initialized as white noise.

Image Style Transfer Using Convolutional Neural Networks, Gatys *et al.*

How to enforce photorealism?

- **Idea:** enforce locally affine color changes on the image so that contours and details are preserved.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{content}} + \mathcal{L}_{\text{style}} + \mathcal{L}_m$$

- **In practice:** quadratic regularization term

$$J(\alpha) = \alpha^\top L \alpha$$

where L is a symmetric definite positive matrix.

Deep photo style transfer, Luan *et al.*
A closed-form solution to natural image matting, Levin *et al.*

Efficient computation of L

Time and space constraints: L dimension is $N \times N$

- Impossible to load in memory for huge images
- Very slow to compute

Fast matting using large kernel matting Laplacian matrices, He *et al.*

Efficient computation of L

Time and space constraints: L dimension is $N \times N$

- Impossible to load in memory for huge images
- Very slow to compute

Solution: Do not compute L explicitly, but define a linear operator $\widetilde{L}$ such that $\widetilde{L}(p) = Lp$, $\forall p \in \mathbb{R}^N$.

Fast matting using large kernel matting Laplacian matrices, He *et al.*

Efficient computation of L

Time and space constraints: L dimension is $N \times N$

- Impossible to load in memory for huge images
- Very slow to compute

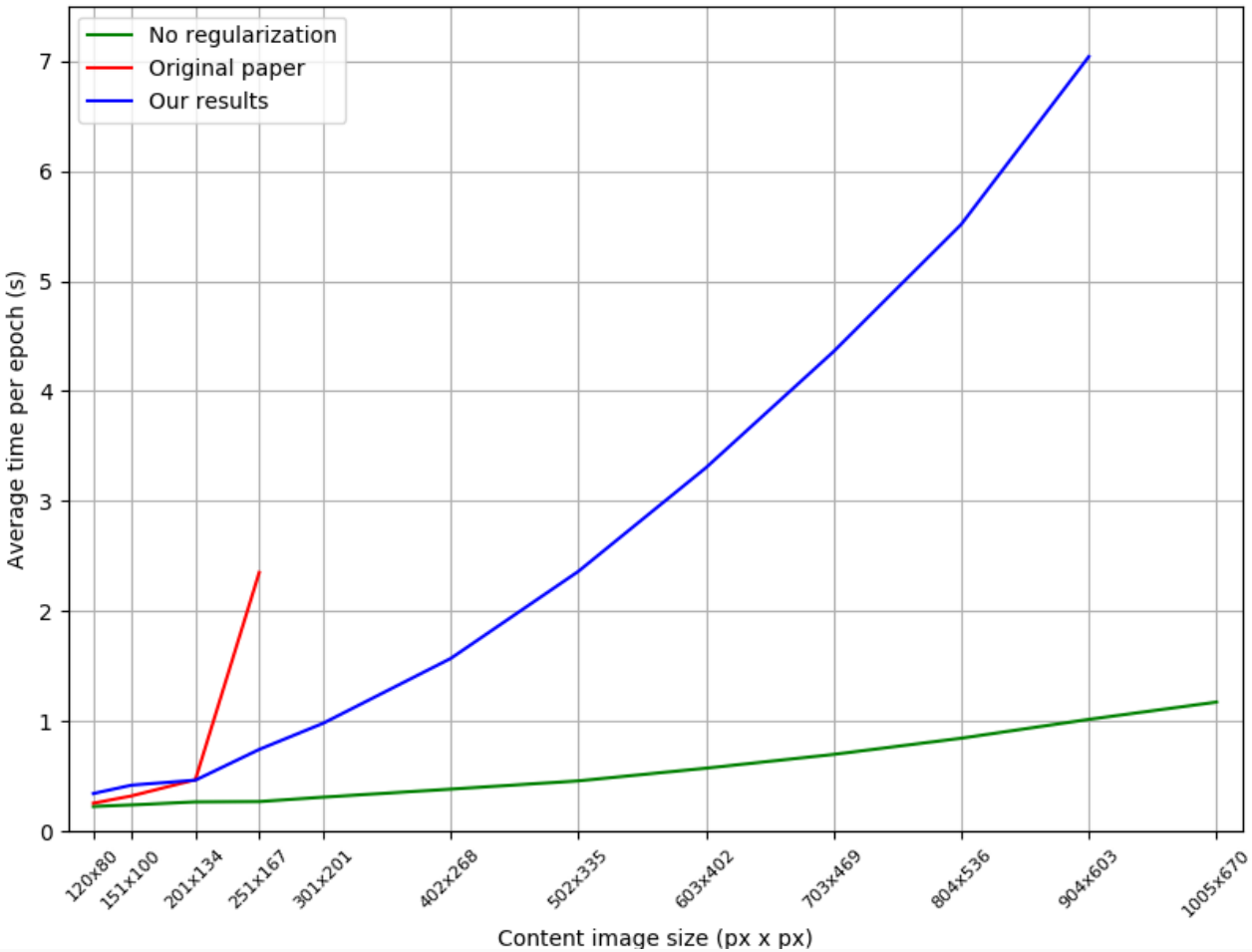
Solution: Do not compute L explicitly, but define a linear operator $\widetilde{L}$ such that $\widetilde{L}(p) = Lp$, $\forall p \in \mathbb{R}^N$.

Contribution: Implement this operator in TensorFlow

- Fast and parallel computation of Lp
- Backpropagation through this operator

Fast matting using large kernel matting Laplacian matrices, He *et al.*

Speed results



This results have been obtained with a NVIDIA GeForce GTX 1050 Ti GPU.

Real time style transfer

Another idea is to train a CNN to do style transfer for a specific style image, using the same loss $\mathcal{L}_{\text{content}} + \mathcal{L}_{\text{style}}$ as before.

Training takes around 8 hours on a GTX 1050 Ti. Once trained, style transfer can be efficiently be computed in one forward pass of the CNN, which is around 1000 times faster than the method presented before.

Objective: improve this method by adding photorealism constraints to the loss

- Implementation works with CNNs pre-trained on paintings images
- However we face issues when training it with new photorealistic style images, still working on debugging it

---

Perceptual Losses for Real-Time Style Transfer and Super-Resolution, Johnson *et al.*

Questions?