



Style and Content

By: Kevin Yu
Ph.D. in Computer Engineering



Background of Style and Content

Started off with letters and faces (W. T. Freeman, J. B. Tenenbaum 1996)

Font and the character separation

Face pose and identity separation

Training fonts	synthetic	actual
A A A A A	A A	A A
B B B B B	B B	B B
C C C C C	C C	C C
D D D D D	D D	D D
E E E E E	E E	E E
F F F F F	F F	F F
G G G G G	G G	G G
H H H H H	H H	H H
I I I I I	I I	I I

Figure 6: Example of style extrapolation in the domain of typography. Training data included upper and lower case alphabets, and digits 0-9.

basis warps	
0 0 0 0 0	0 0
1 1 1 1 1	1 1
2 2 2 2 2	2 2
3 3 3 3 3	3 3
4 4 4 4 4	4 4
5 5 5 5 5	5 5
6 6 6 6 6	6 6
7 7 7 7 7	7 7
8 8 8 8 8	8 8
9 9 9 9 9	9 9

Figure 7: Warp bases.

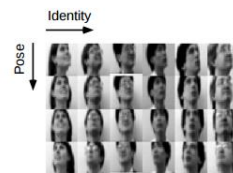


Figure 8: A subset of the face images used for pose classification.

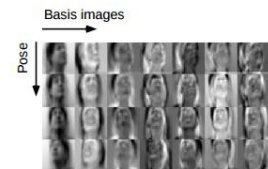


Figure 9: A subset of the pose-specific basis faces. Note that in the bilinear model, each basis face plays the same role across poses.

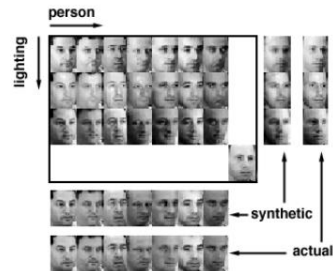


Figure 10: An example of translation.

High-level Style and Content Manipulation

Goal

Transfer style from one image to another image

Texture transformation : low-level == Style transfer : high-level

Edge detectors, etc. find low-level features

CNN's are used for high-level features

A Neural Algorithm of Artistic Style

Deep image representation

VGG Networks

Content representation

Training a CNN on object recognition

Style representation

Style content is obtained using a Gram matrix

Style transfer

Style Transfer

The content of A is kept while the style of each subimage is implemented on the initial A image.

Below is the loss function they minimize.

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

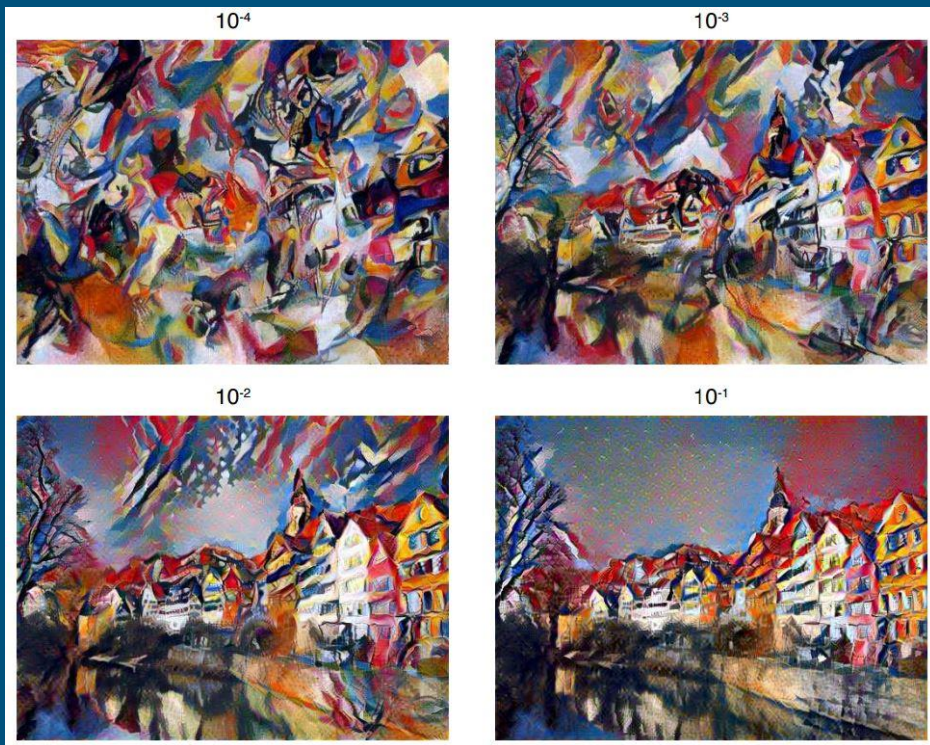


Tradeoffs

Linear combination of loss functions

Resolution is proportional to the speed of the style transfer

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



Initial Conclusion

Content and Style are easily separable using CNN's

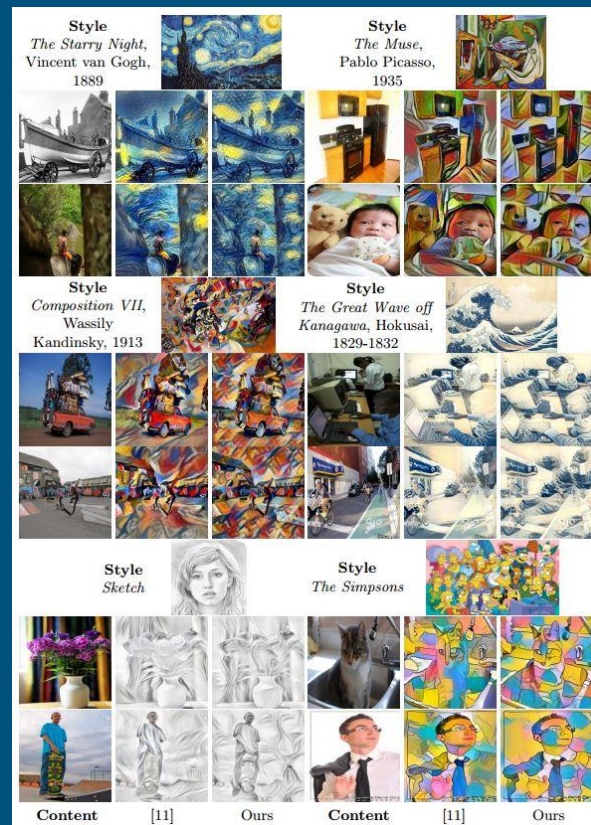
We are able to synthesize an image with the content and style of two separate images

The process is currently slow due to the speed of the algorithm

Using Feed-Forward Image Transformation

Three orders of magnitude faster

Using perceptual loss



Adding Markov Random Fields

Increased quality of images



Conclusion

Improvements on texture and content transformation

Feed-Forward CNN's == Faster synthesis

Markov Random Fields == Higher quality synthesis