Image Style Transfer using Convolutional Neural Networks

Clément Chadebec

ENS - MVA

January 20, 2020

Overview

1. Presentation of the method

2. Experiments
   - Trade-off Content-Style
   - Choice of layer in CNN
   - Initialisation of Gradient Descent
   - Stochastic results

The method

Goal: Keep the content of an image $\vec{p}$ while applying the style of an image $\vec{a}$ to it

Figure: Content image $\vec{p}$

Figure: Style image $\vec{a}$

Architecture of the Method

- Use of a pre-trained Convolutional Neural Network VGG19

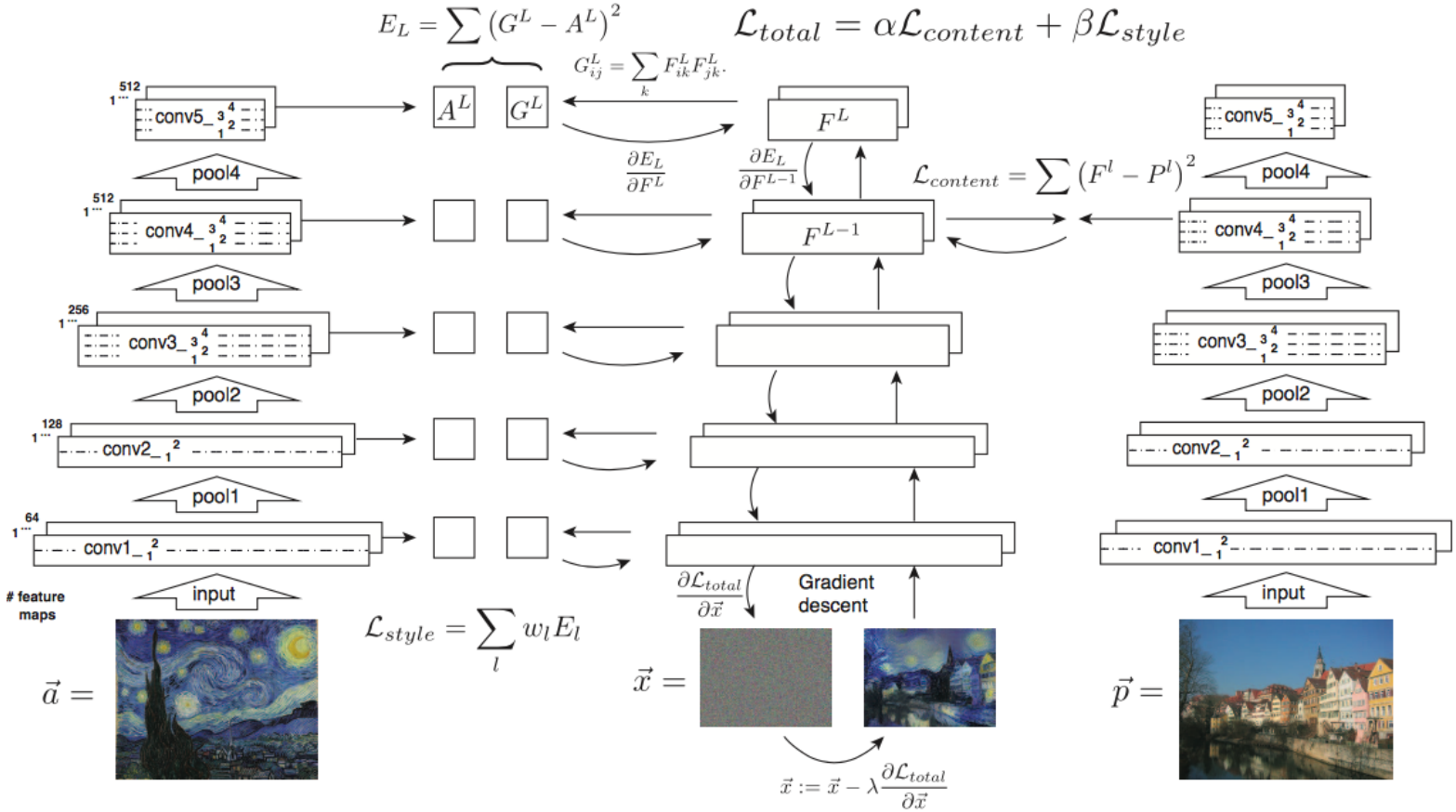

Figure: Architecture of the method

$\Longrightarrow$ need for a representation of the images

Content Representation

- Let P^l be the feature map of the content image $\overrightarrow{p}$ produced by VGG19 at the output of layer l
- Let F^l be the feature map of the image $\overrightarrow{x}$ produced by VGG19 at the output of layer l

$$\mathcal{L}_{content}(\overrightarrow{p}, \overrightarrow{x}, l) = \frac{1}{2} \sum_{i,j} (F^l_{i,j} - P^l_{i,j})^2$$

- The computation of the gradient gives

$$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{i,j}} = \begin{cases} (F^l - P^l)_{i,j} & \text{if} \quad F^l_{i,j} > 0 \\ 0 & \text{if} \quad F^l_{i,j} < 0 \end{cases}$$

where $F^l_{i,j}$ denotes the coefficient of the i-th filter (or feature) of the output of layer l at position j

Style Representation

- Several layers are considered
- Let E_l be the contribution of layer l in the loss
- Let w_l be a weight associated to the loss E_l in the total style loss
- We introduce the Gram matrix to compute E_l

$$G^l_{i,j} = \sum_k F^l_{i,k} F^l_{j,k}$$

and we obtain

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{i,j} - A^l_{i,j})^2$$

and finally we have the gradient

$$\frac{\partial \mathcal{L}_{style}}{\partial F^l_{i,j}} = \begin{cases} w_l \frac{1}{N_l^2 M_l^2} \left((F^l)^\top (G^l - A^l) \right)_{j,i} & \text{if} \quad F^l_{i,j} > 0 \\ 0 & \text{if} \quad F^l_{i,j} < 0 \end{cases}$$

Trade-off Content - Style

- The total loss is as follows

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

- Starting point $\longrightarrow$ White Gaussian noise
- number of iterations $n = 3000$

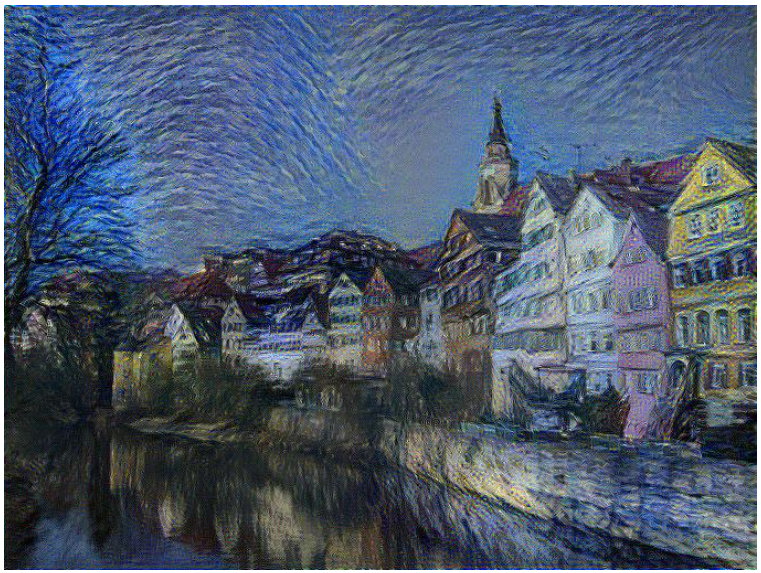

Figure: $\alpha/\beta = 10^{-1}$

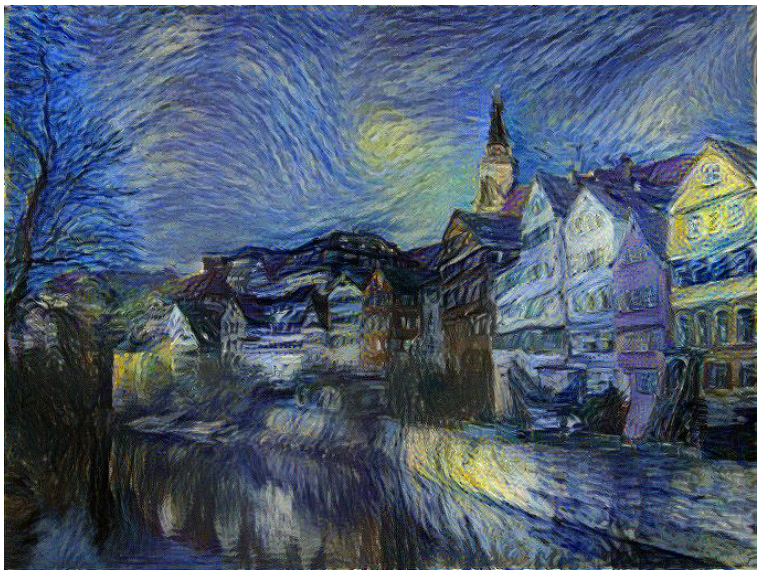

Figure: $\alpha/\beta = 10^{-2}$

Trade-off Content - Style

- The total loss is as follows

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

- Starting point $\longrightarrow$ White Gaussian noise
- number of iterations $n = 3000$

Figure: $\alpha/\beta = 10^{-3}$

Figure: $\alpha/\beta = 10^{-4}$

Layers' influence - Content

- Starting point $\longrightarrow$ White Gaussian noise
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

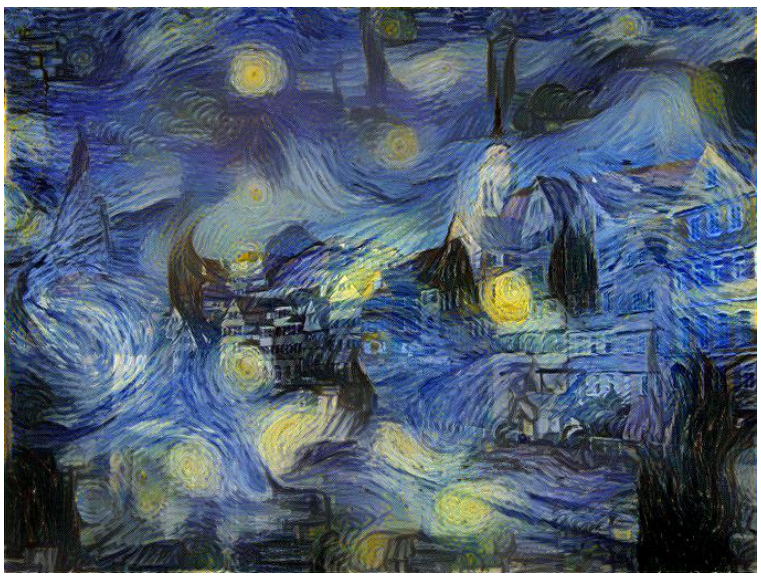

Figure: conv2_2

Figure: conv4_2

Layers' influence - Style

- Starting point $\longrightarrow$ White Gaussian noise
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

Figure: conv2_1

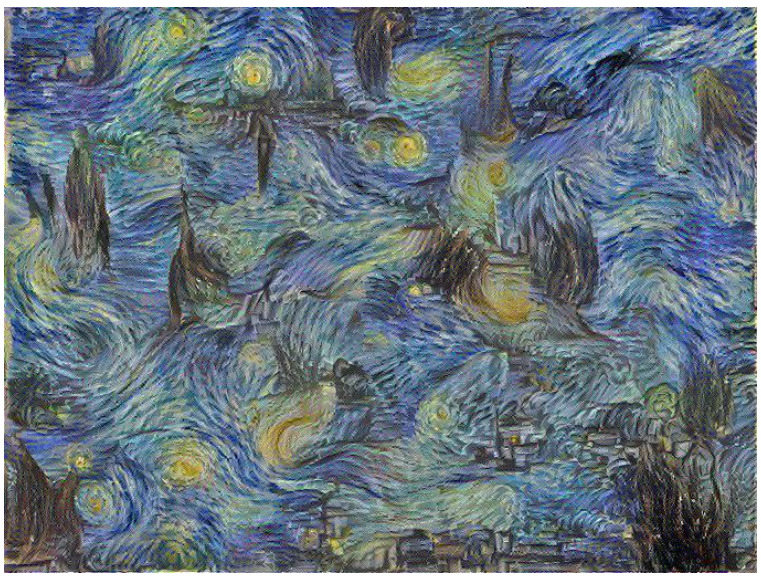

Figure: conv4_1

Layers' influence - Style

- Example of features map

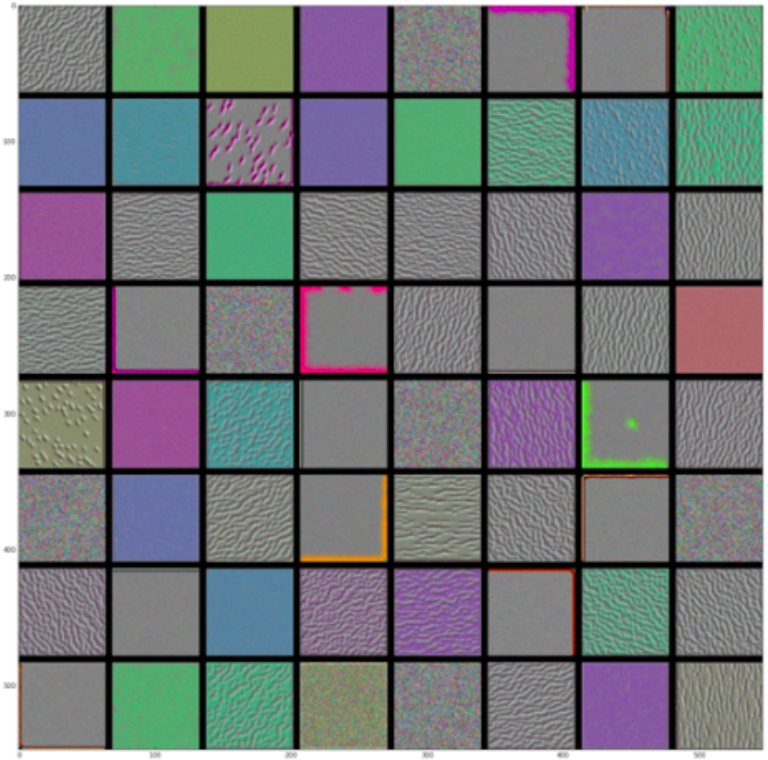

Figure: "low" level layer

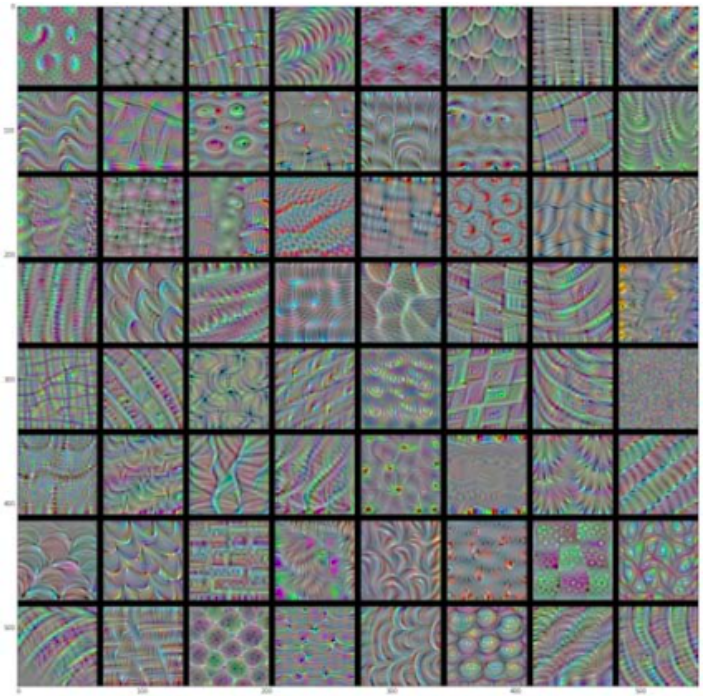

Figure: "high" level layer

Initialisation

- Starting point $\longrightarrow$ White Gaussian noise
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

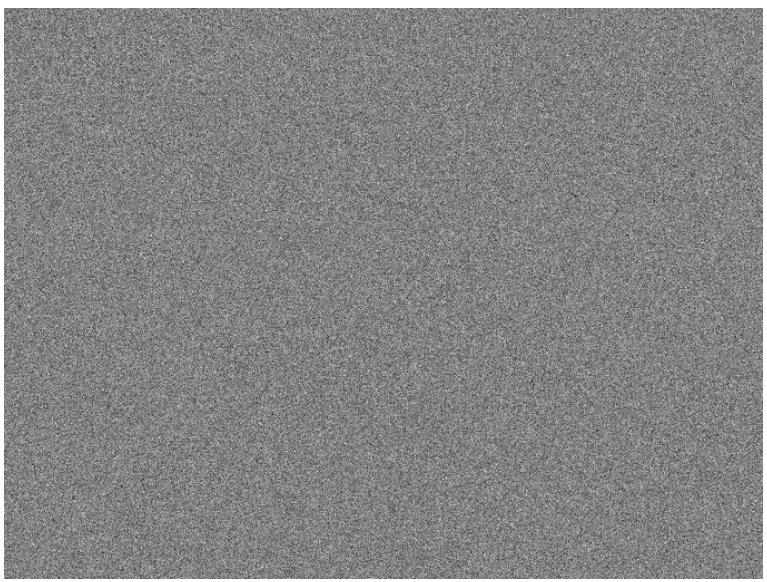

Figure: Gaussian noise

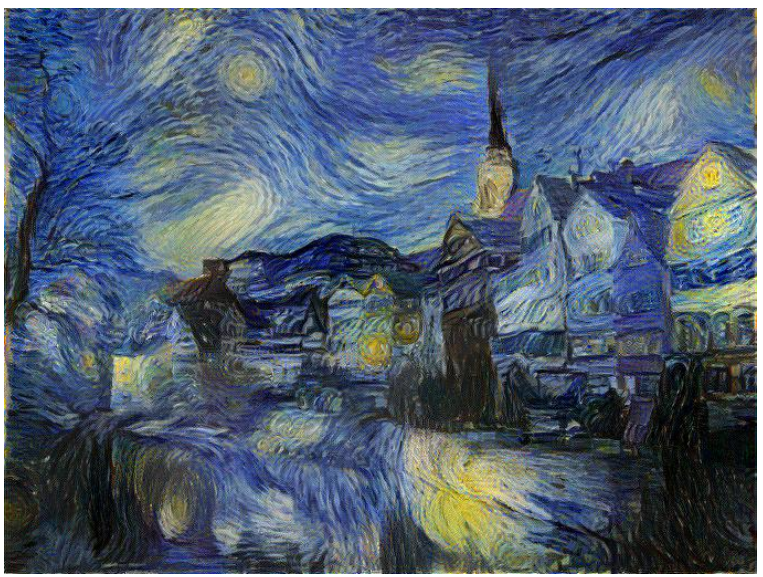

Figure: Outcome

Initialisation

- Starting point $\longrightarrow$ Content image
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

Figure: Content image

Figure: Outcome

Initialisation

- Starting point $\longrightarrow$ Pathological case
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

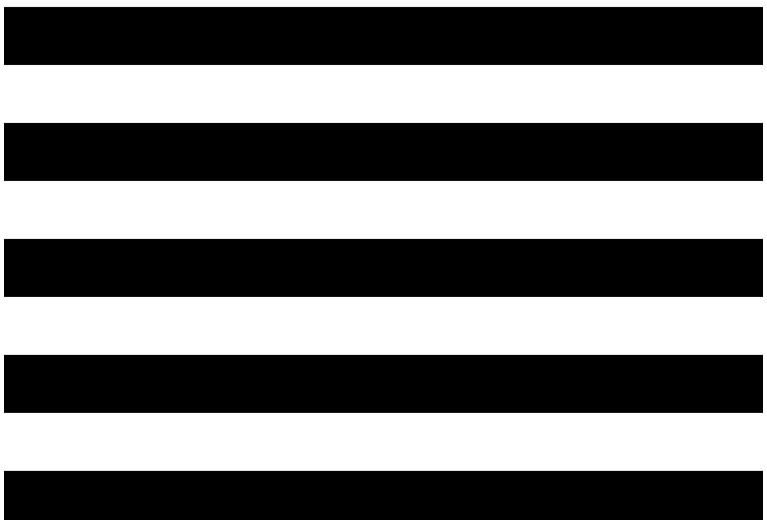

Figure: Pathological image

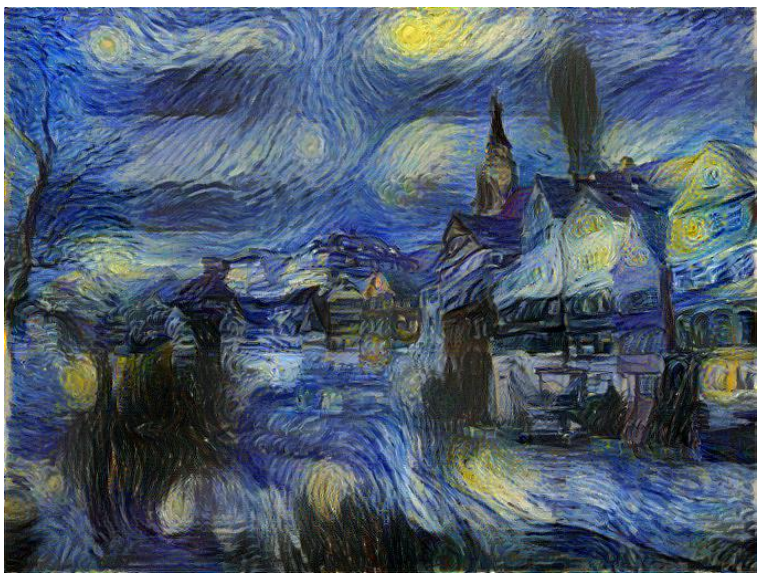

Figure: Outcome

Initialisation

- Starting point $\longrightarrow$ Pathological case
- number of iterations $n = 10000$
- weight factor $\alpha/\beta = 10^{-3}$

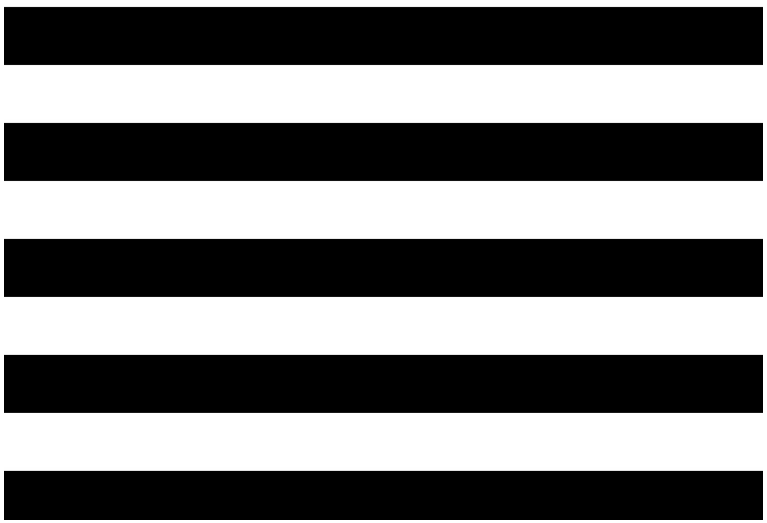

Figure: Content image

Figure: Outcome

Stochastic results

- Starting point $\longrightarrow$ Gaussian noise
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

Figure: Outcome 1

Figure: Outcome 2

Stochastic results

- Starting point $\longrightarrow$ Gaussian noise
- number of iterations $n = 3000$
- weight factor $\alpha/\beta = 10^{-3}$

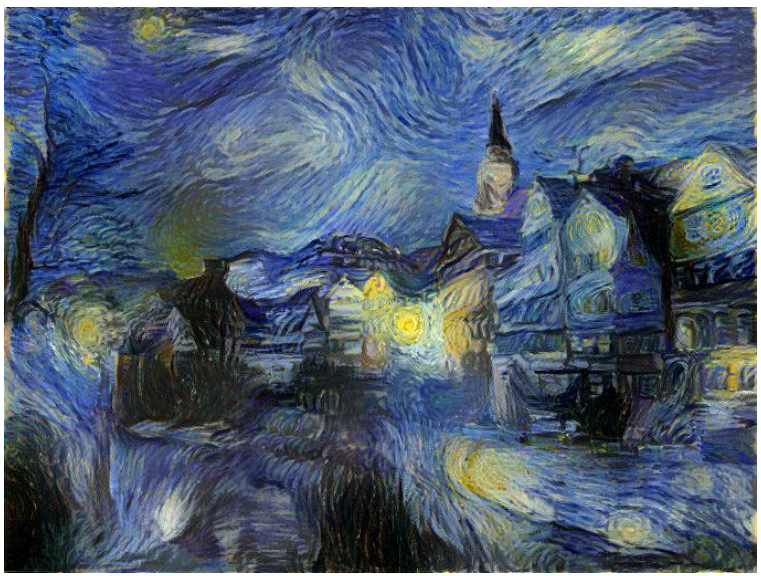

Figure: Outcome 3

Figure: Outcome 4

Photo realistic

- Starting point $\longrightarrow$ Content image
- number of iterations $n = 5000$
- weight factor $\alpha/\beta = 10^{-3}$

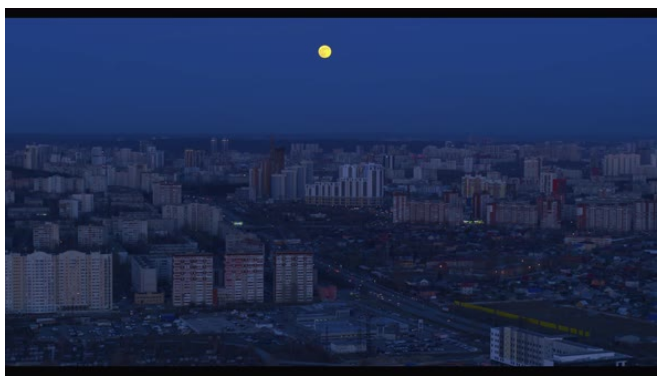

Figure: Dark image

Figure: Reference image

Photo realistic

- Starting point $\longrightarrow$ Content image
- number of iterations $n = 5000$
- weight factor $\alpha/\beta = 10^{-3}$

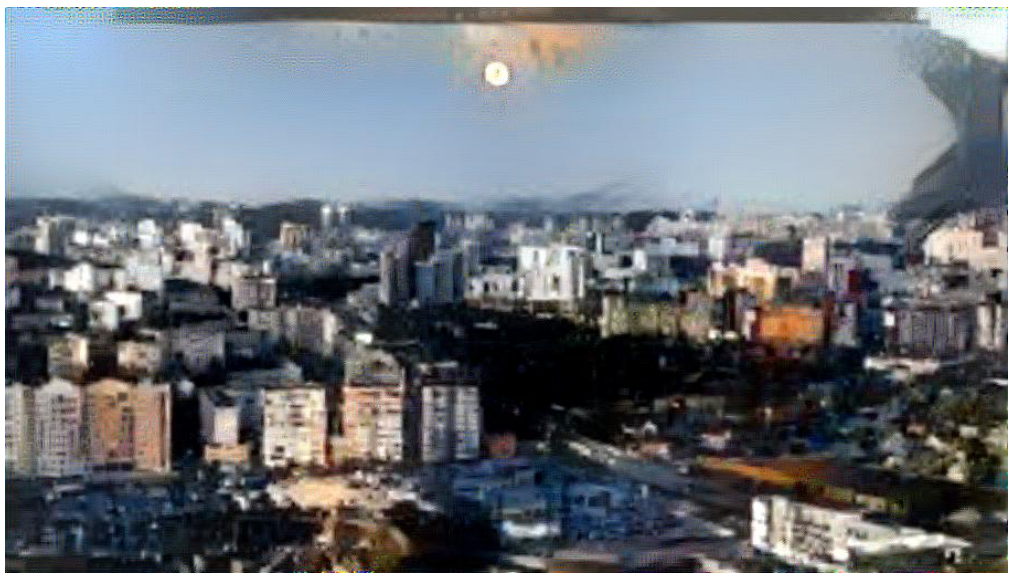

Figure: Outcome

Video

To be shown