An Exploration of Neural Style Transfer

Andrew Kang

May 30, 2017

Table of Contents

Introduction

Introduction: The paper that started it all

A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

Introduction: Motivations

- Emulate human capabilities
- Algorithmic understanding to artistic imagery
- Computer generation of art

Introduction: What is style transfer?

- Two input images
  - Content image
  - Style image

Introduction: What is style transfer?

- Two input images
  - Content image
  - Style image
- One output image
  - “Combined” image

Introduction: What is style transfer?

- Two input images
  - Content image
  - Style image
- One output image
  - "Combined" image

Simple concept!

Introduction: What is style transfer?

- Two input images
  - Content image
  - Style image
- One output image
  - "Combined" image

Simple concept!

...But how do we do it?

Feature spaces

- Each image has a representation in the network space
  - Outputs of convolutional layers

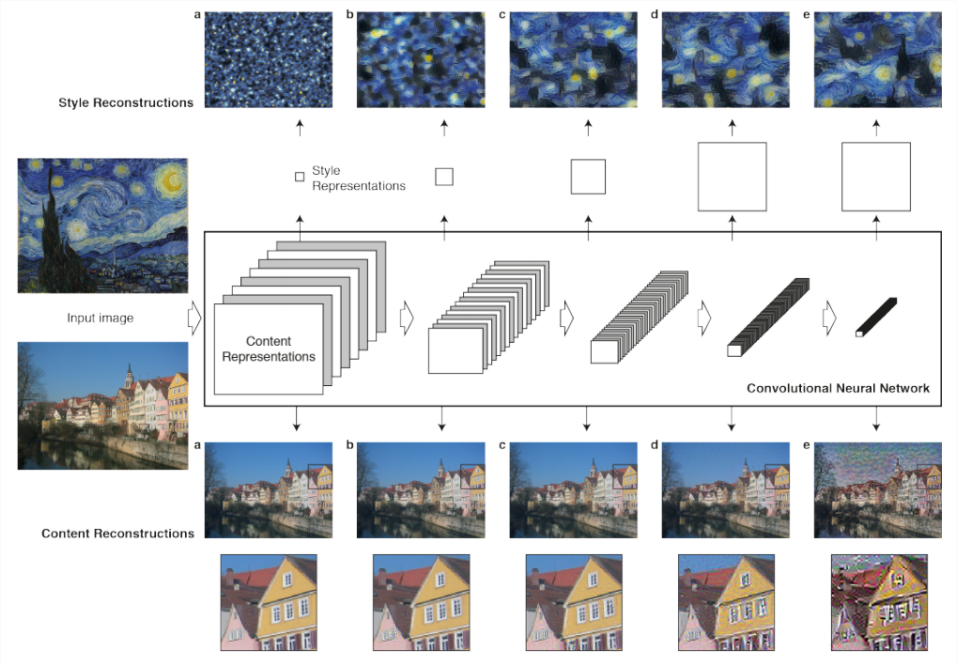
Introduction: Feature spaces

Feature spaces

- Each image has a representation in the network space
  - Outputs of convolutional layers
- **Content:** responses of higher layers
- **Style:** correlation between filter responses of layers

Feature spaces

- Each image has a representation in the network space
  - Outputs of convolutional layers
- **Content:** responses of higher layers
- **Style:** correlation between filter responses of layers



Introduction: Content loss

- A convolutional layer has N_l filters of size M_l
- $P^\ell, F^\ell \in \mathcal{R}^{N_\ell \times M_\ell}$ are the responses of layer ℓ from the content and combined images

Introduction: Content loss

- A convolutional layer has N_l filters of size M_l
- $P^\ell, F^\ell \in \mathcal{R}^{N_\ell \times M_\ell}$ are the responses of layer ℓ from the content and combined images

Content loss & derivative:

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^l - P_{ij}^l\right)^2 .$$

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^l} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 . \end{cases}$$

Introduction: Style loss

- Gram matrix $G^{\ell} \in \mathcal{R}^{N_\ell \times N_\ell}$ encodes correlations between filter responses

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

Introduction: Style loss

- Gram matrix $G^\ell \in \mathcal{R}^{N_\ell \times N_\ell}$ encodes correlations between filter responses

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

- $A^\ell, G^\ell \in \mathcal{R}^{N_\ell \times N_\ell}$ are the Gram matrices of the responses of layer ℓ from the style and combined images
- Helper term:

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

Introduction: Style loss

Style loss & derivative:

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left((F^l)^{\mathrm{T}} \left(G^l - A^l \right) \right)_{ji} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 \,. \end{cases}$$

Introduction: Variation loss

- An additional loss term seen in other papers
- In our implementation, sum of horizontal and vertical variations

Introduction: Variation loss

- An additional loss term seen in other papers
- In our implementation, sum of horizontal and vertical variations

Variation loss: (loose definition)

$$\sum_i \sum_j \left(\left(F^{\ell}_{i+1,j} - F^{\ell}_{i,j} \right)^2 + \left(F^{\ell}_{i,j+1} - F^{\ell}_{i,j} \right)^2 \right)^{1.25}$$

Introduction: Total loss

Total loss:

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

Introduction: Reconstruction

- Start from **white noise** image
- Perform **gradient descent** until total loss is minimized
- Responses grow increasingly similar

Introduction: Remarks

- **Lower layers** give pixel-wise representations
- **Higher layers** give smoother abstract representations

Approach

Approach: Implementation

- VGG19 network
- Fully automated with concise argument parsing
- Can start from results of a previous run
- Minimizer class provides interface and logging

Approach: Implementation

- L-BFGS-B algorithm to minimize loss with memory constraints
- Loss gradients are handled using Keras backend
- `block4_conv2` used as content layer
- `block<n>_conv1` used as style layers
  - Each such layer gets equal normalized weight
  - All other layers get 0 weight

Experiments

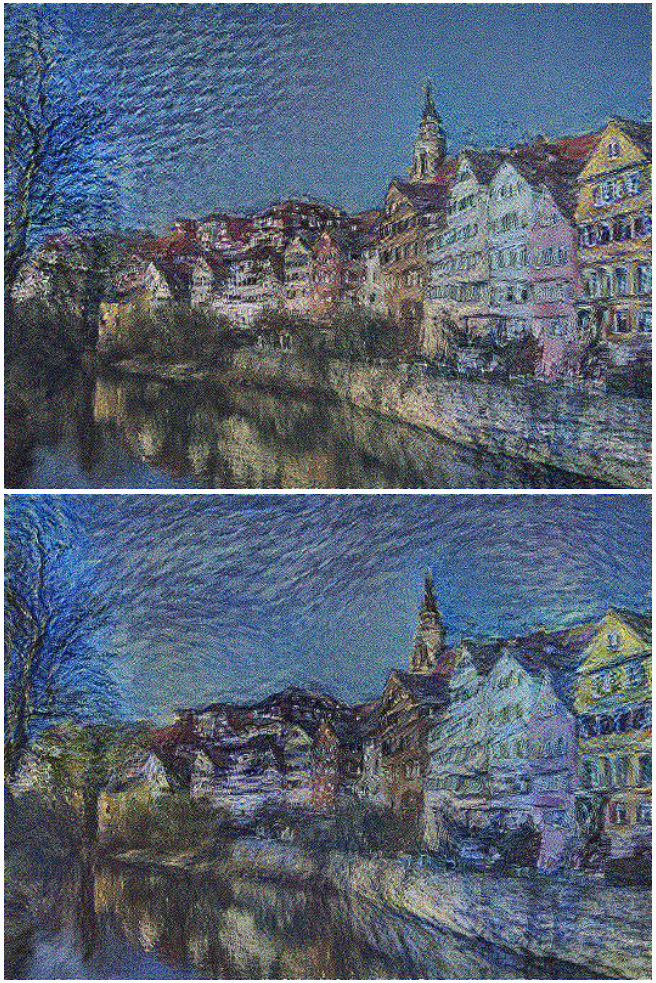
Experiments: Varying style weight

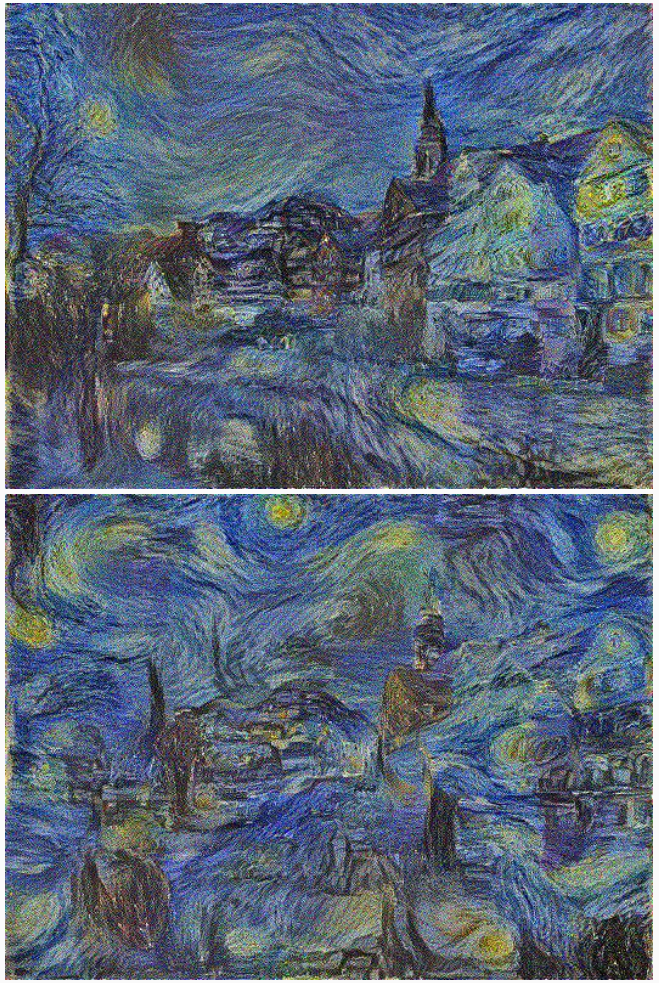
$$c = 1 \qquad s \in [10^2, 10^3, 10^4, 10^5] \qquad v = 0$$

+

Experiments: Varying style weight

$$c = 1 \qquad s \in [10^2, 10^3, 10^4, 10^5] \qquad v = 0$$





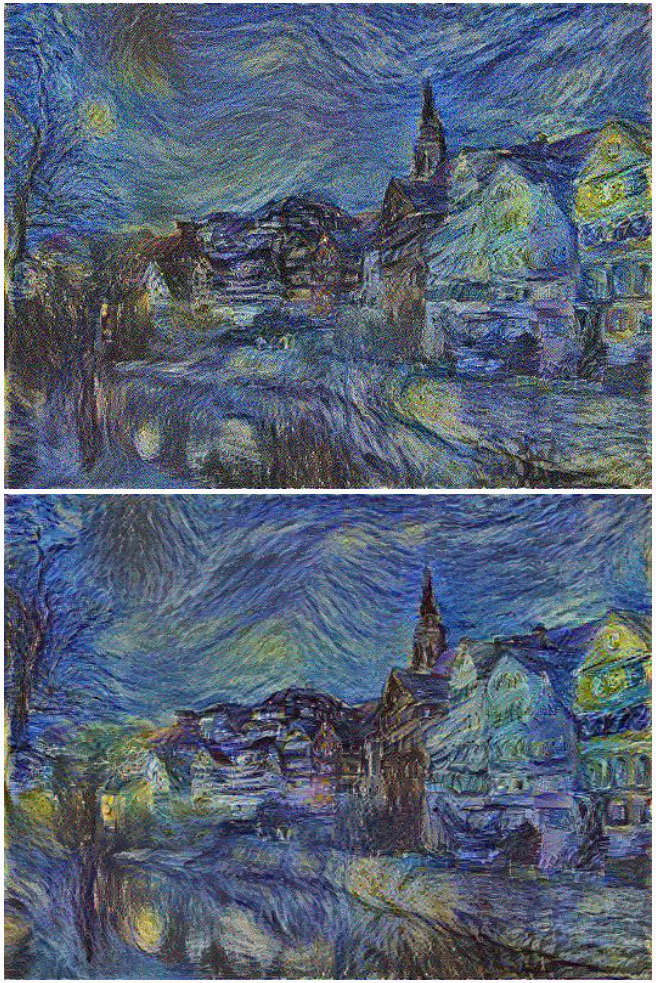
Experiments: Varying variation weight

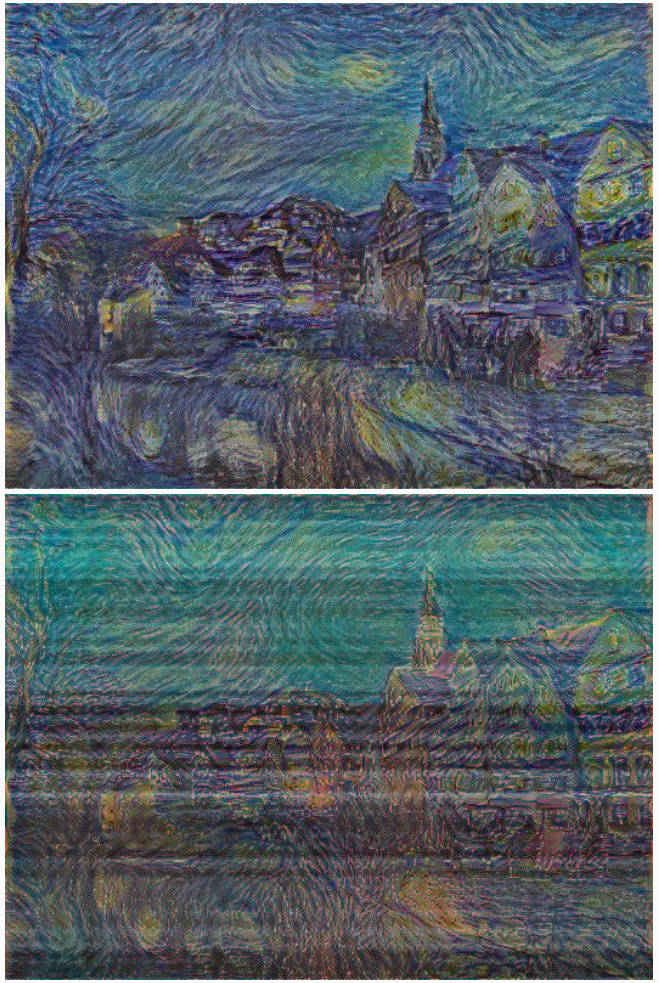
$$c = 1 \qquad s = 10^4 \qquad v \in [1, 10, 10^2, 10^3]$$

+

Experiments: Varying variation weight

$$c = 1 \qquad s = 10^4 \qquad v \in [1, 10, 10^2, 10^3]$$





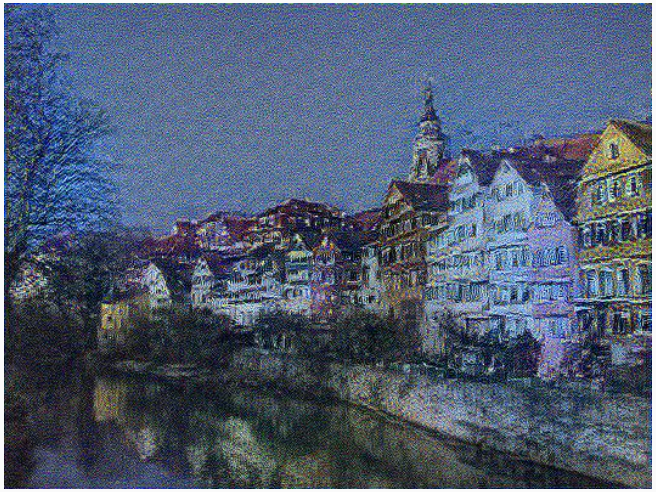
Experiments: Introducing additional higher layers

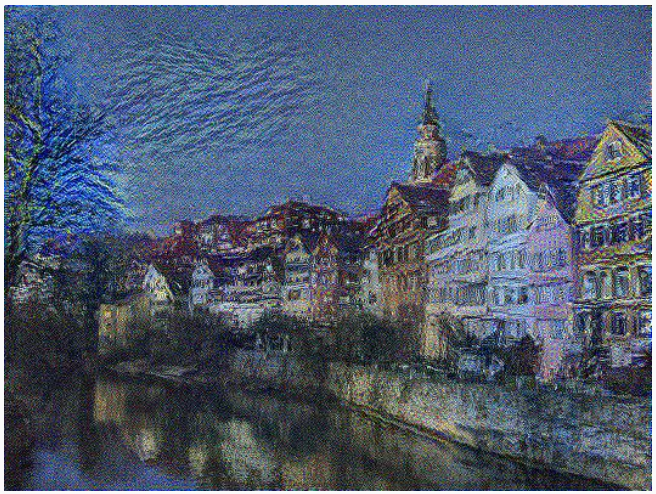
$c = 1$ $s = 10^4$ $v = 0$ $2 \rightarrow 5$ layers used

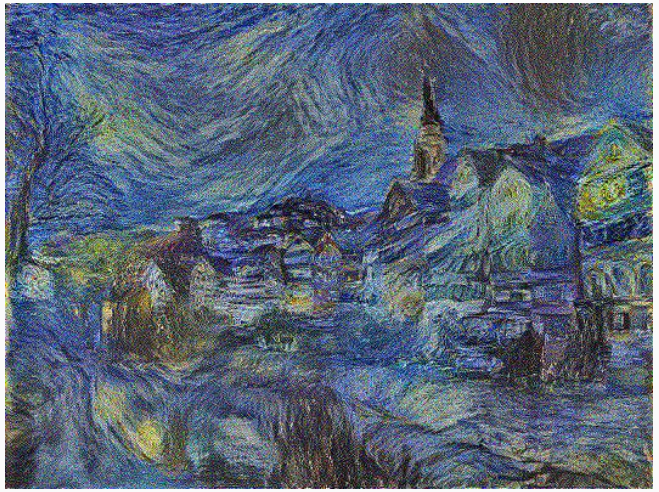
+

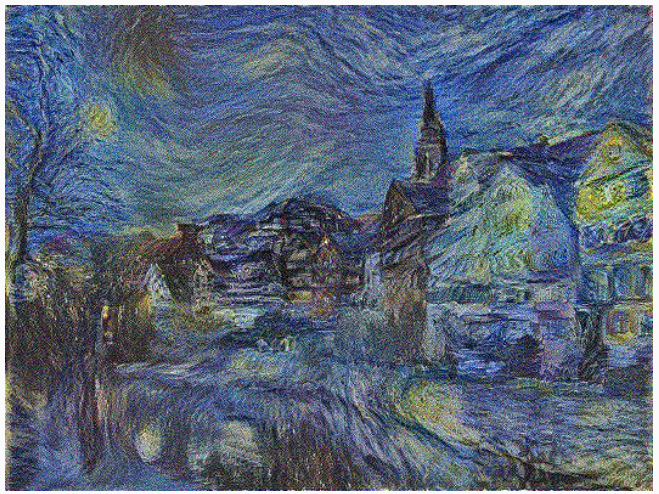
Experiments: Introducing additional higher layers

$c = 1$ $s = 10^4$ $v = 0$ $2 \rightarrow 5$ layers used









Experiments: Changes per 20 iterations

$$c = 1 \qquad s = 10^4 \qquad v = 0$$

+

Experiments: Changes per 20 iterations

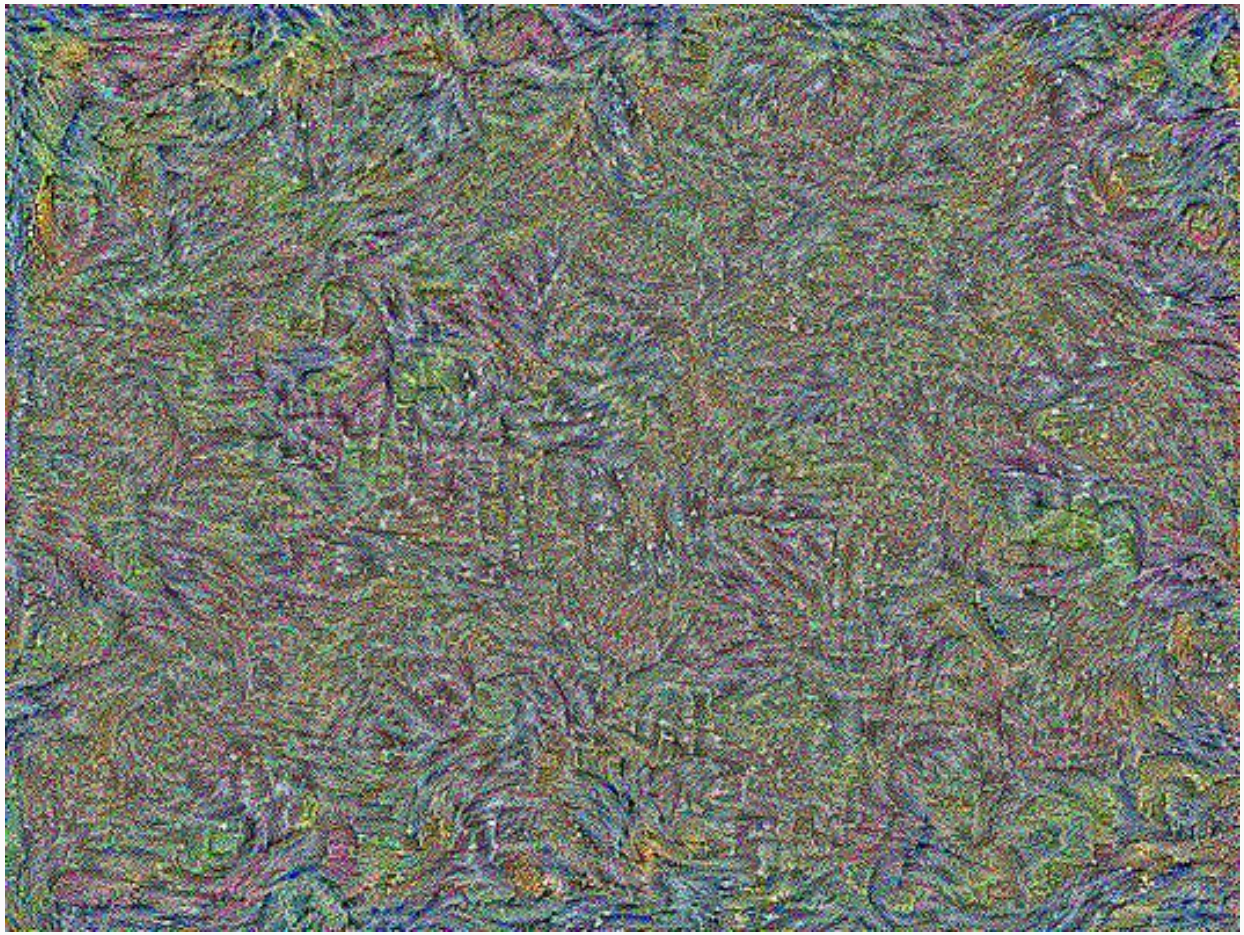
Start of animation

Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$

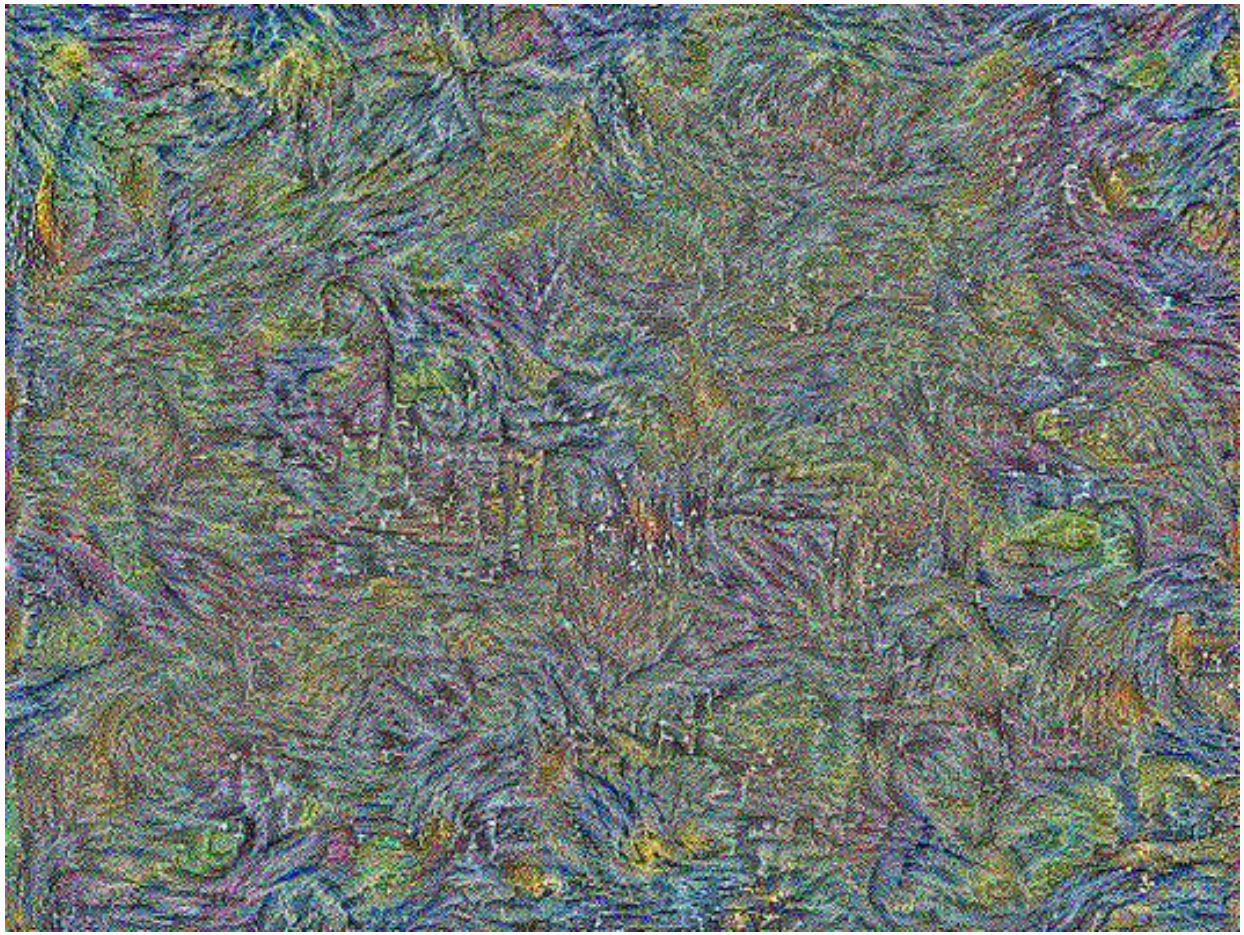
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



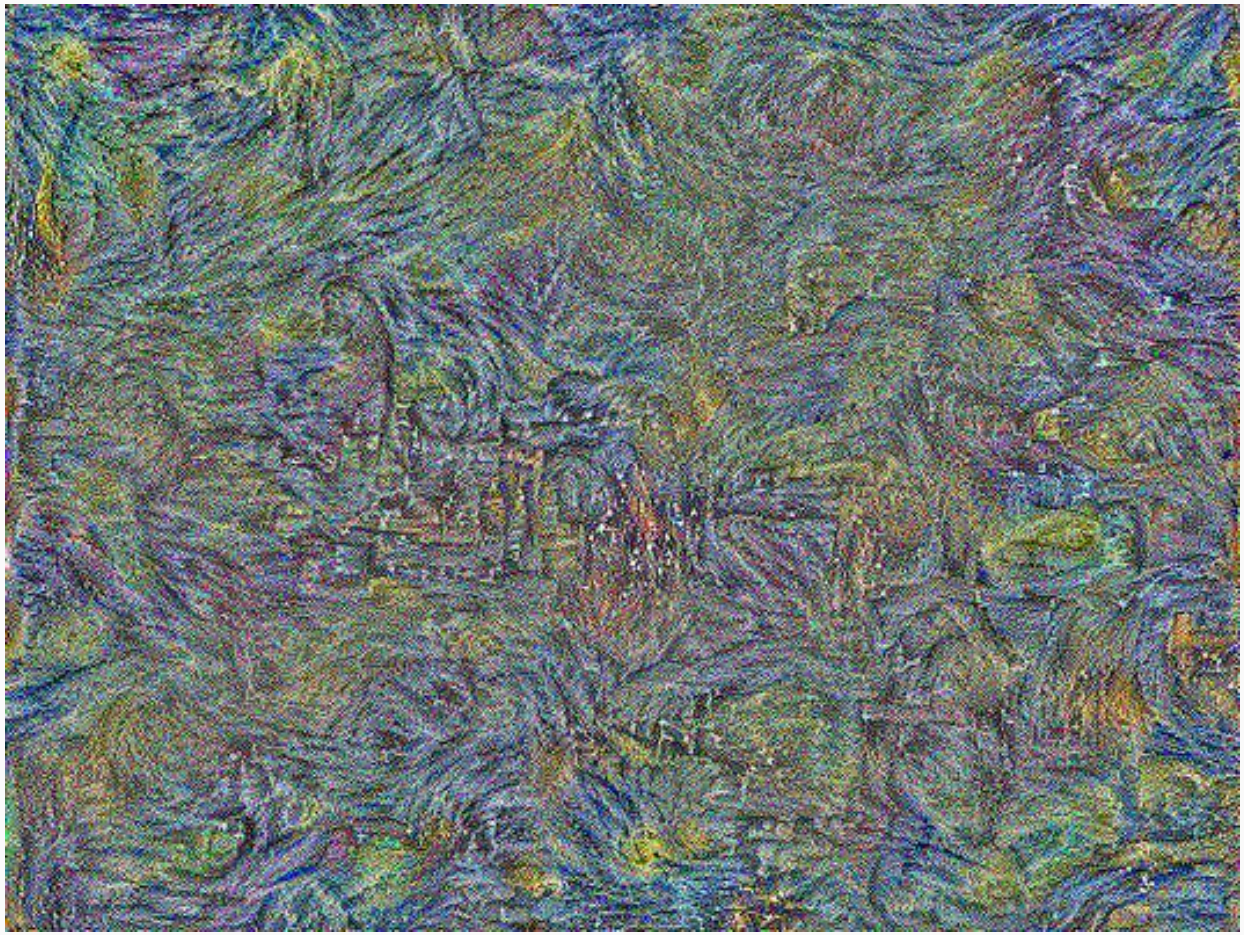
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



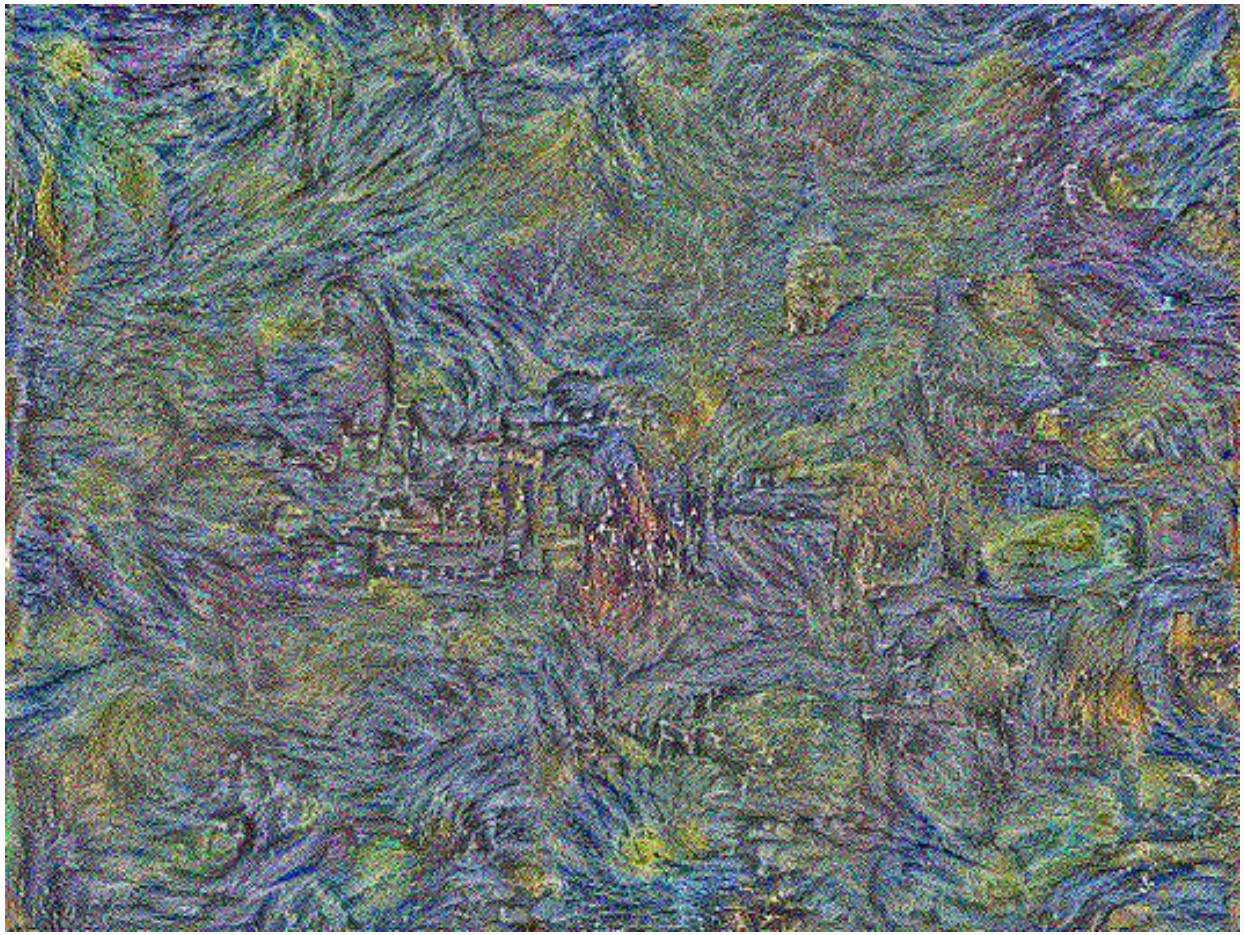
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



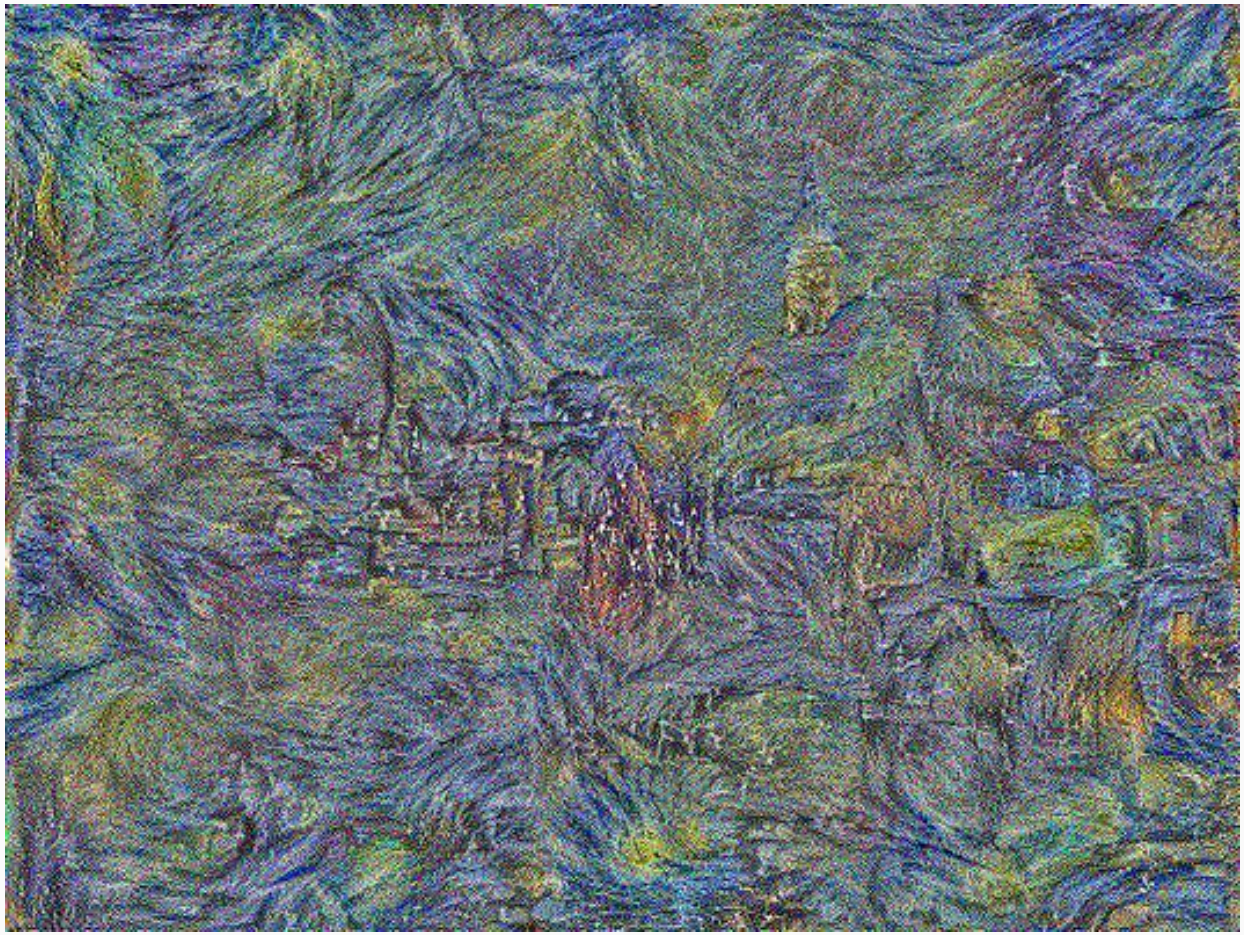
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



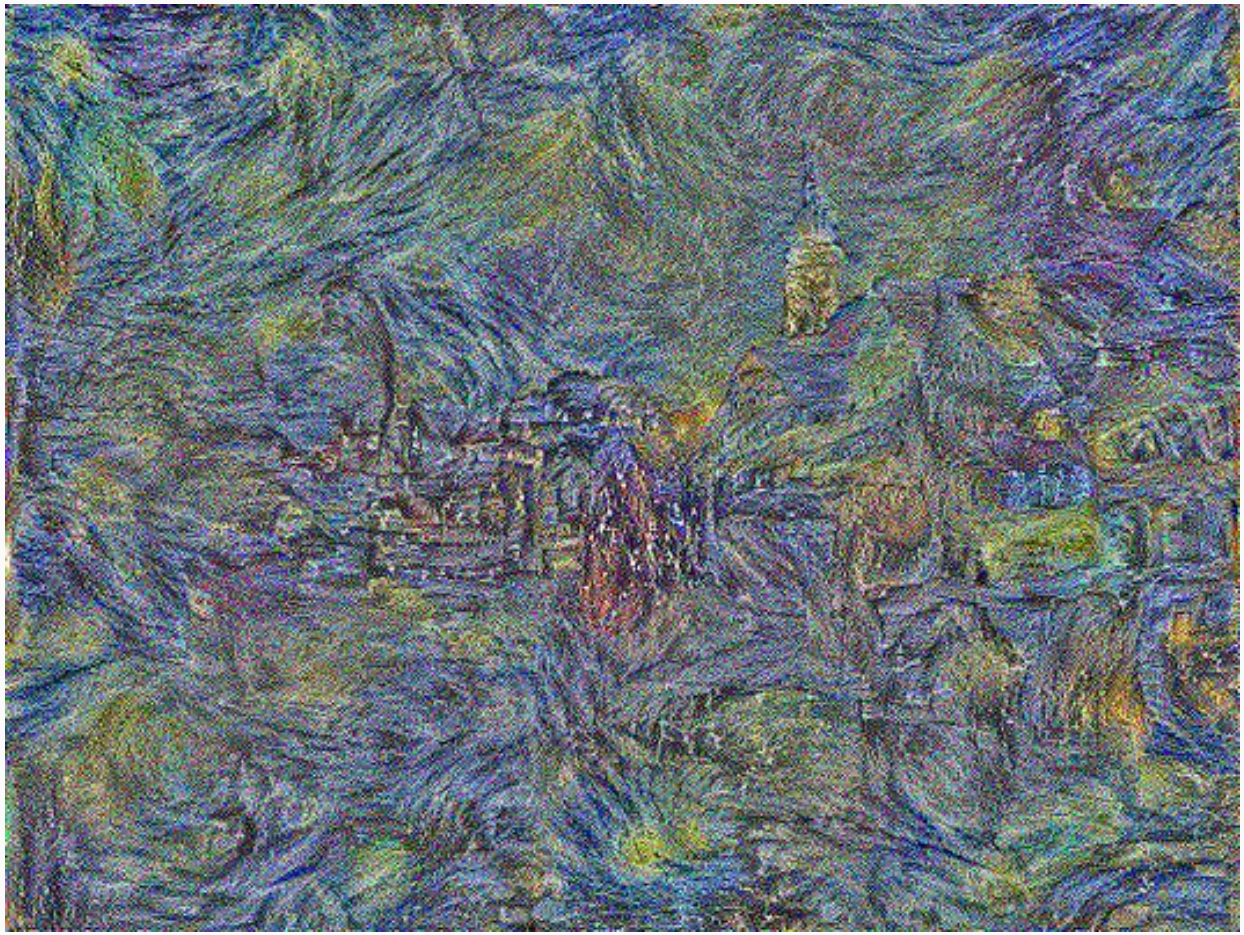
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



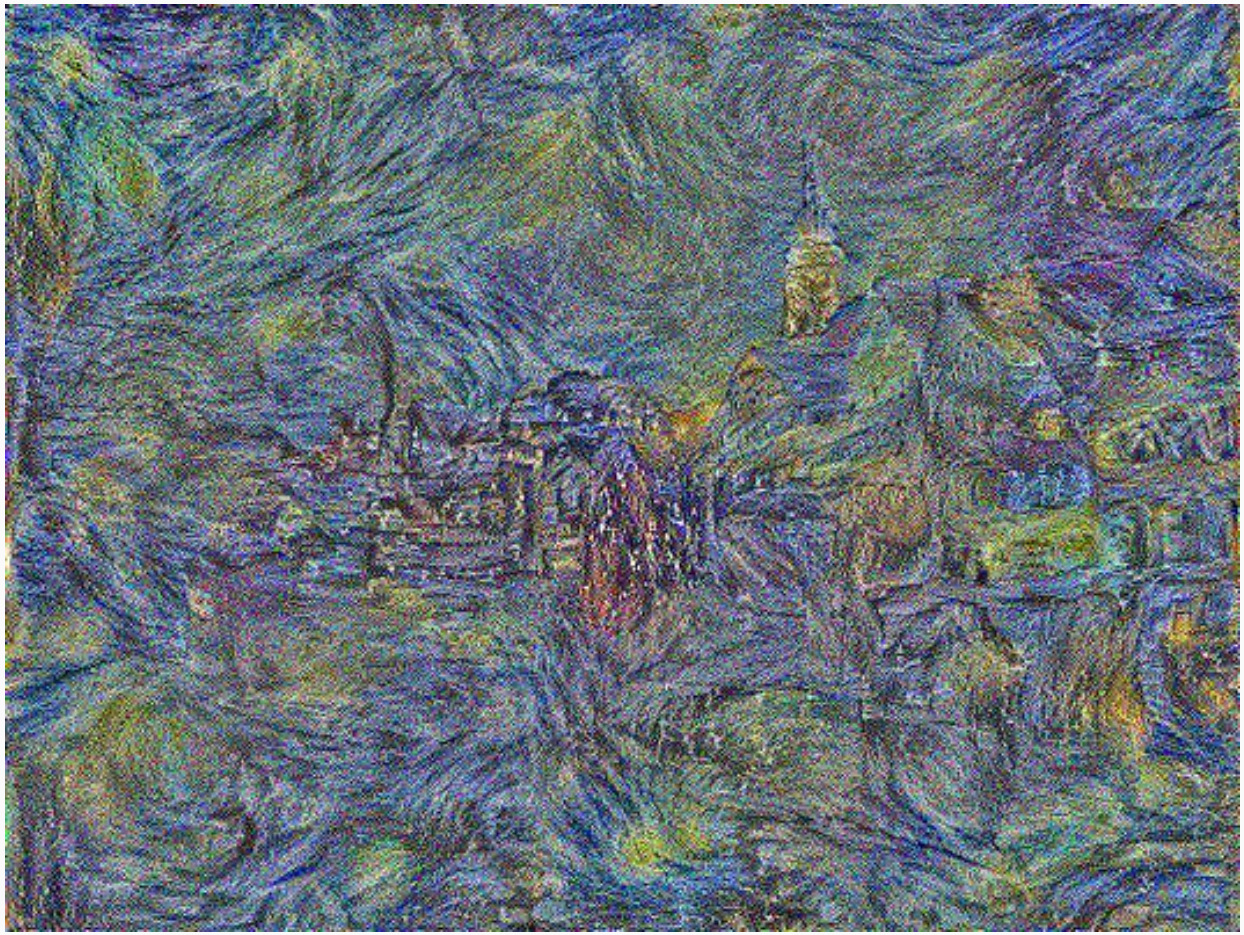
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



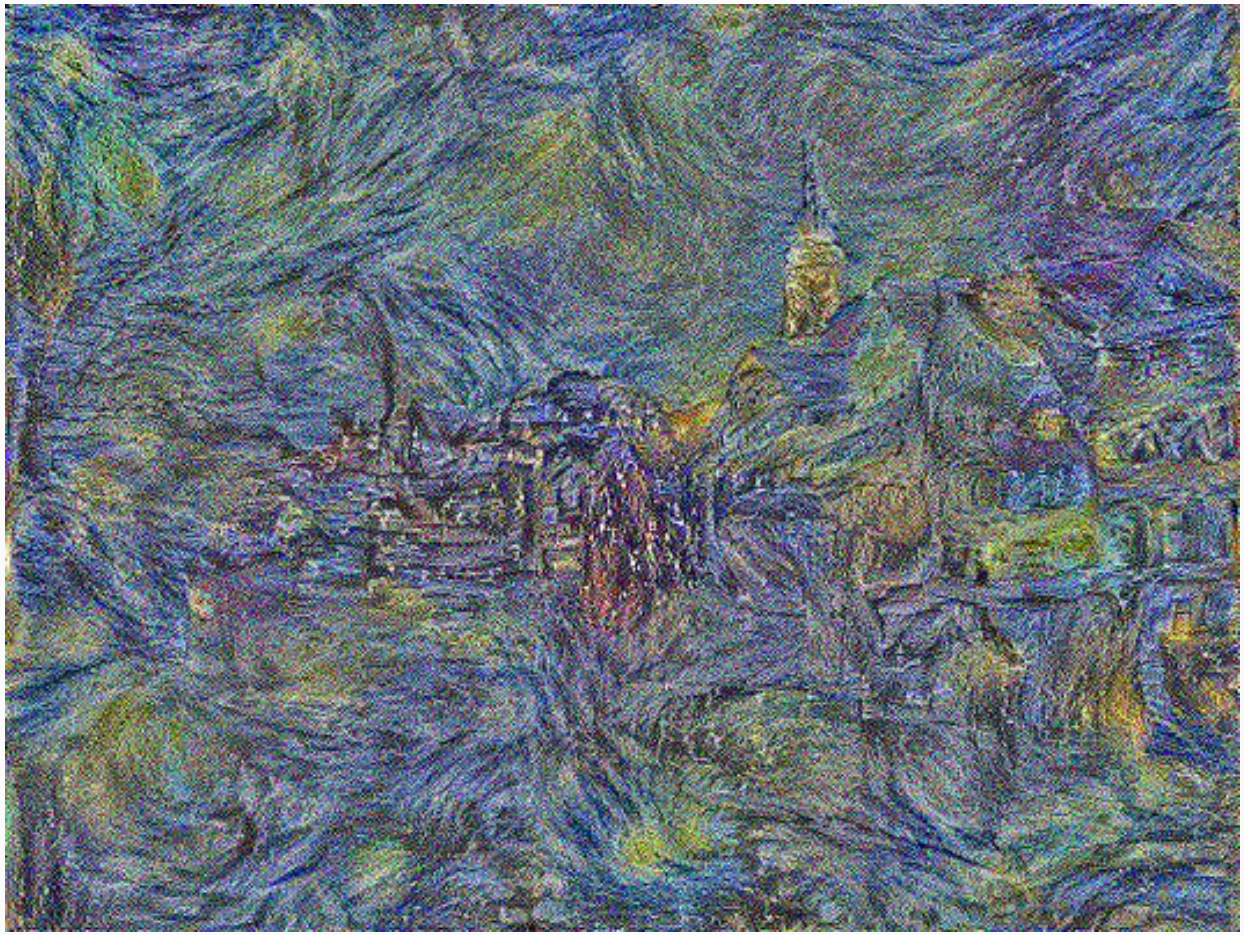
Experiments: Changes per 20 iterations

$c = 1$ $s = 10^4$ $v = 0$



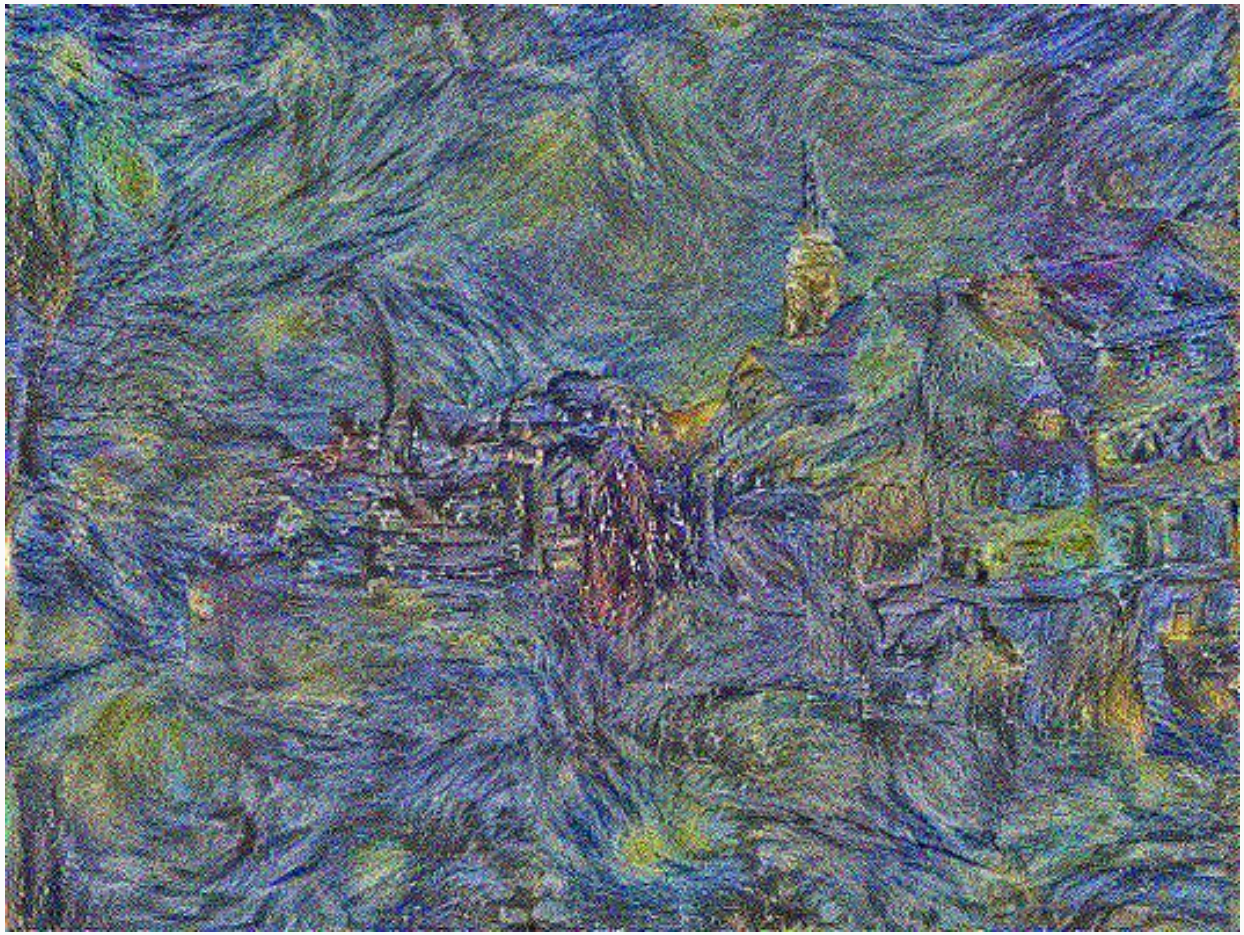
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



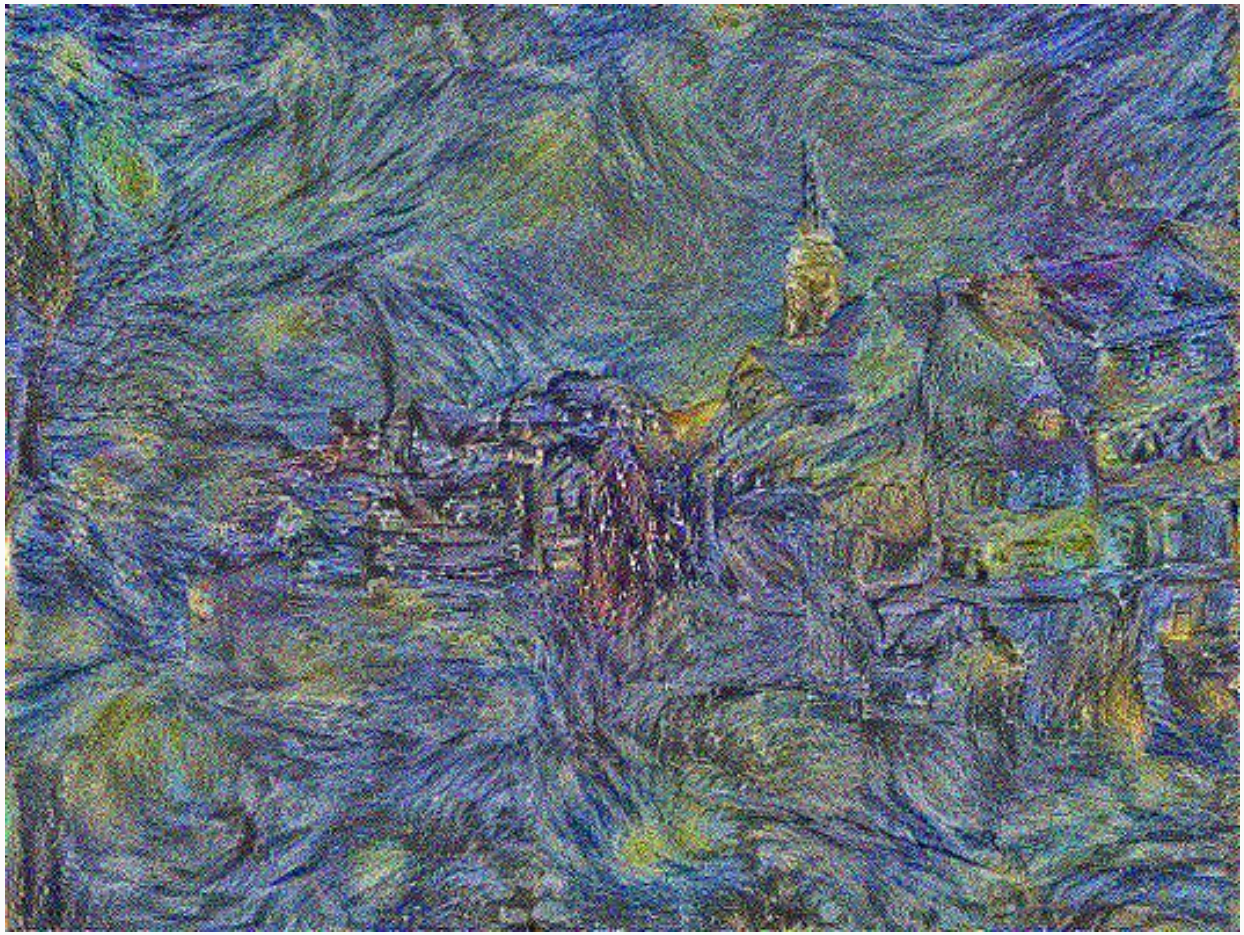
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



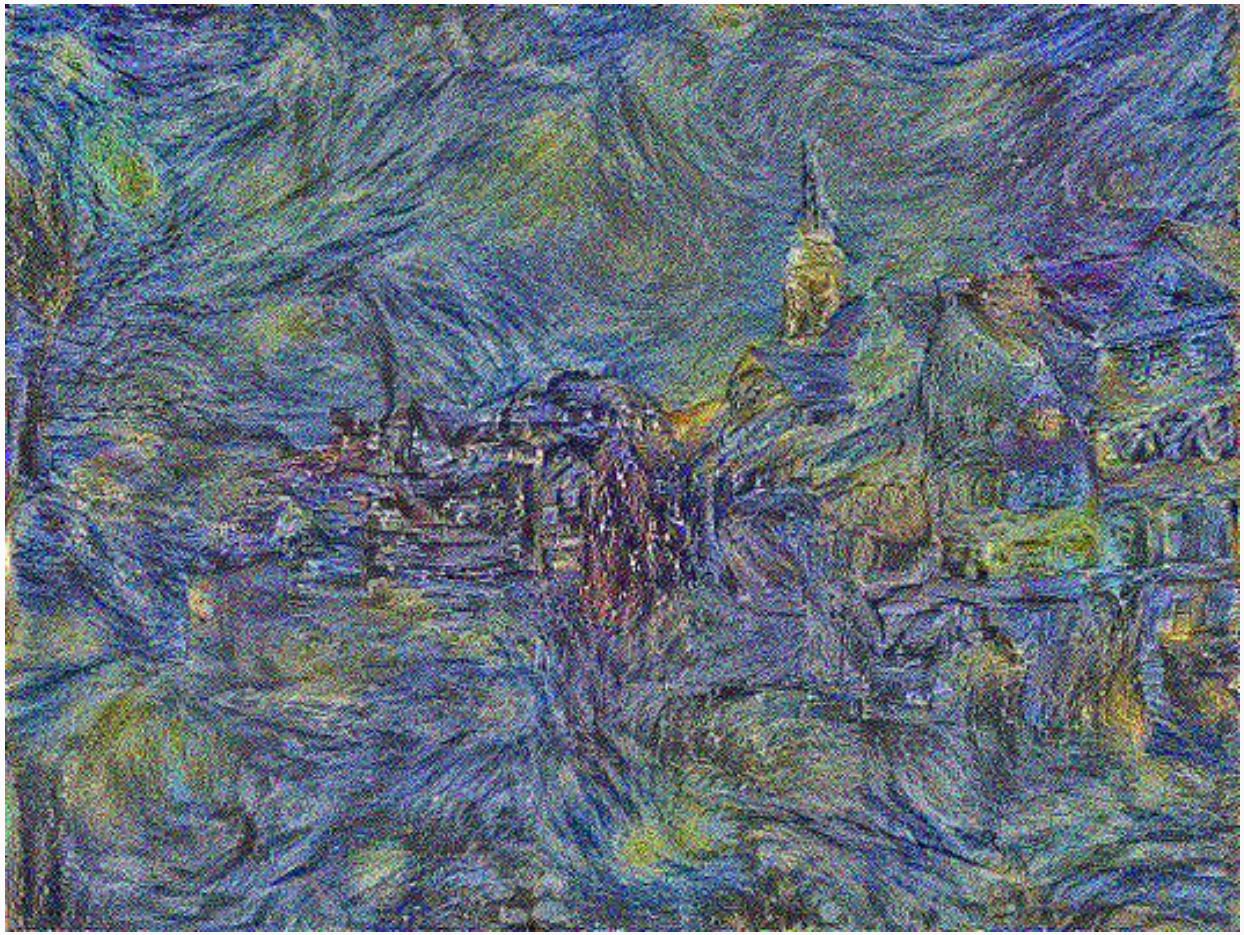
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



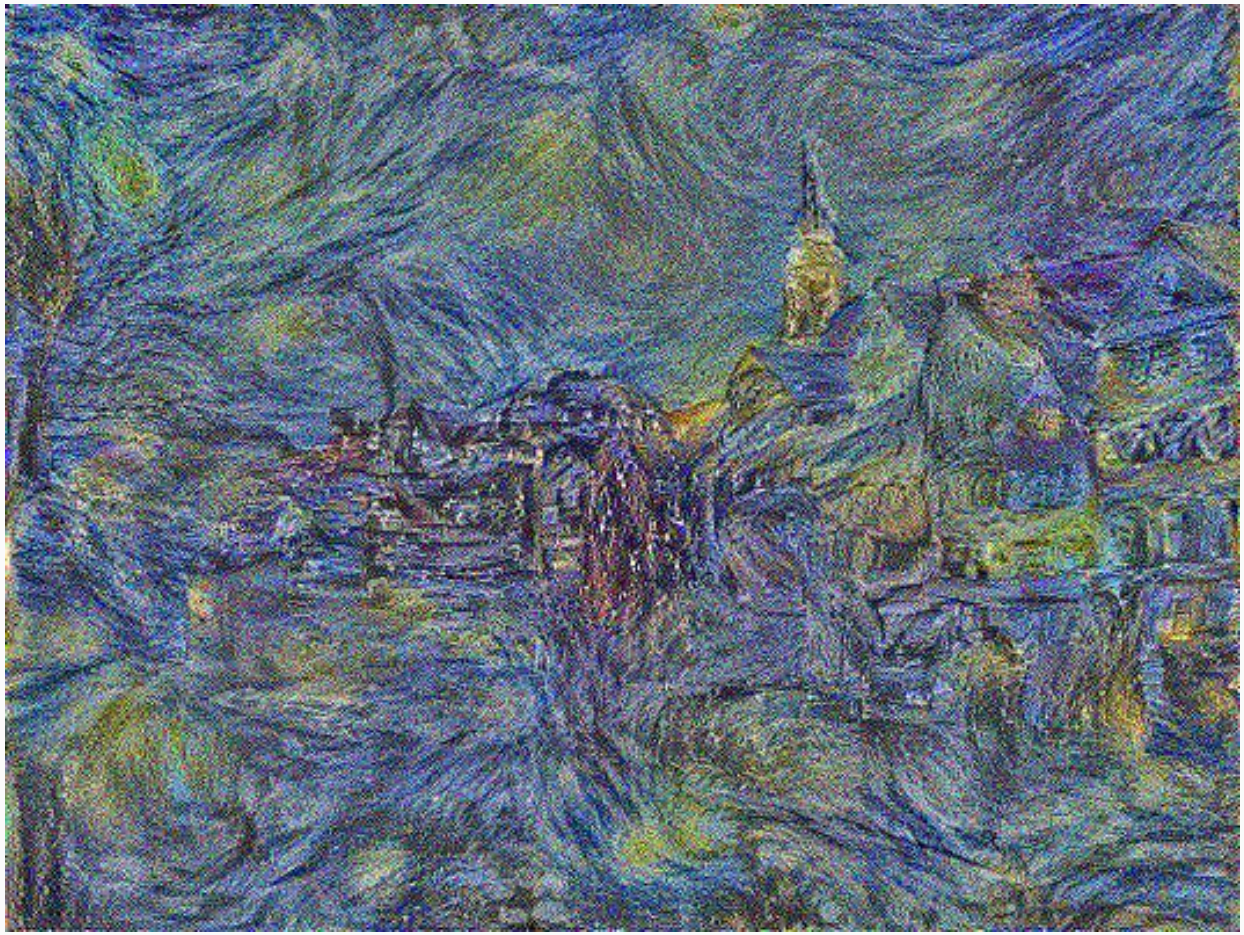
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



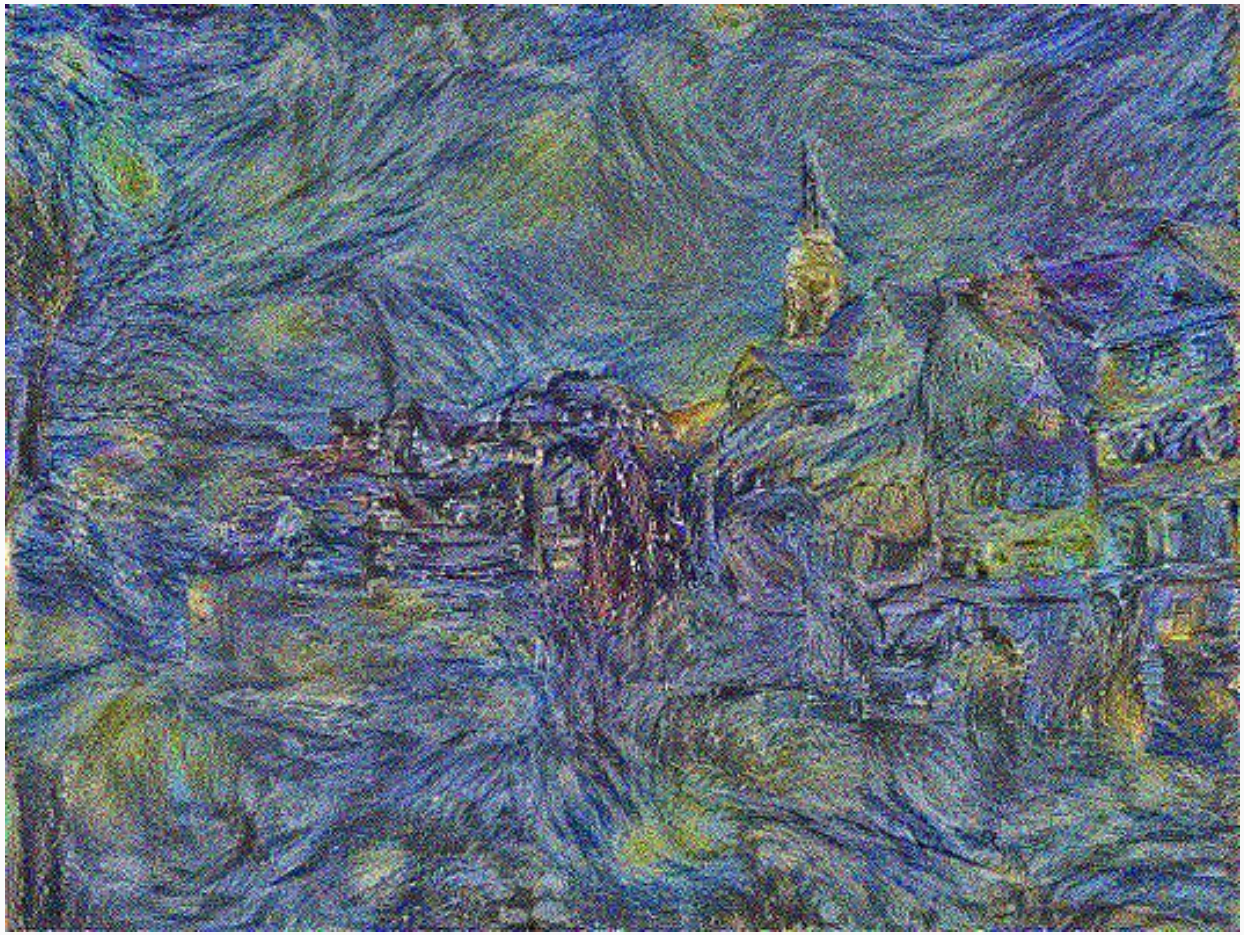
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



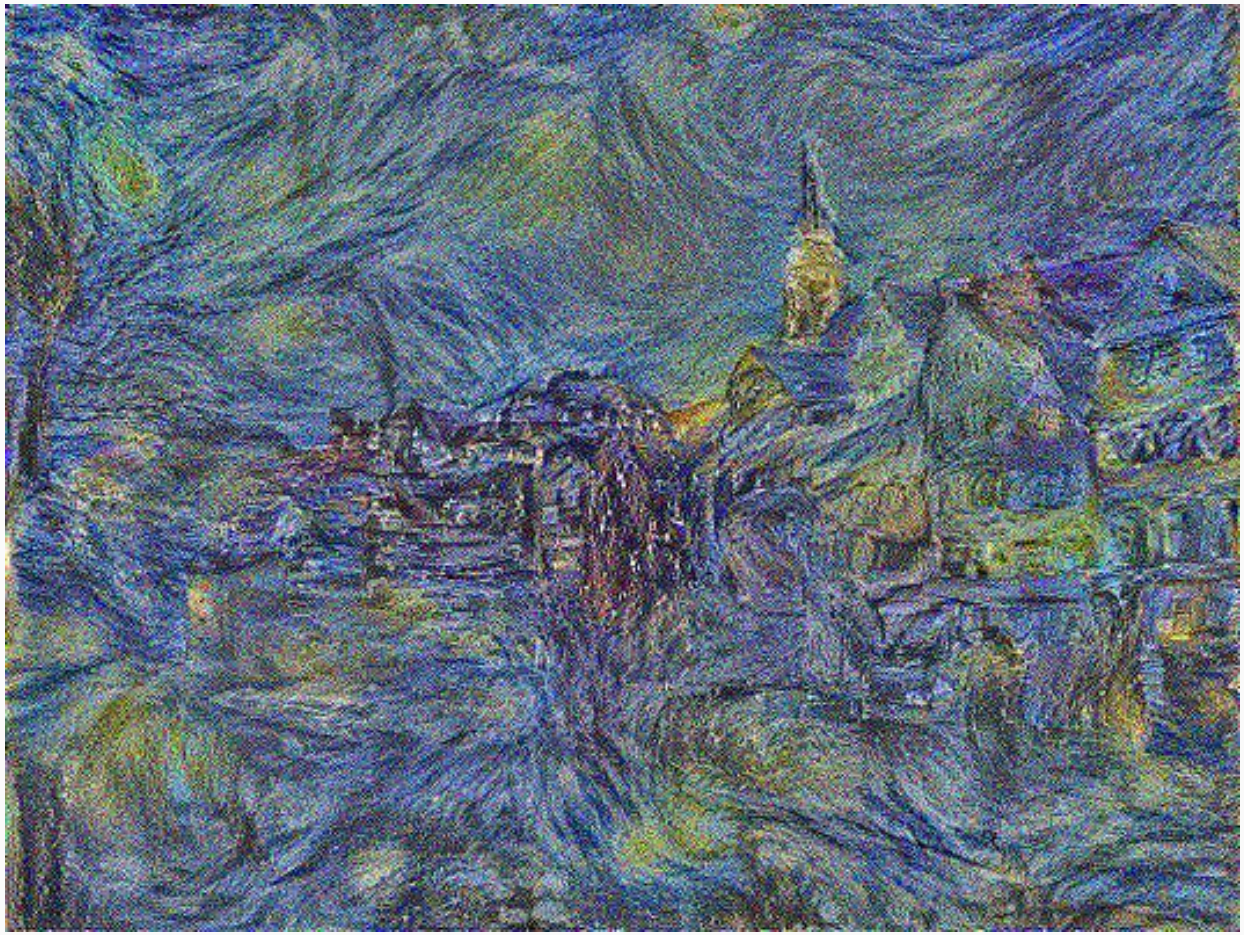
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



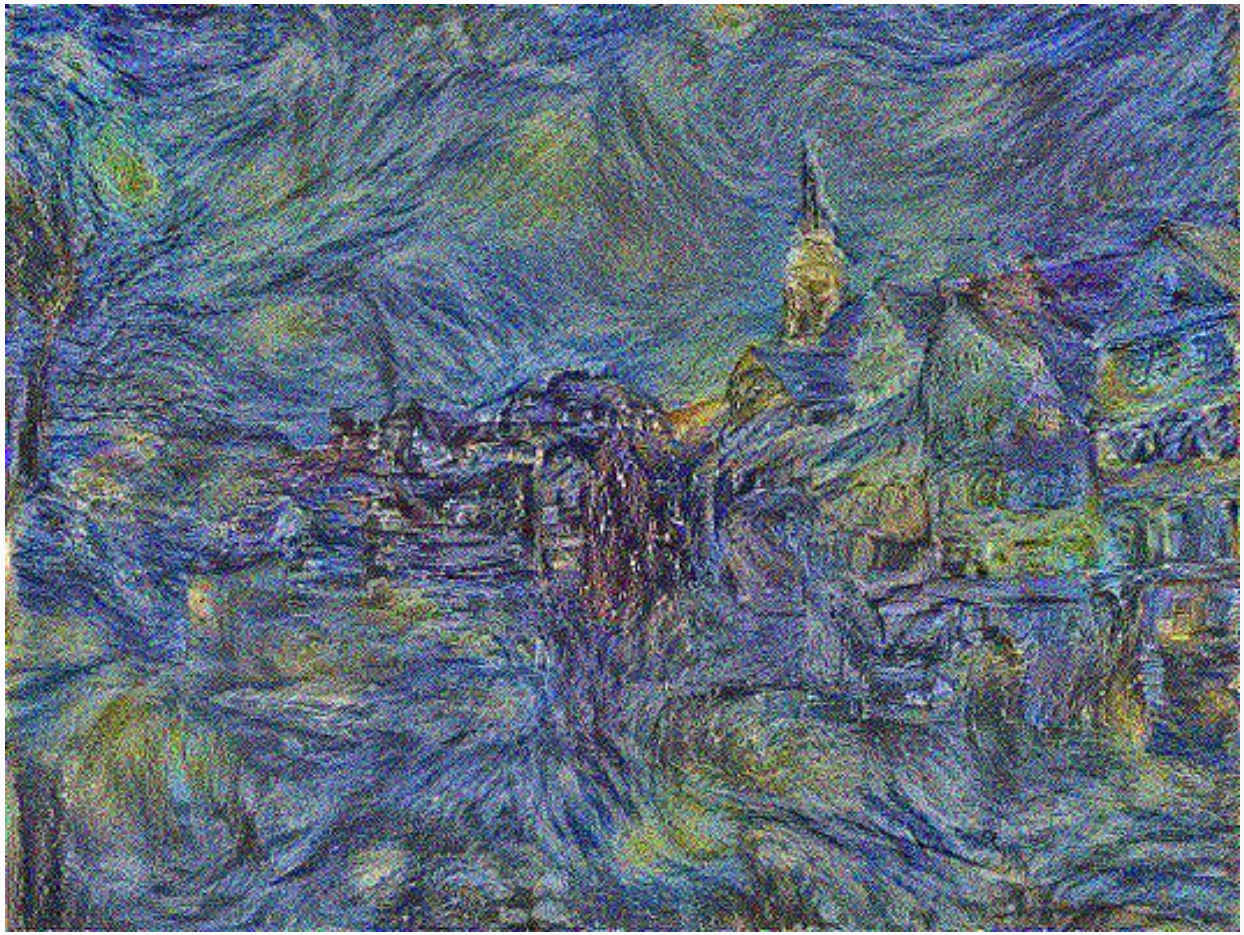
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



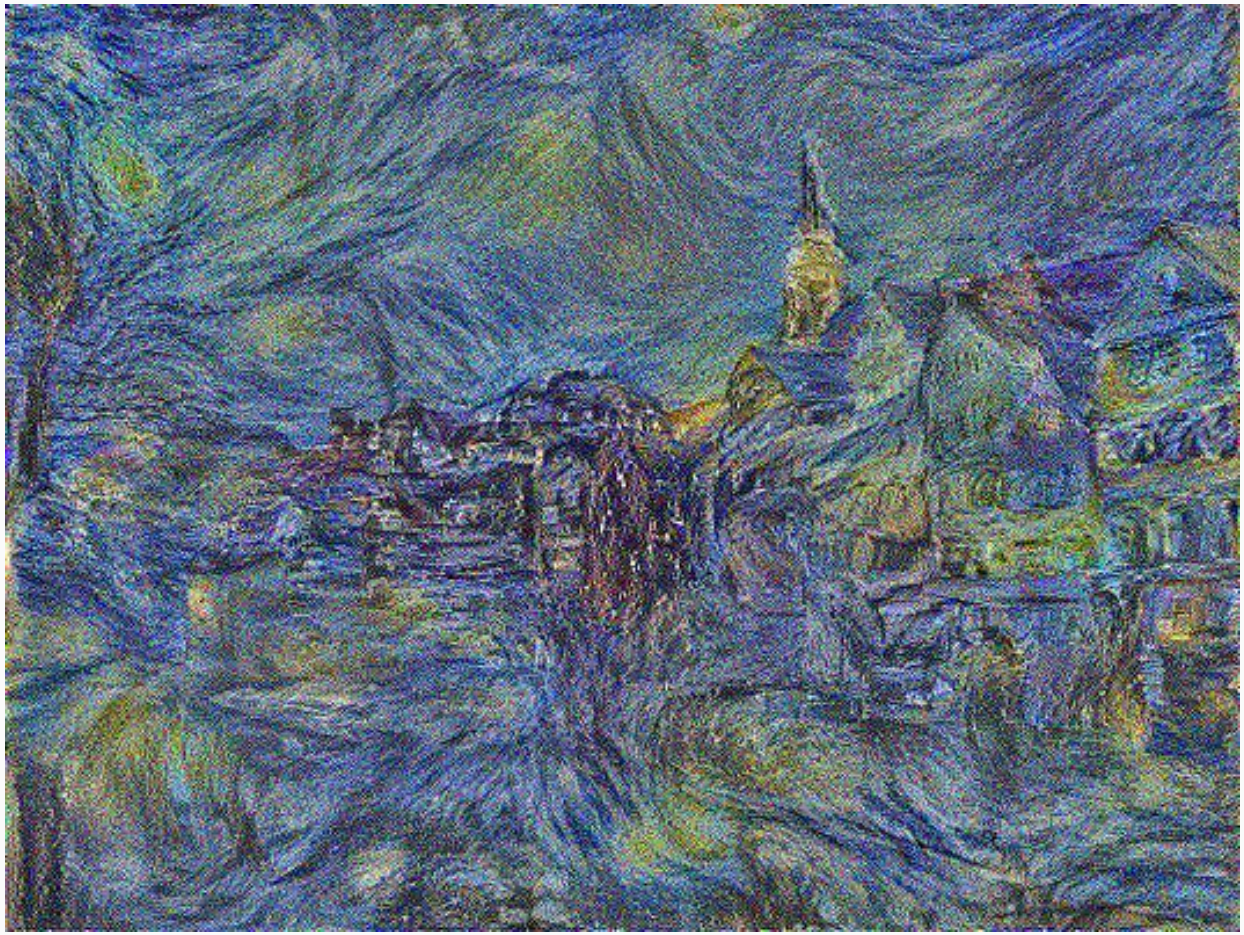
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



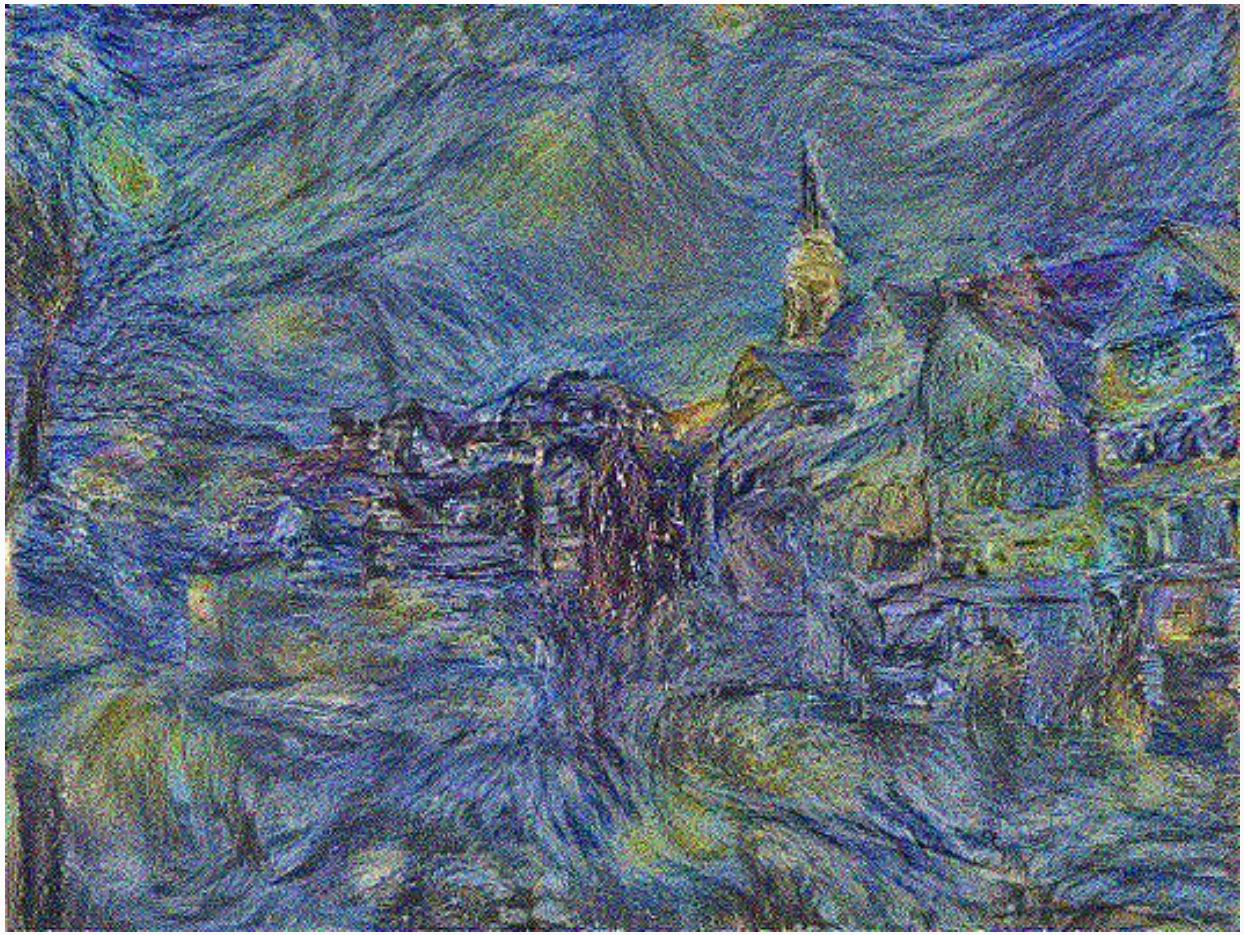
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



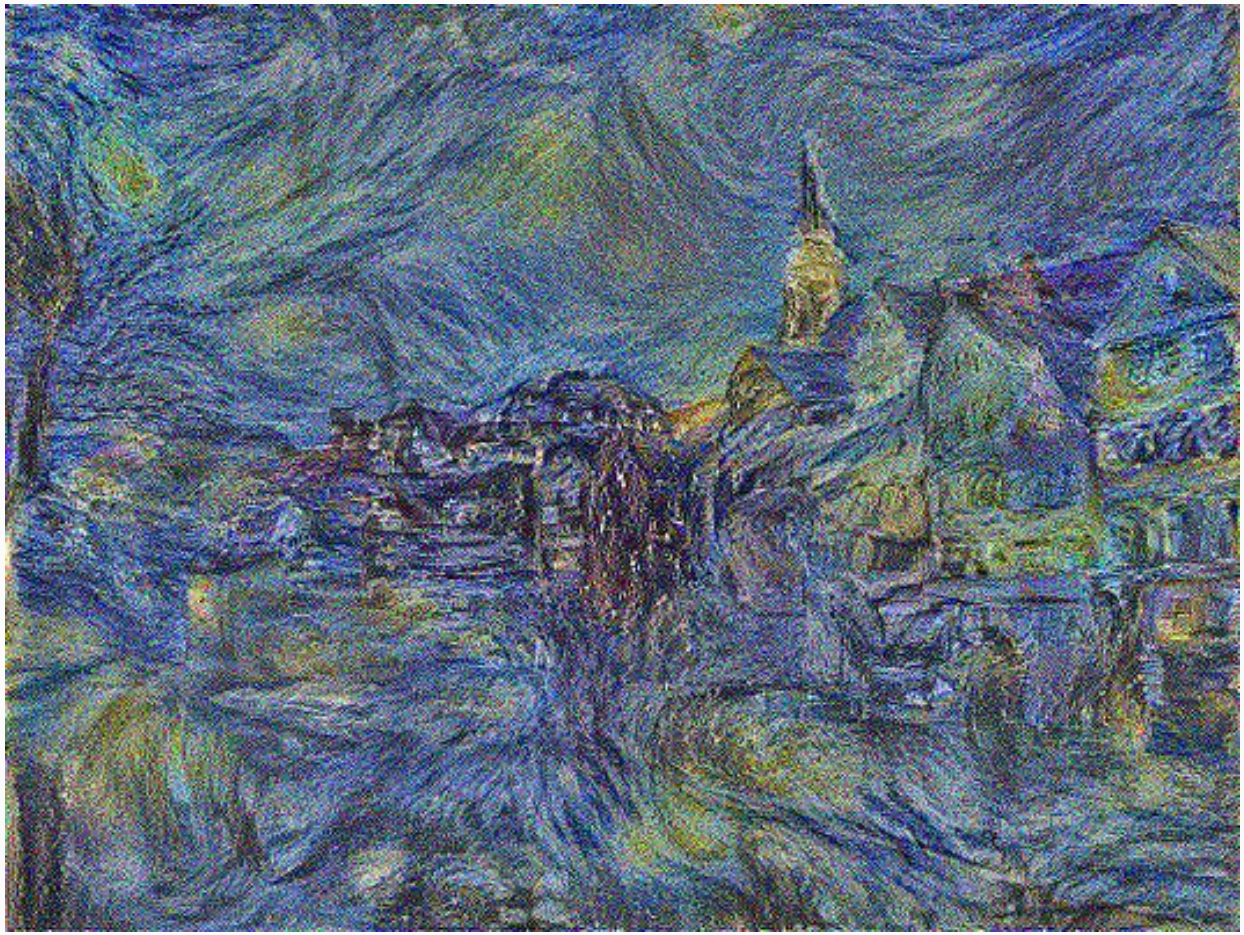
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



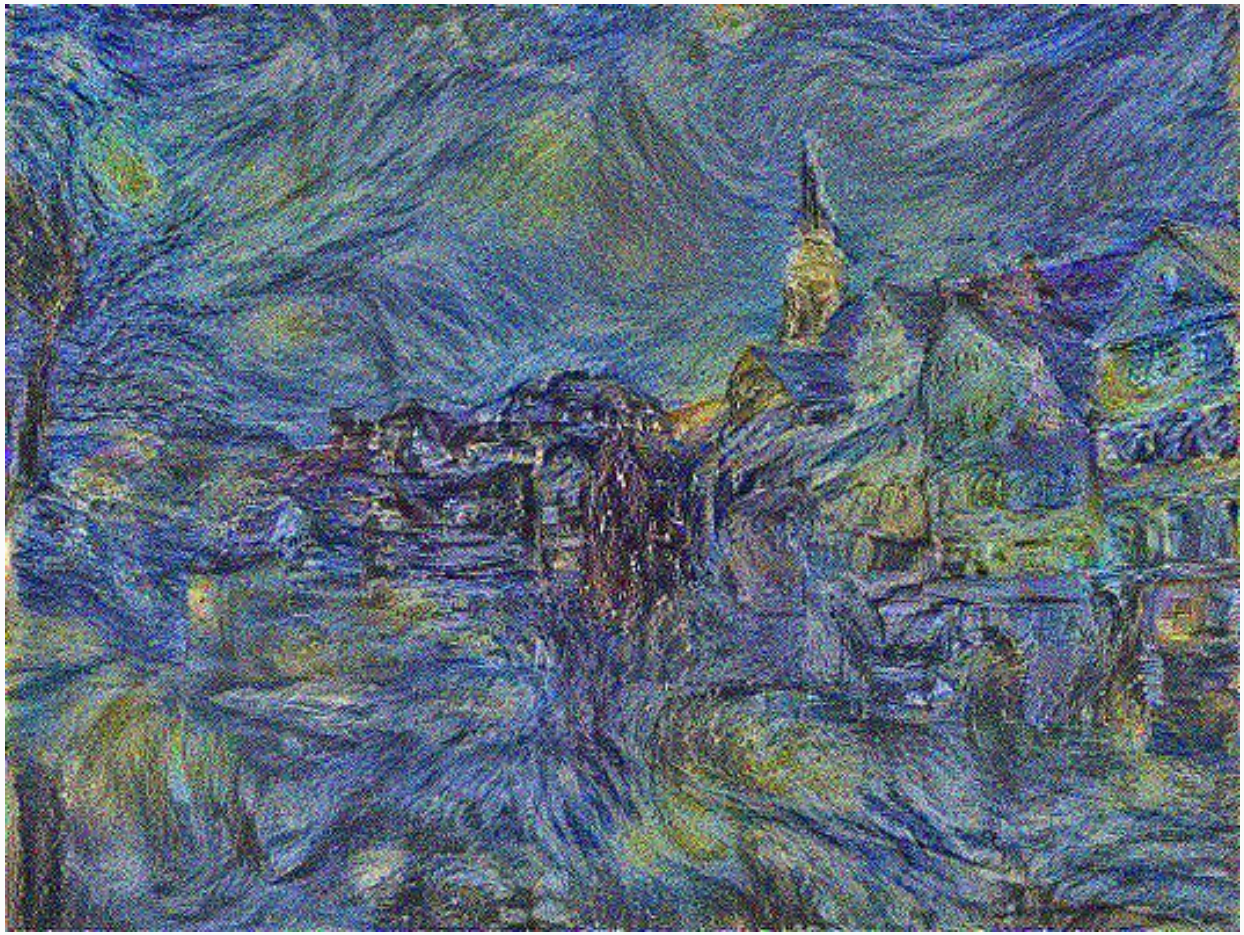
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



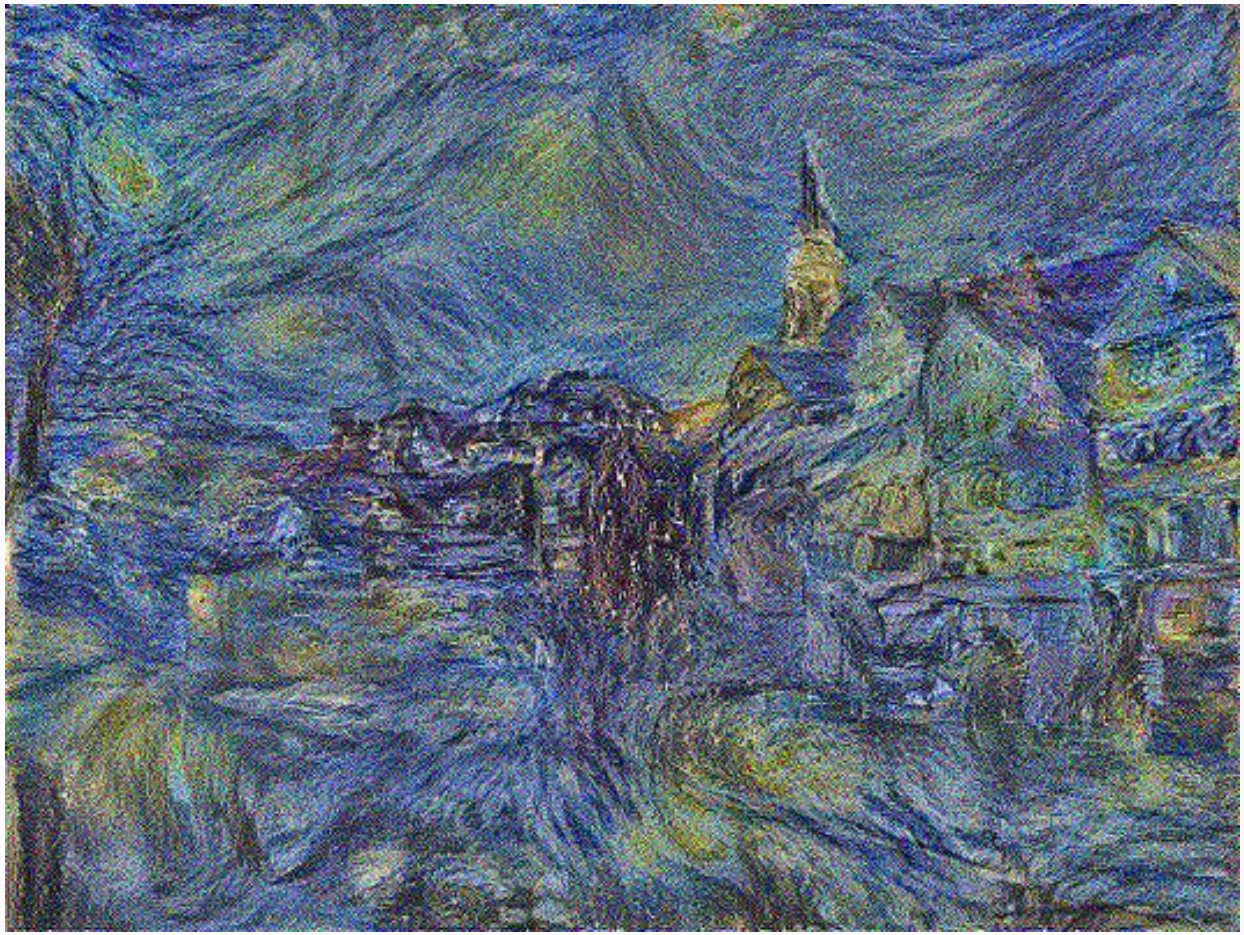
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$

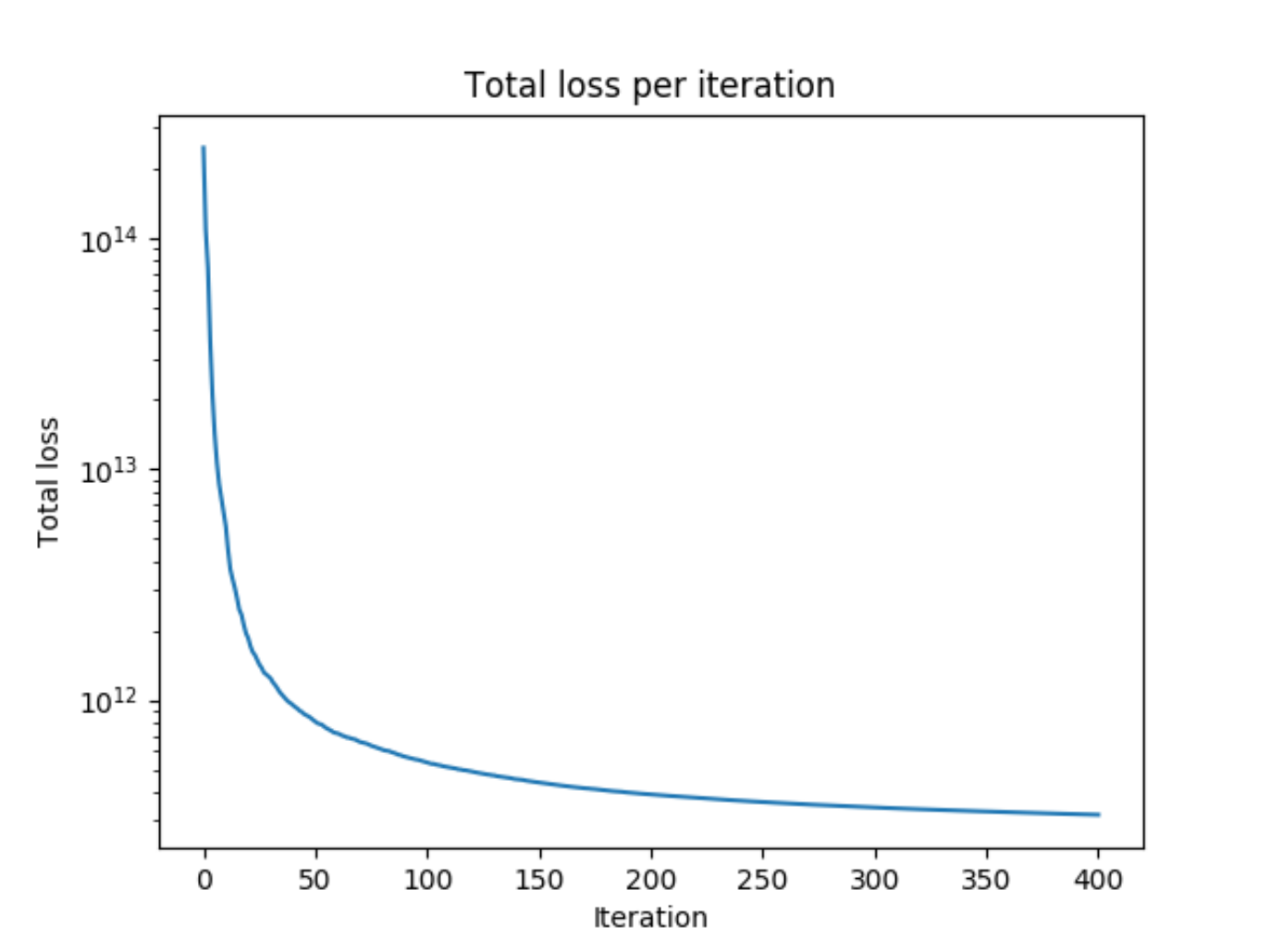


Experiments: Changes per 20 iterations

End of animation

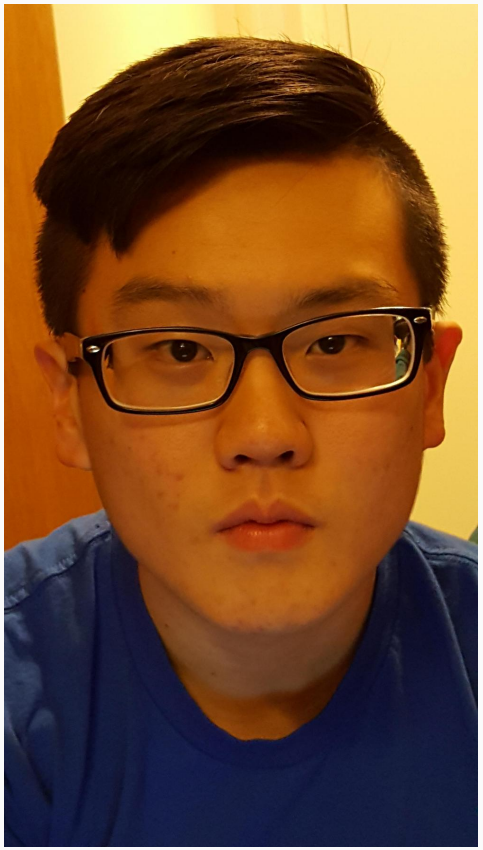
Experiments: Changes per 20 iterations

$c = 1 \qquad s = 10^4 \qquad v = 0$



Experiments: Multiple style images

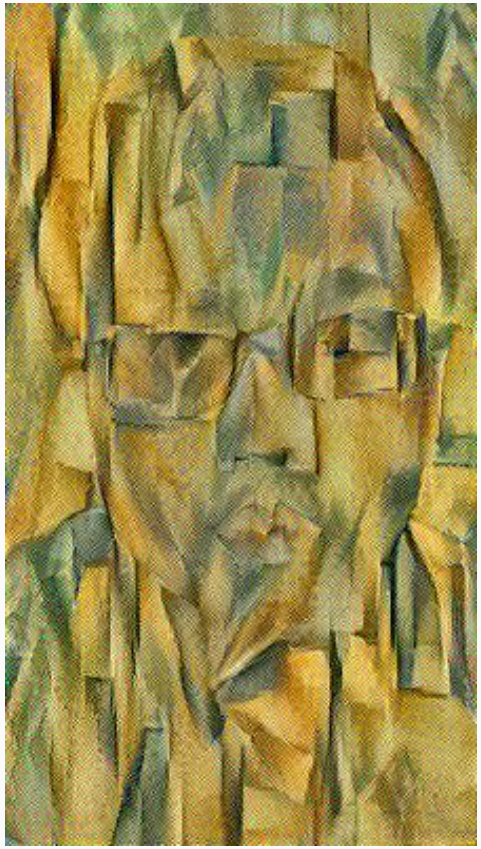
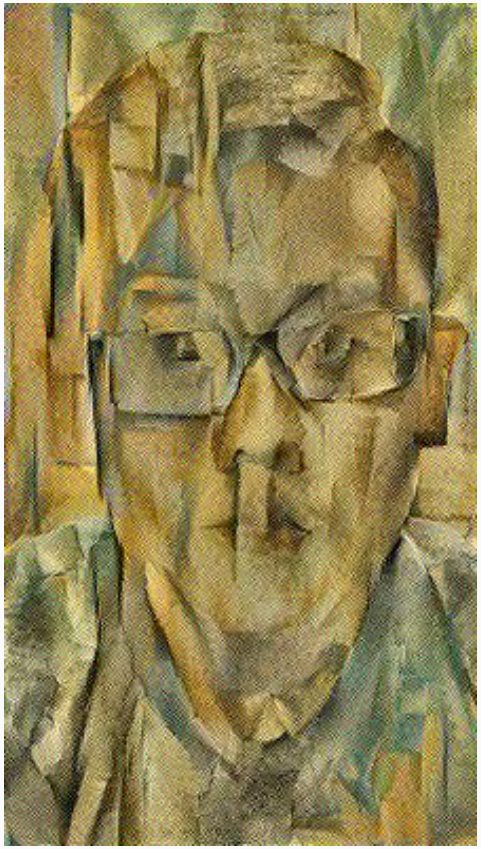
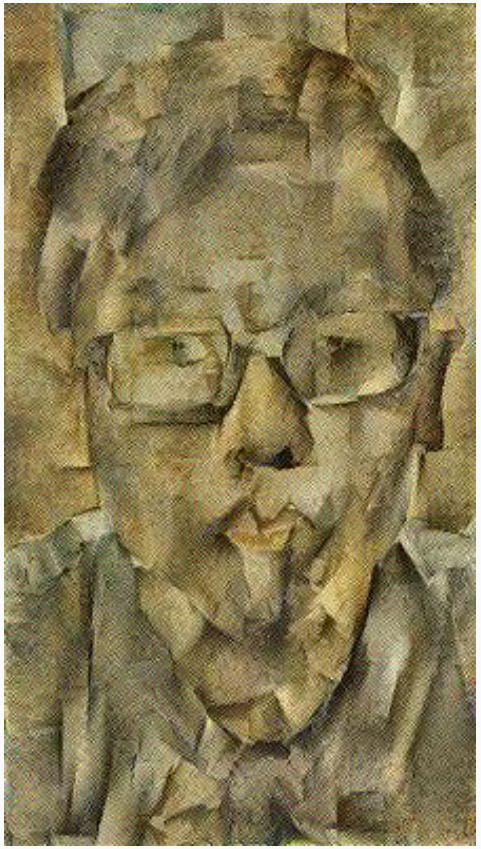
$c = 1 \qquad s = 10^4 \qquad v = 0$



+

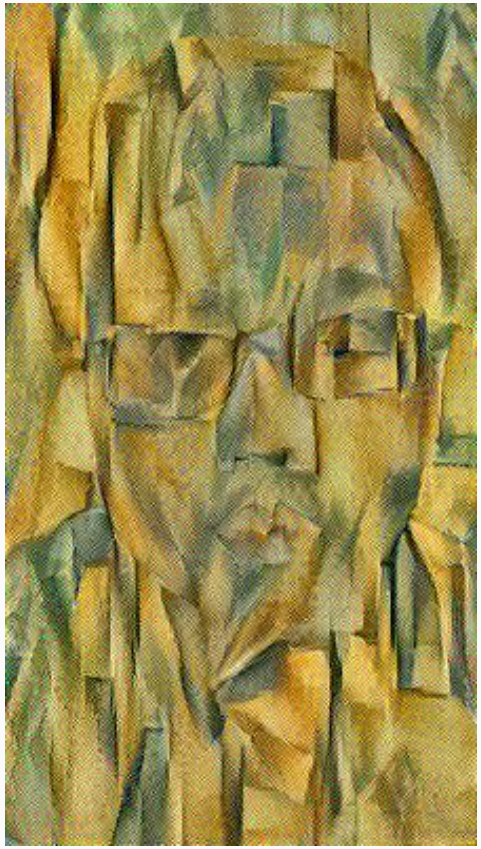
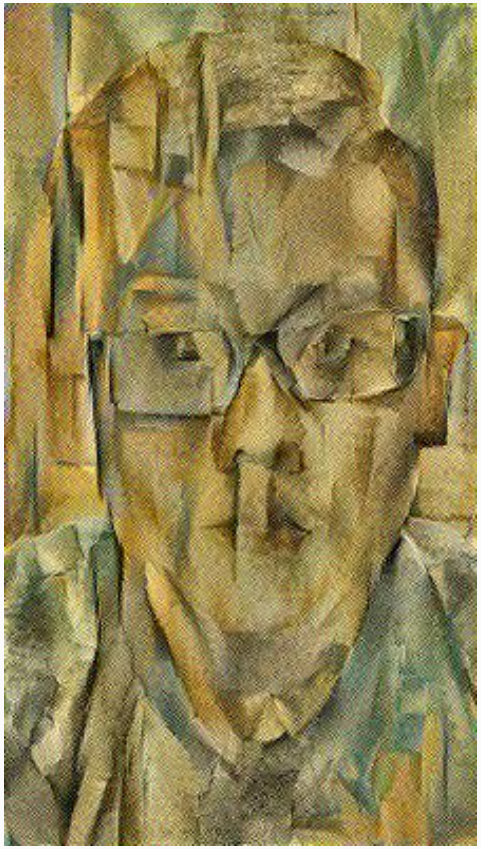
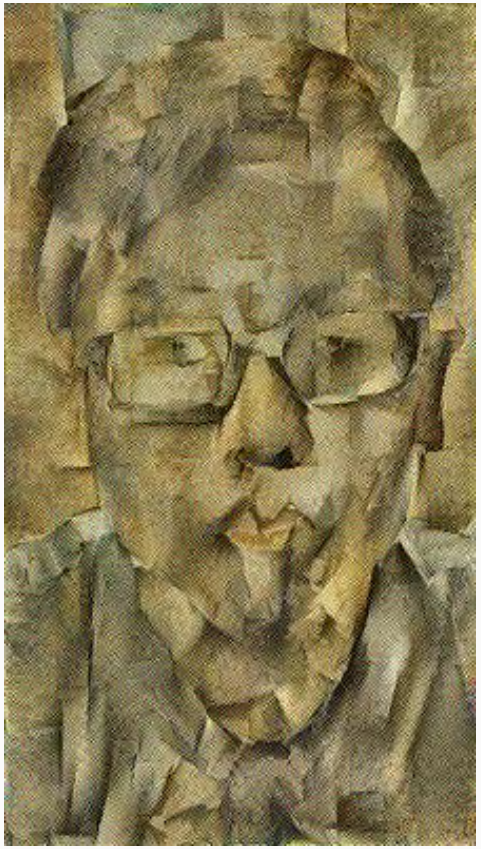
Experiments: Multiple style images

$c = 1 \qquad s = 10^4 \qquad v = 0$

Experiments: Multiple style images

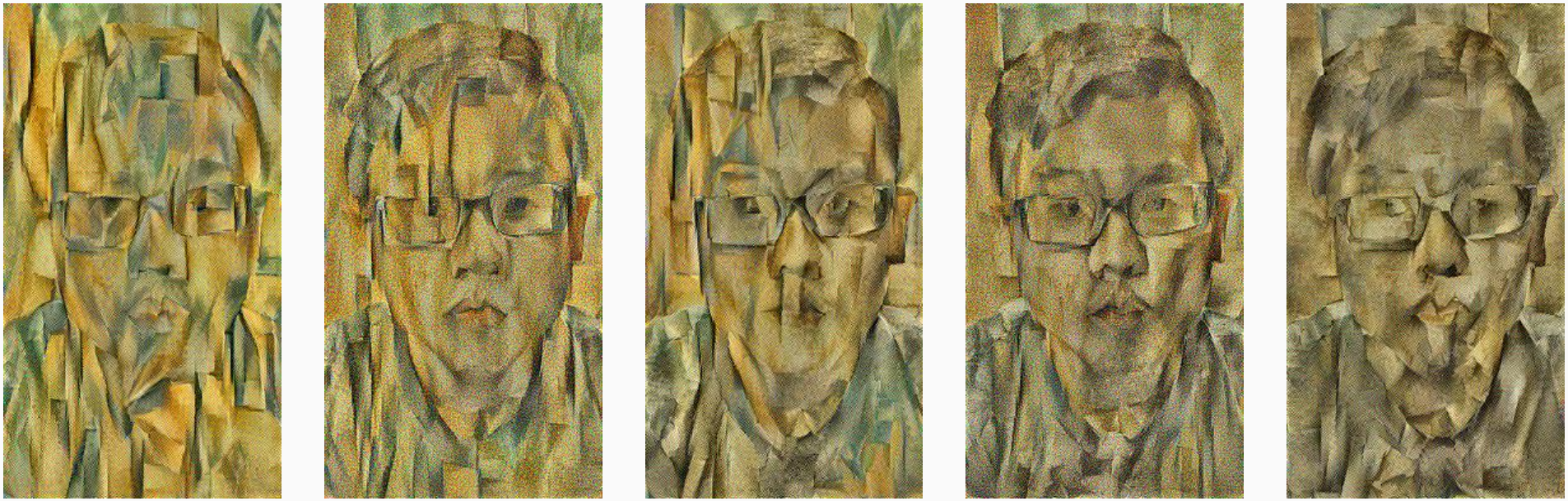
$c = 1 \qquad s = 10^4 \qquad v = 0$

2 out of 3 people voted for the multi-style image!

Experiments: Multiple style images

$c = 1 \qquad s = 10^4 \qquad v = 0$



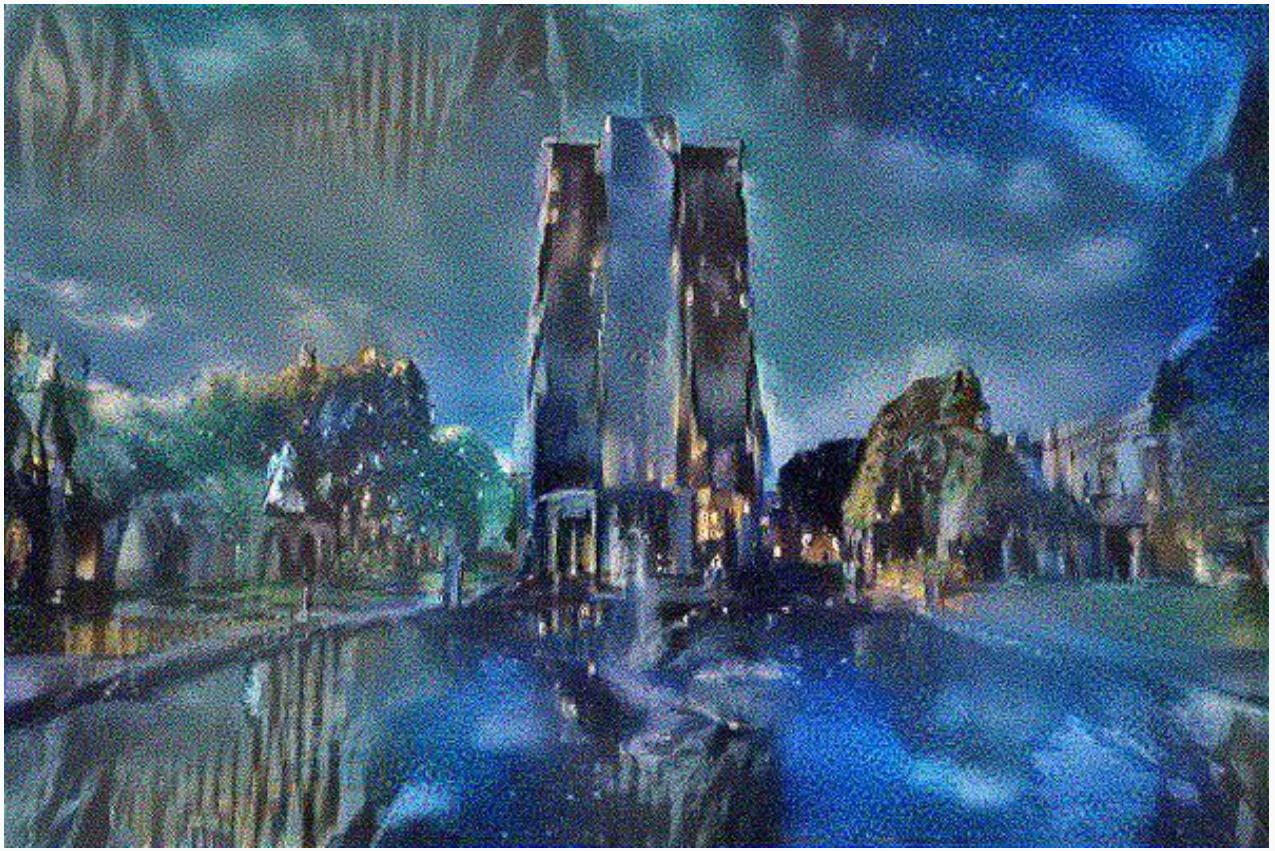
Experiments: Different initializations

$$c = 1 \qquad s = 10^5 \qquad v = 0$$

+

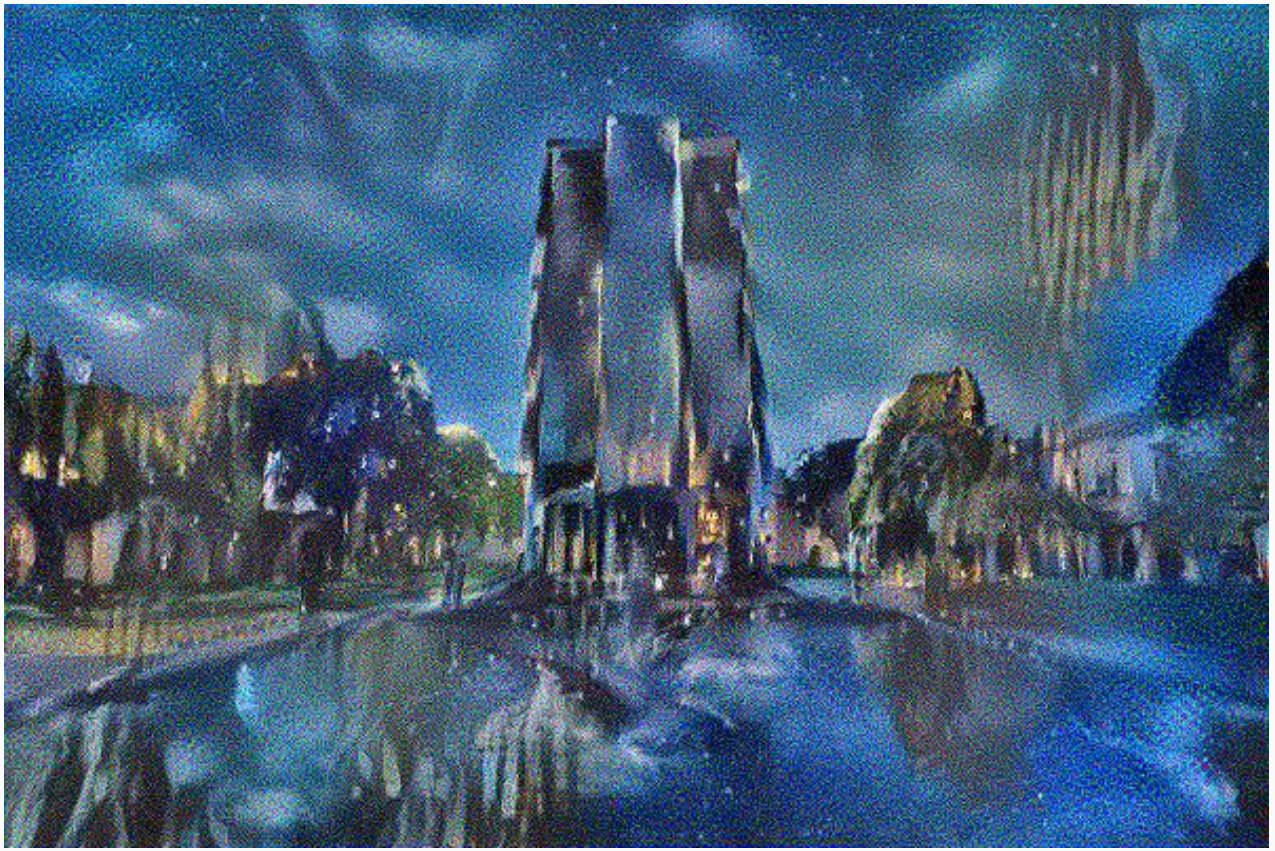
Experiments: Different initializations

$c = 1 \qquad s = 10^5 \qquad v = 0$



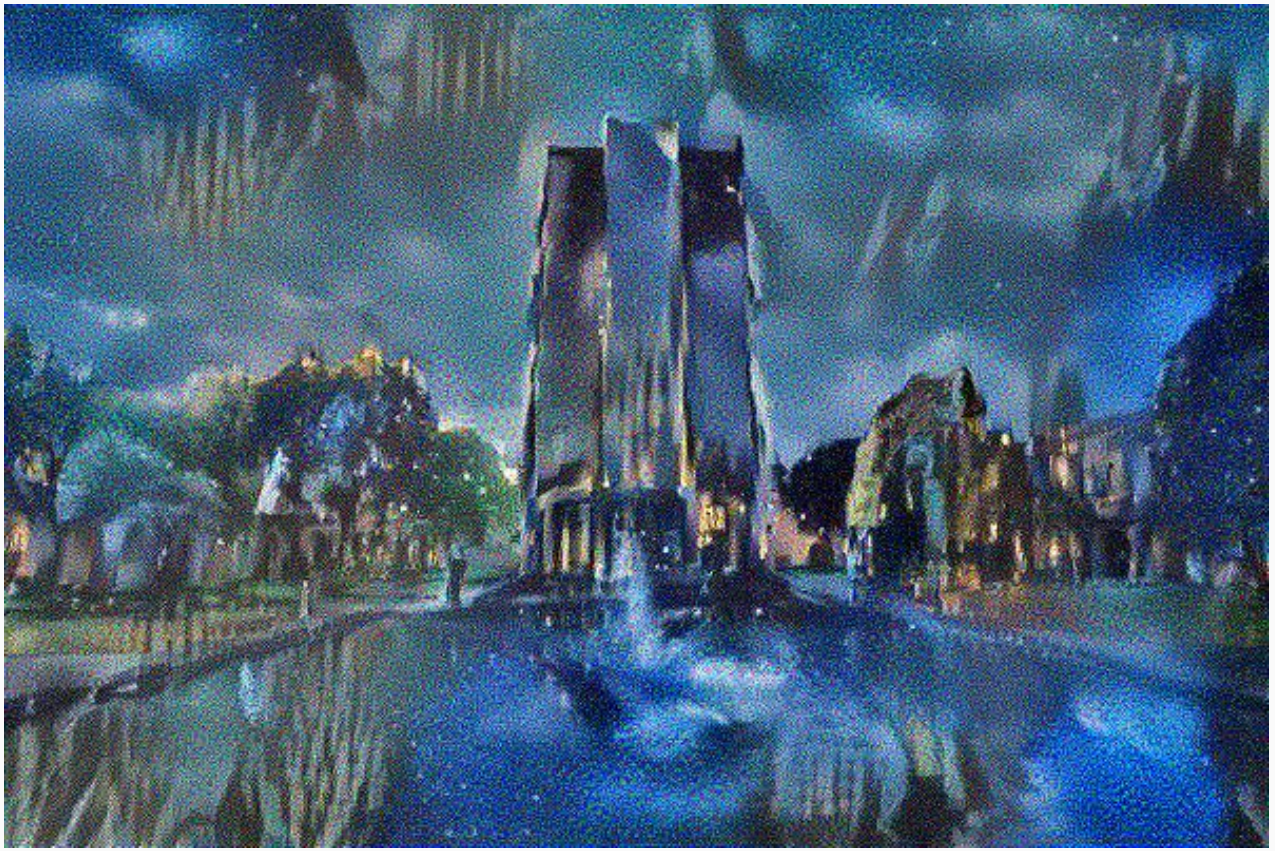
Experiments: Different initializations

$c = 1 \qquad s = 10^5 \qquad v = 0$



Experiments: Different initializations

$c = 1 \qquad s = 10^5 \qquad v = 0$



Experiments: Miscellaneous examples

$c = 1 \qquad s = 10^4 \qquad v = 0$

Experiments: Miscellaneous examples

$c = 1 \qquad s = 10^4 \qquad v = 0$

Limitations

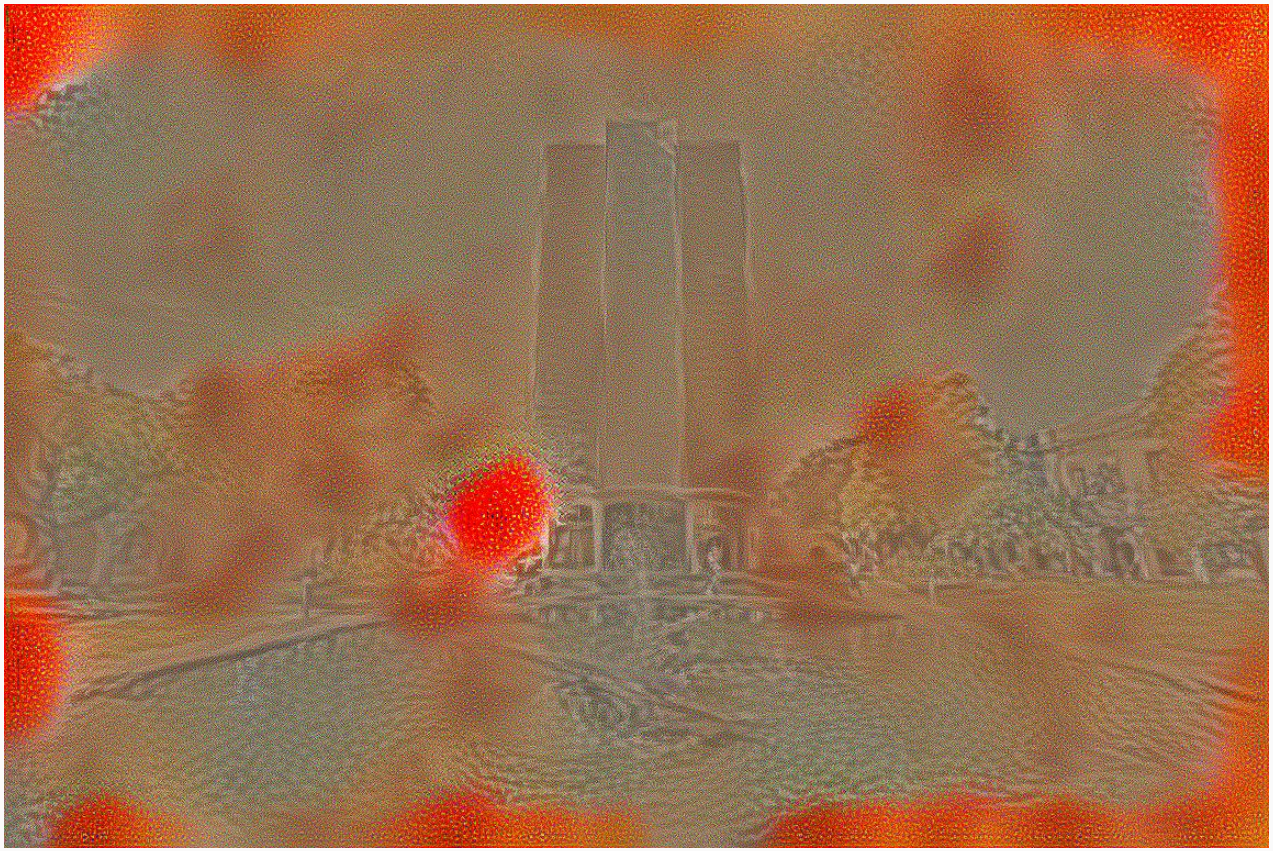
Limitations: Color & Edges

$$c = 1 \qquad s = 10^4 \qquad v = 0$$

$+$

Limitations: Color & Edges

$c = 1 \qquad s = 10^4 \qquad v = 0$



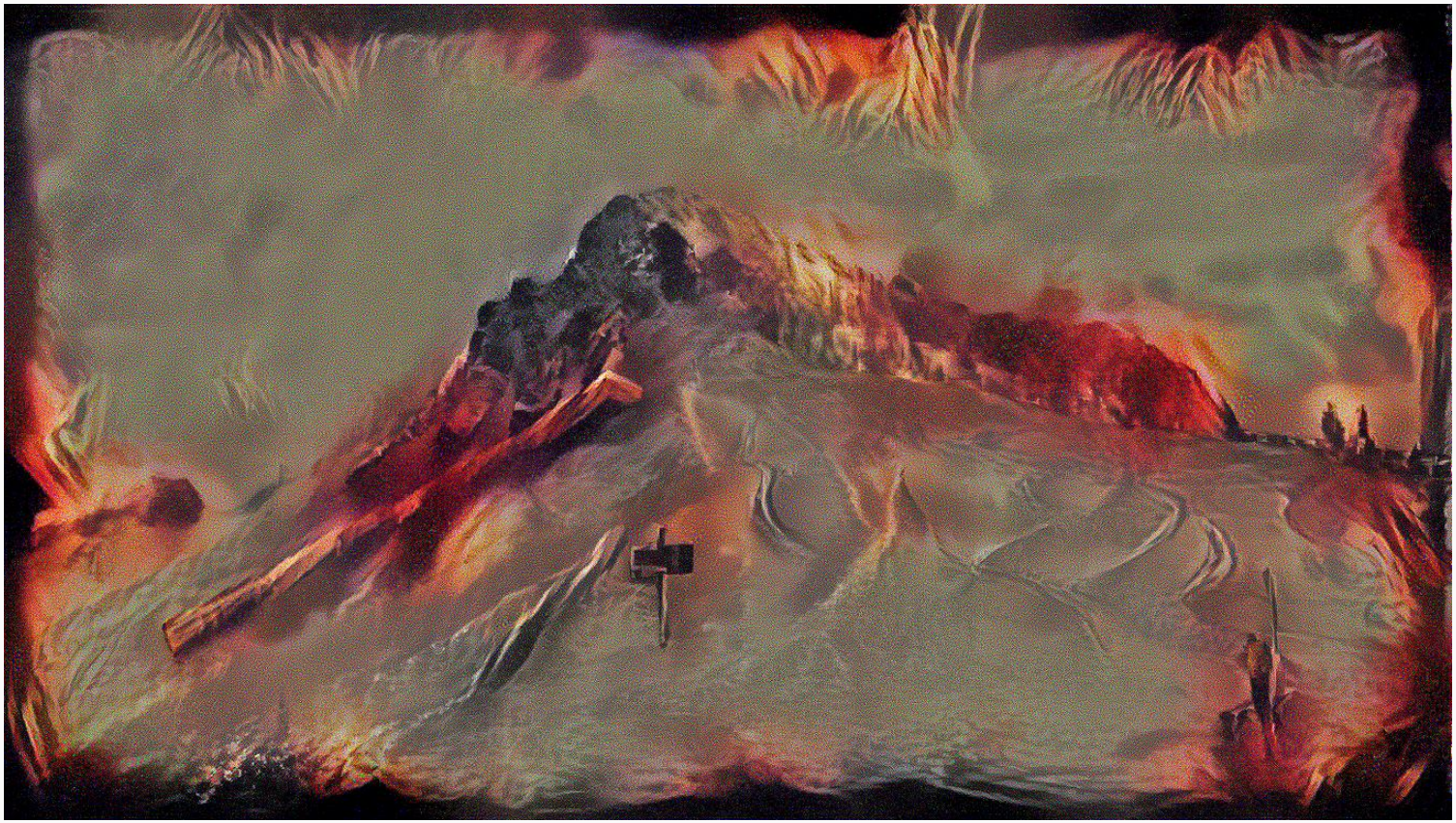
Limitations: Color & Edges

$$c = 1 \qquad s = 10^4 \qquad v = 0$$

+

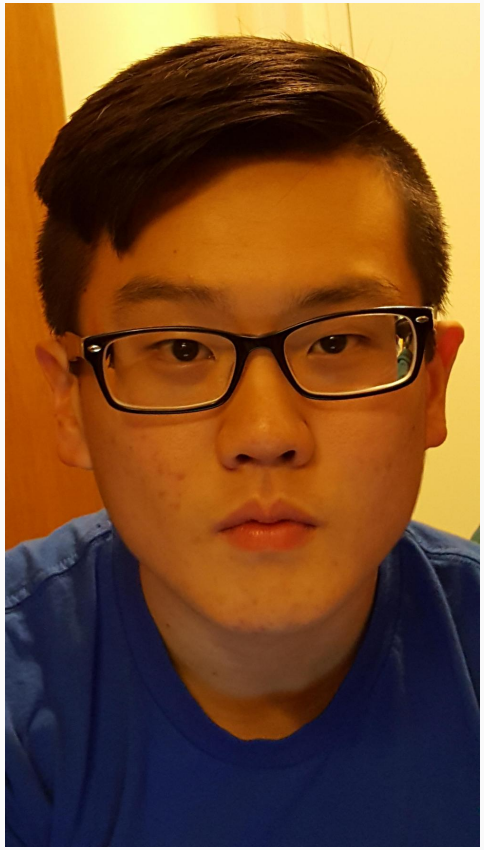
Limitations: Color & Edges

$c = 1 \qquad s = 10^4 \qquad v = 0$



Limitations: Noise

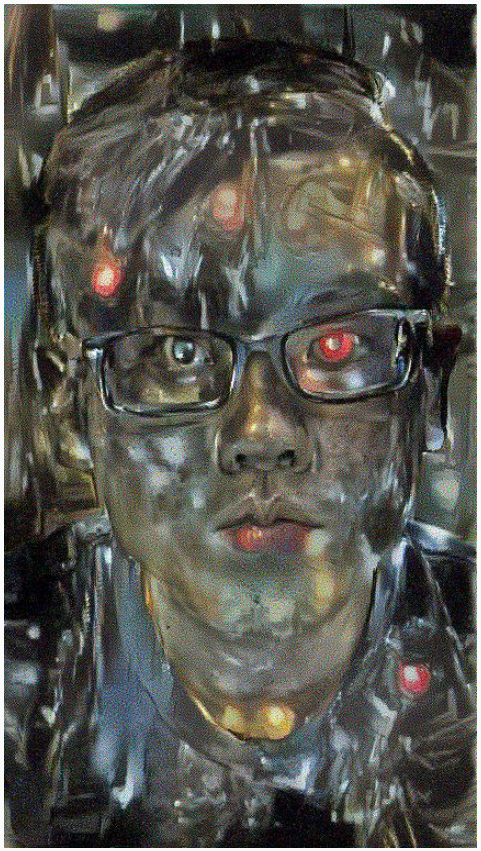
$$c = 1 \qquad s = 10^5 \qquad v = 0$$



$+$

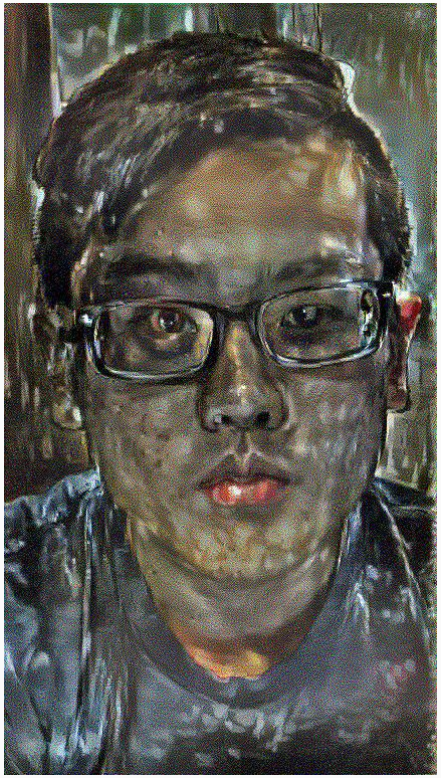
Limitations: Noise

$c = 1 \qquad s = 10^5 \qquad v = 0$

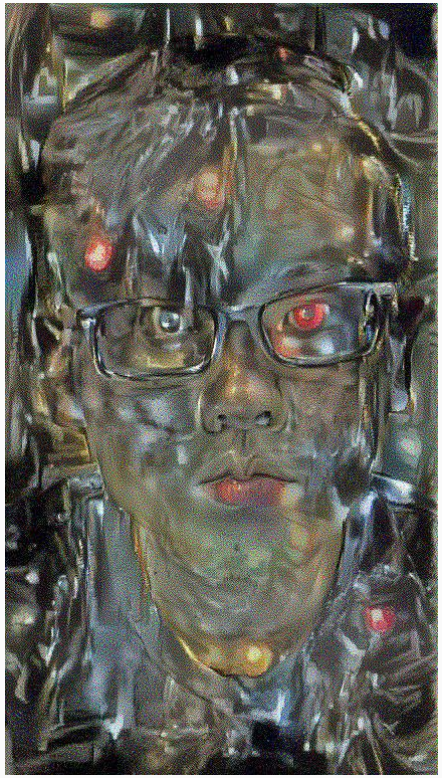


Limitations: Noise

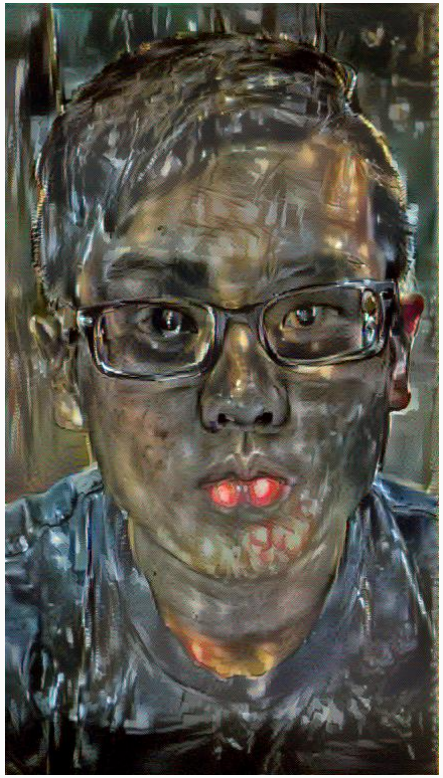
$c = 1, s = 10^4, v = 0$



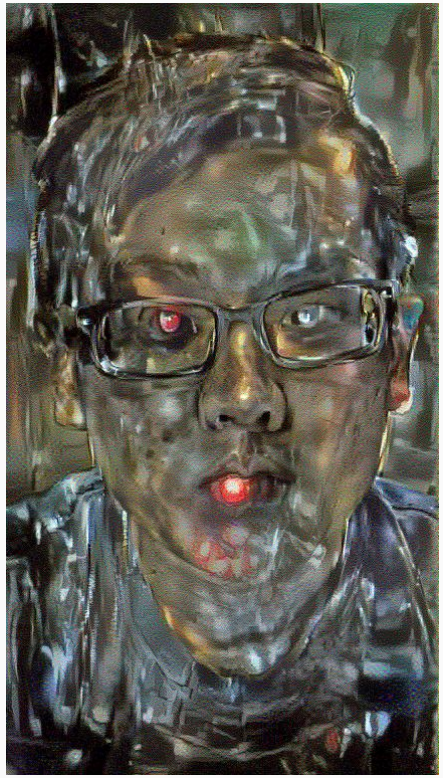
$c = 1, s = 10^5, v = 0$



$c = 1, s = 3 \cdot 10^4, v = 5$



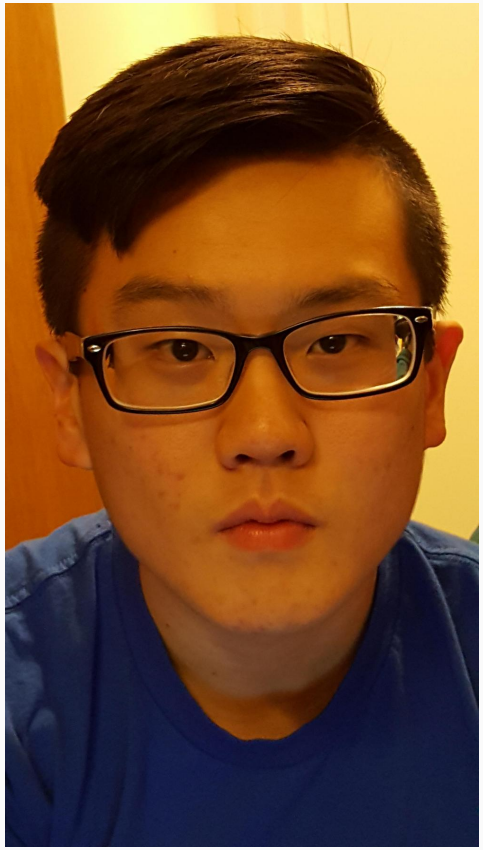
$c = 1, s = 5 \cdot 10^4, v = 1$



Different parameters did not give any better results.

Limitations: Faces

$$c = 1 \qquad s = 10^4 \qquad v = 0$$



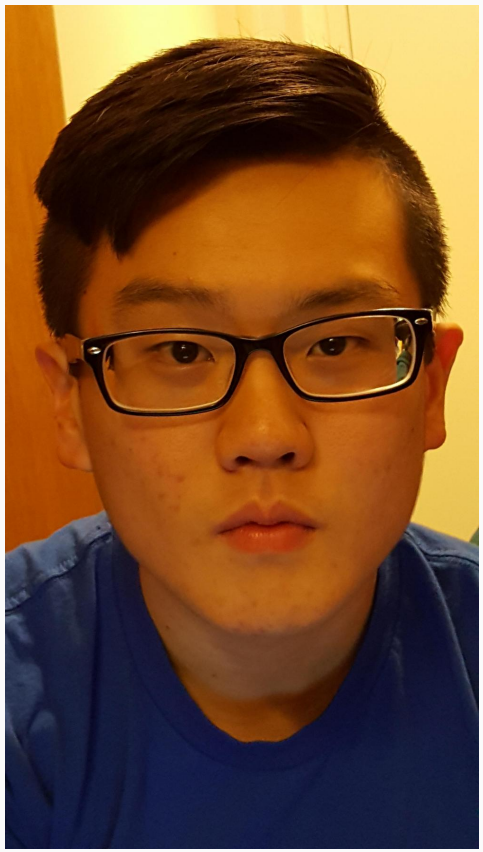
$+$

Limitations: Faces

$c = 1 \qquad s = 10^4 \qquad v = 0$

Limitations: Incompatible images

$c = 1 \qquad s = 10^4 \qquad v = 0$



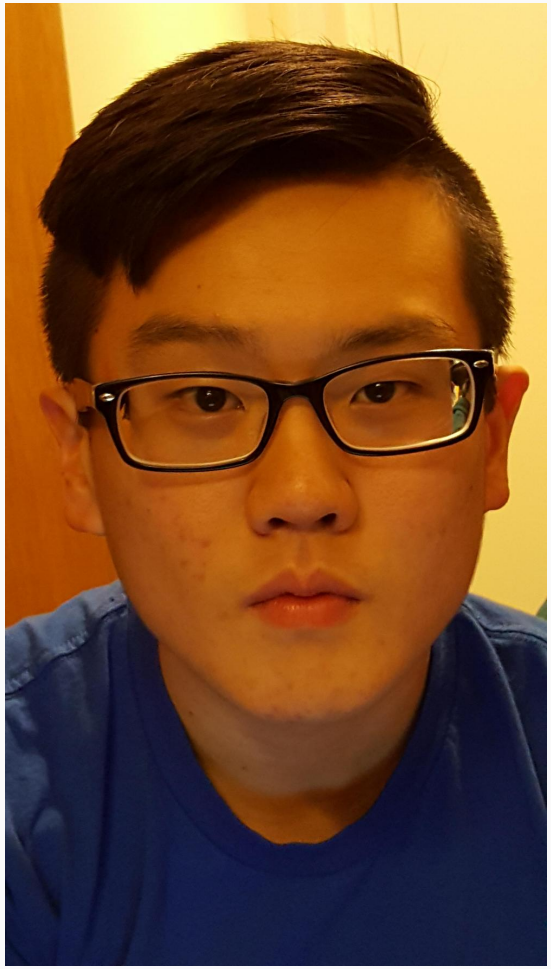
+

Limitations: Incompatible images

$c = 1 \qquad s = 10^4 \qquad v = 0$

Limitations: Fine stylistic details

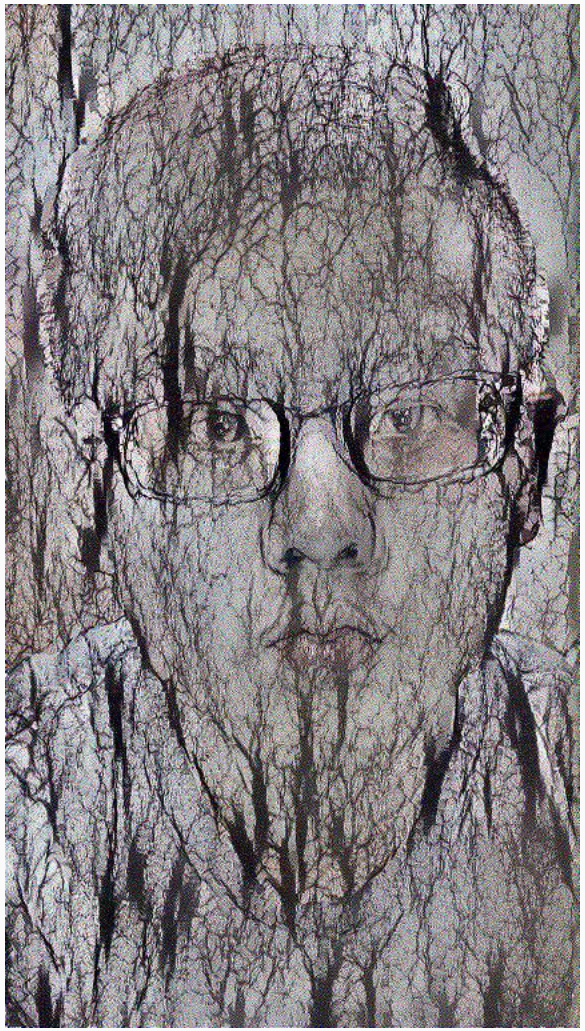
$$c = 1 \qquad s = 10^4 \qquad v = 0$$



+

Limitations: Fine stylistic details

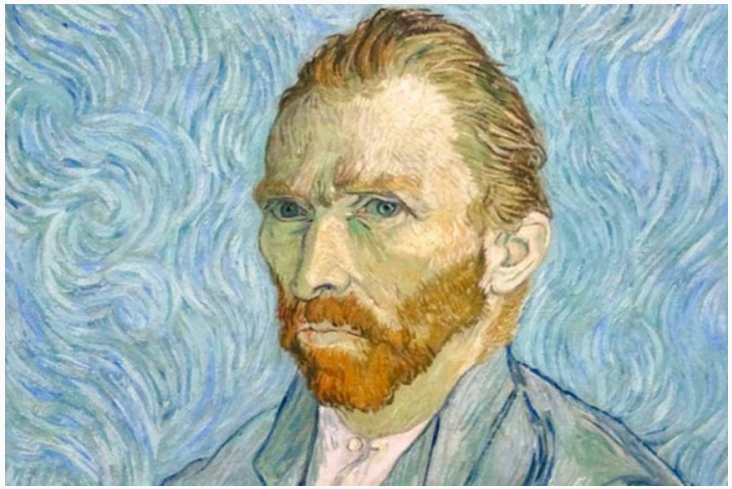
$c = 1 \qquad s = 10^4 \qquad v = 0$



Limitations: Parameter tuning

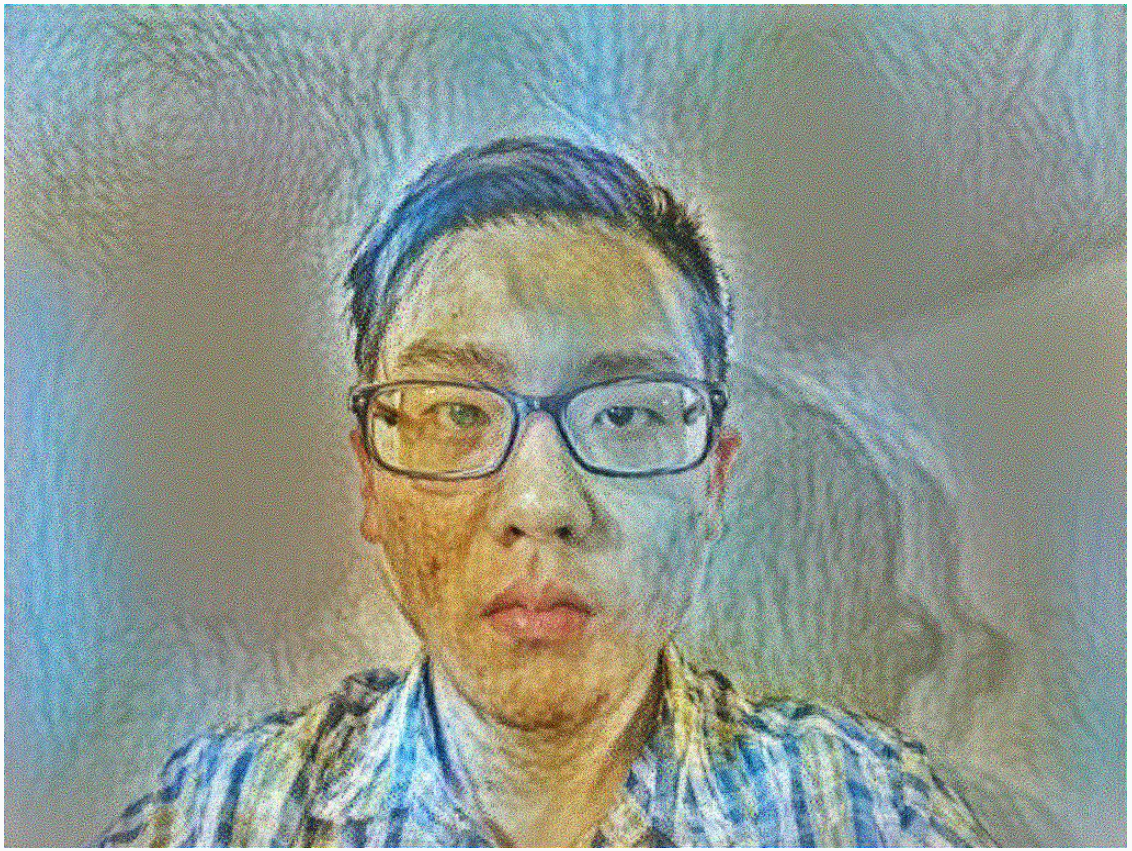
$$c = 1 \qquad s = 10^4 \qquad v = 0$$

$+$



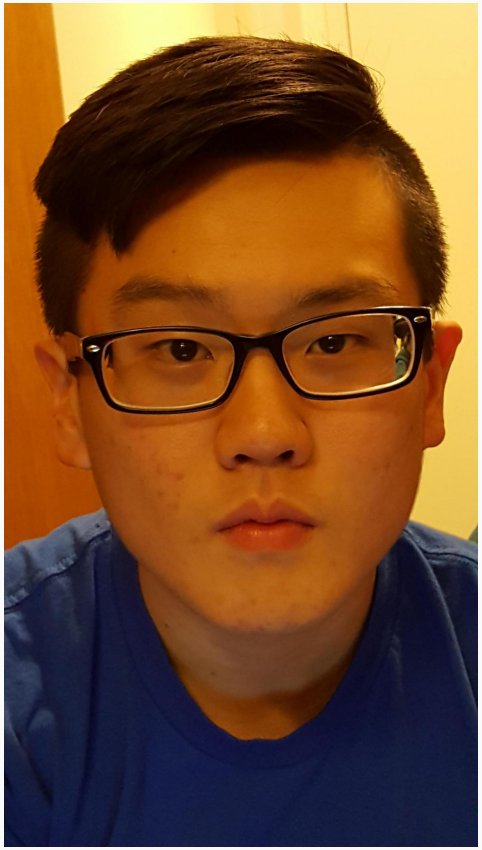
Limitations: Parameter Tuning

$c = 1 \qquad s = 10^4 \qquad v = 0$



Limitations: Parameter tuning

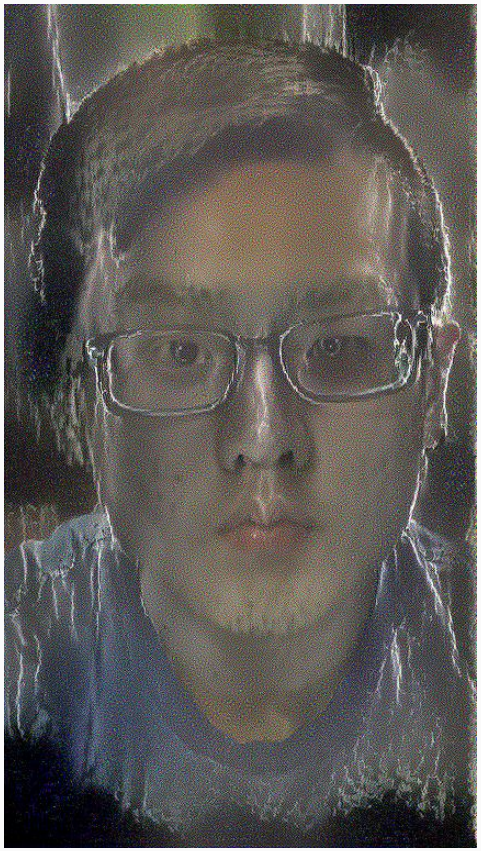
$$c = 1 \qquad s = 10^4 \qquad v = 0$$



+

Limitations: Parameter Tuning

$c = 1 \qquad s = 10^4 \qquad v = 0$



Discussion

Discussion: Advantages

- With the right images, results are great
- Parameters hold across the same image size
- Multiple style images generalize well

Discussion: Disadvantages

- Style images that are not uniform work poorly
- Faces work poorly
- Strong colors dominate, especially at the edges
- Parameters differ across image sizes

Discussion: Disadvantages

- Iteration times scale superlinearly with image sizes
- Shapes are not generally preserved
- Photorealism works... at a distance

Discussion: Solutions

- *Deep Style Photo Transfer* by Luan et al.
- This approach only works well with art
- It does not carry over well to photographs
  - Photorealism not something that can be defined
  - However, there is a certain quality to the colors and structures
- Fix this issue by preserving the affinity of local patches.

Discussion: Solutions

- *Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses* by Risser et al.
- Unstable and requires parameter tuning
- Fix this issue by preserving histogram of activations.

Next Steps

Next Steps

- **Realistic:** Try more examples to justify every claim we made
- **Hopeful:** Implement histogram losses
- **Stretch:** Implement photorealism

Questions?