Neural Style Transfer

Juan Camilo Fernandez
ML Engineer @LandingAI
jcfernandezp@unal.edu.co

What is Style Transfer?

Content Image

Style Image

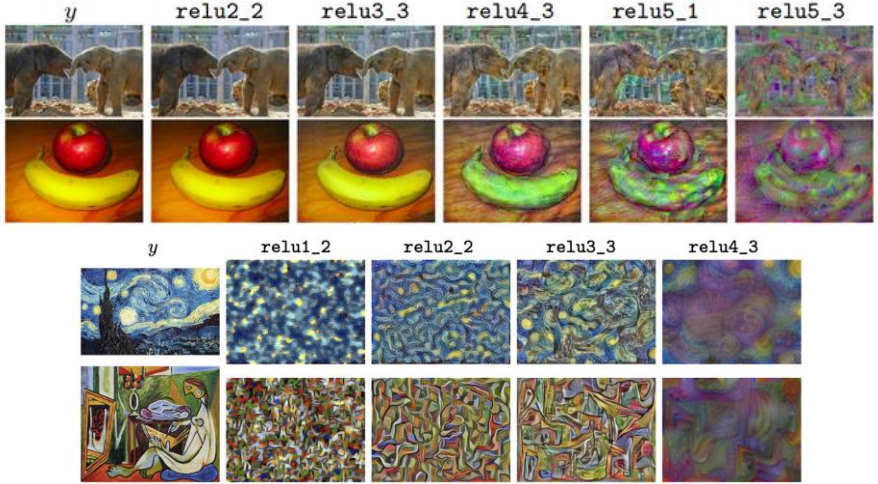
Generated image

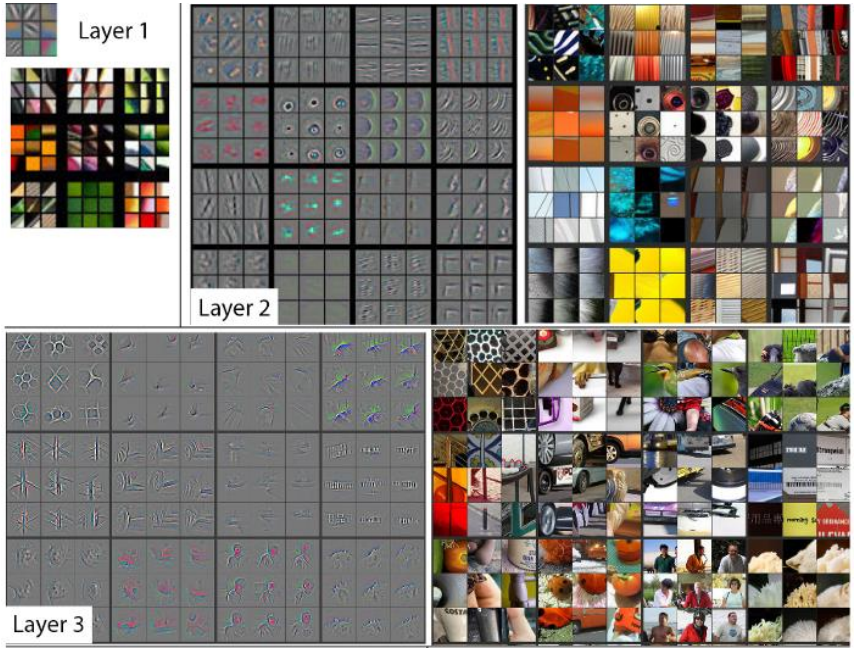
What do we need?

1. A pretty dope pre-trained convolutional neural network
2. A custom (and kinda weird) loss function

Why CNN?

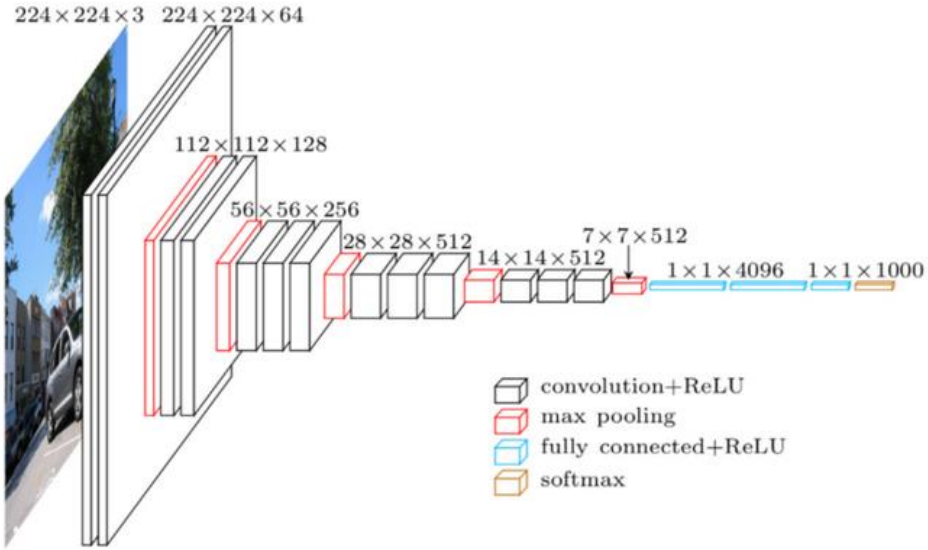
They detect important features in images!

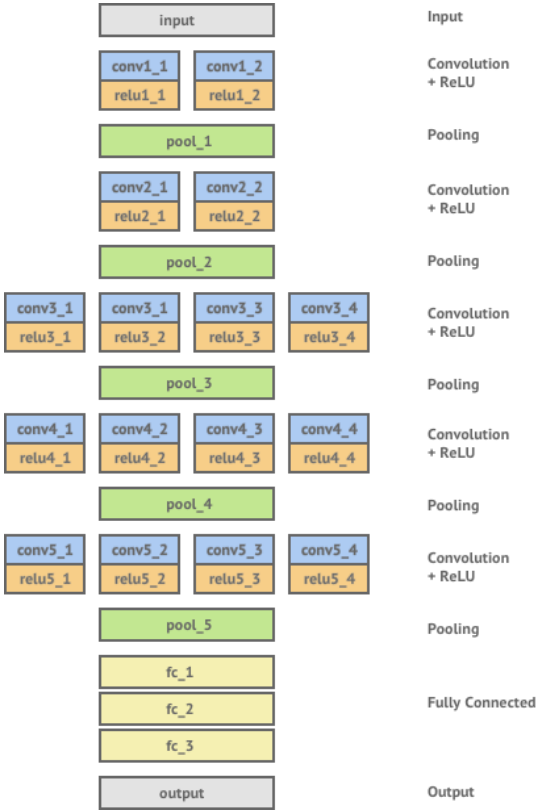




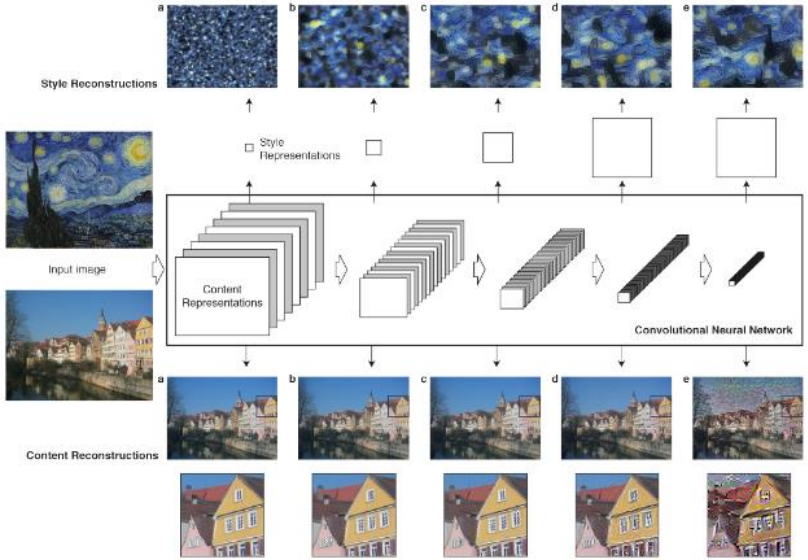
Which CNN?

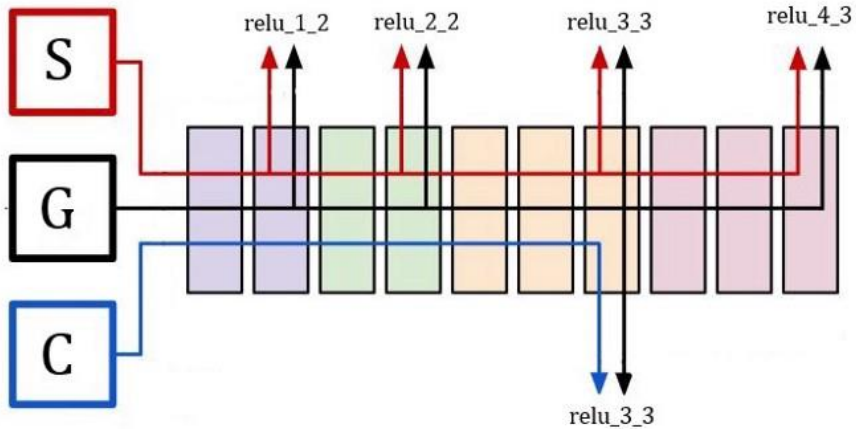
Usually use VGG16 but you can use whatever you want!





How does this work?





And the loss function?

Definimos:

S = Style Image Feature Space
C = Content Image Feature Space
G = Generated Image Feature Space

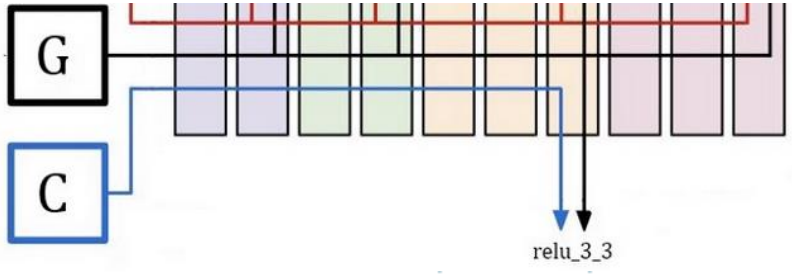
$$L_{total}(S, C, G) = \alpha L_{content}(C, G) + \beta L_{style}(S, G)$$

Sometimes adding a total variational loss smooth the optimization problem:

$$L_{total}(S, C, G) = \alpha L_{content}(C, G) + \beta L_{style}(S, G) + L_{var}(G)$$

Optimize using ADAM or L-HBFS

Content Loss?



$$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix} \quad \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 1 & 0 & 2\end{pmatrix}$$

L is the layer we're extracting features from. Typically chosen as the 3rd layer of VGG.

$$L_{content}(C, G, L) = \frac{1}{2}\sum_{ij}(a[L](C)_{ij} - a[L](G)_{ij})^2$$

$$\left(\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix} - \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 1 & 0 & 2\end{pmatrix}\right)^2 = \frac{1}{2}\sum_{ij}\begin{pmatrix}1 & 1 & 9\\ 9 & 25 & 25\\ 36 & 64 & 49\end{pmatrix} = \frac{219}{2}$$

Style Loss?

$$L_{style}(S, G) = \sum_{l=0}^{L} w_l * L_{GM}(S, G, l)$$

Donde:

L = Number of layers for style.
$\boldsymbol{L_{gm}}$ = Gram matrix loss for layer l

$$L_{GM}(S, G, l) = \frac{1}{4N_l^2 M_l^2} \sum_{ij} (GM[l](S)_{ij} - GM[l](G)_{ij})^2$$

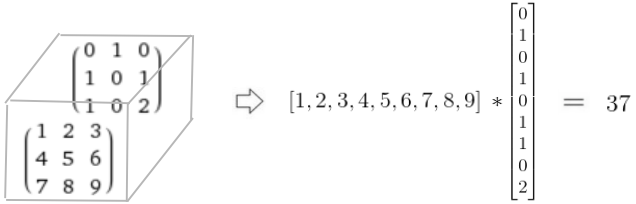
Donde:

GM = Gram matrix of the feature space
$\mathbf{N_l}$ = Number of channels of feature space

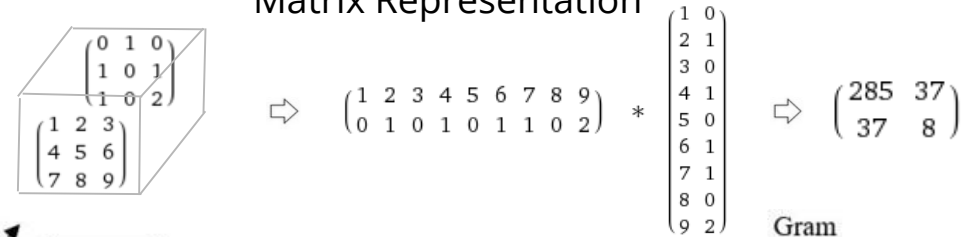
$\mathbf{M_l}$ = Height * Width of the feature

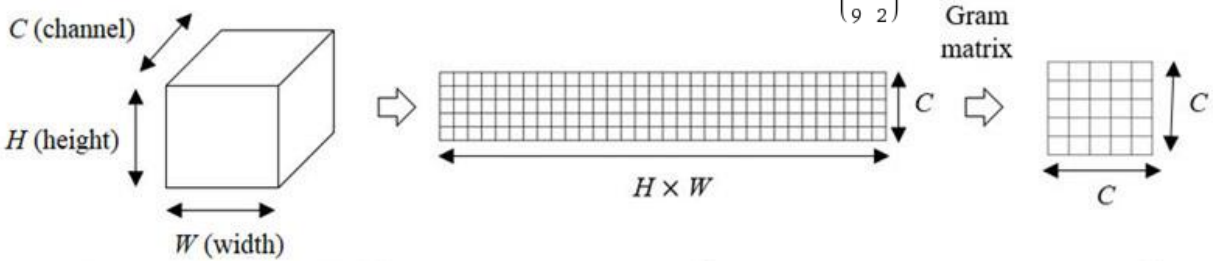
But what the f*** is the gram matrix?

Feature Cross Correlation



Matrix Representation





How this work?

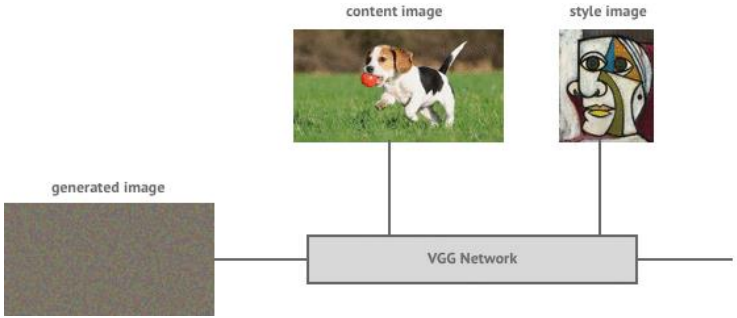

Style target

$$L_{total}(S, C, G) = \alpha L_{content}(C, G) + \beta L_{style}(S, G)$$

$$L_{content}(C, G, L) = \frac{1}{2} \sum_{ij} (a[L](C)_{ij} - a[L](G)_{ij})^2$$

$$L_{style}(S, G) = \sum_{l=0}^{L} w_l * L_{GM}(S, G, l)$$

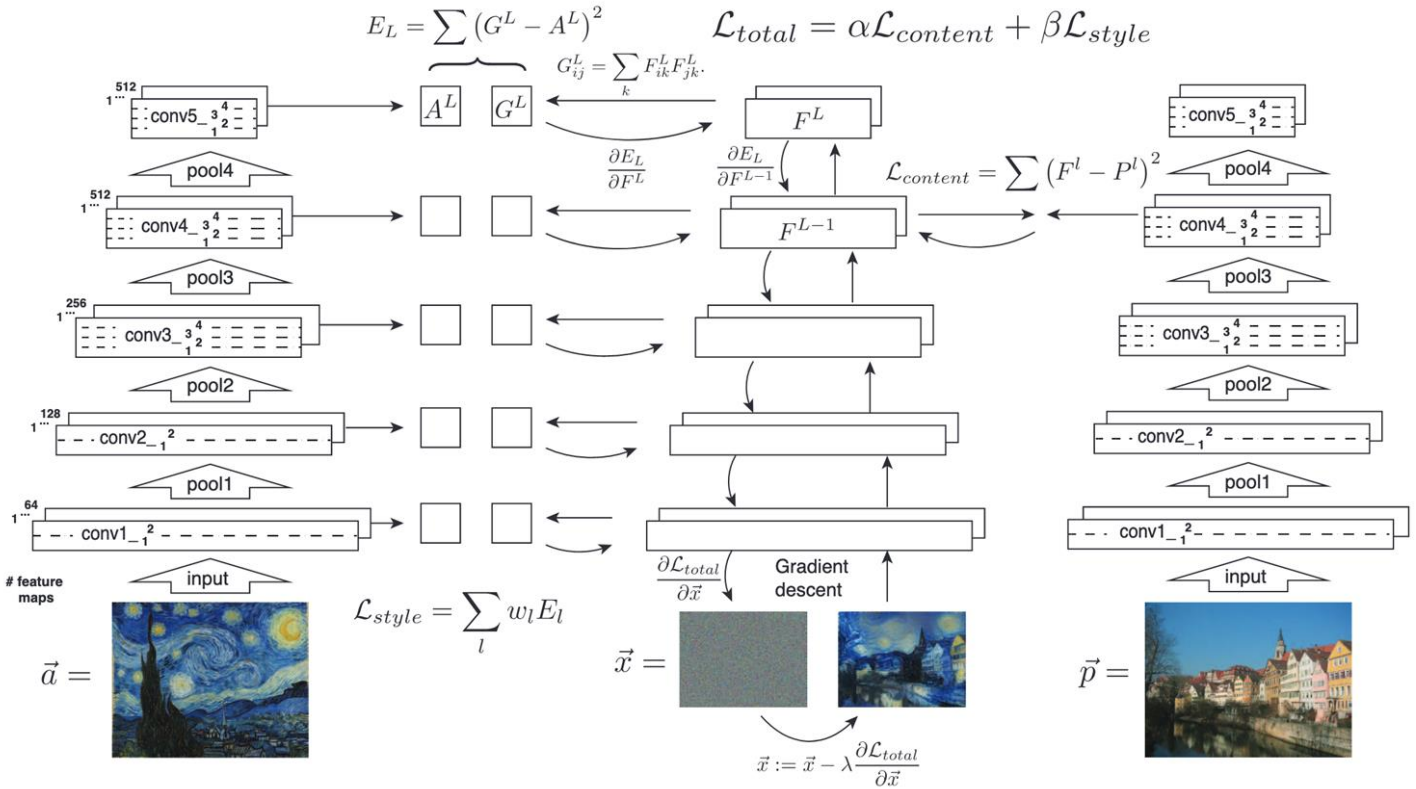
$$L_{GM}(S, G, l) = \frac{1}{4 N_l^2 M_l^2} \sum_{ij} (GM[l](S)_{ij} - GM[l](G)_{ij})^2$$



A Neural Algorithm of Artistic Style by Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.

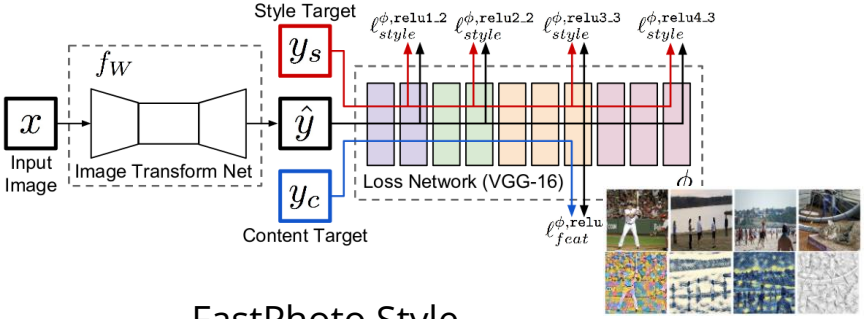
@eliotandres

How's the general framework?

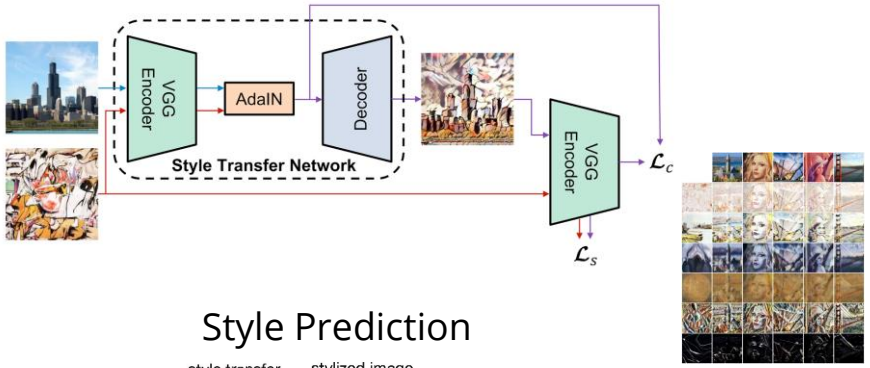


What's the state of the art?

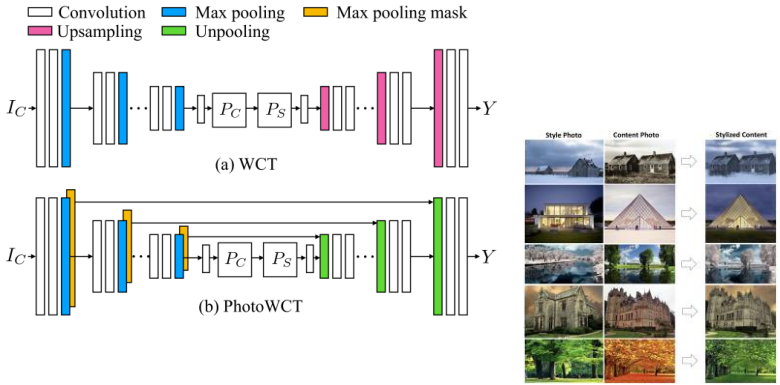
ImageTransform Net



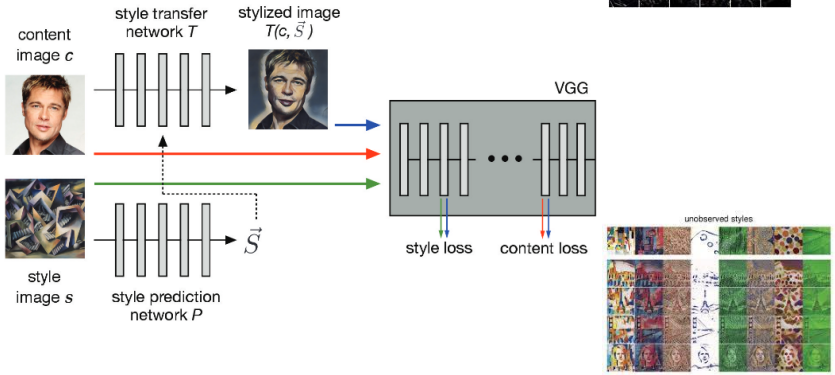
AdaIn



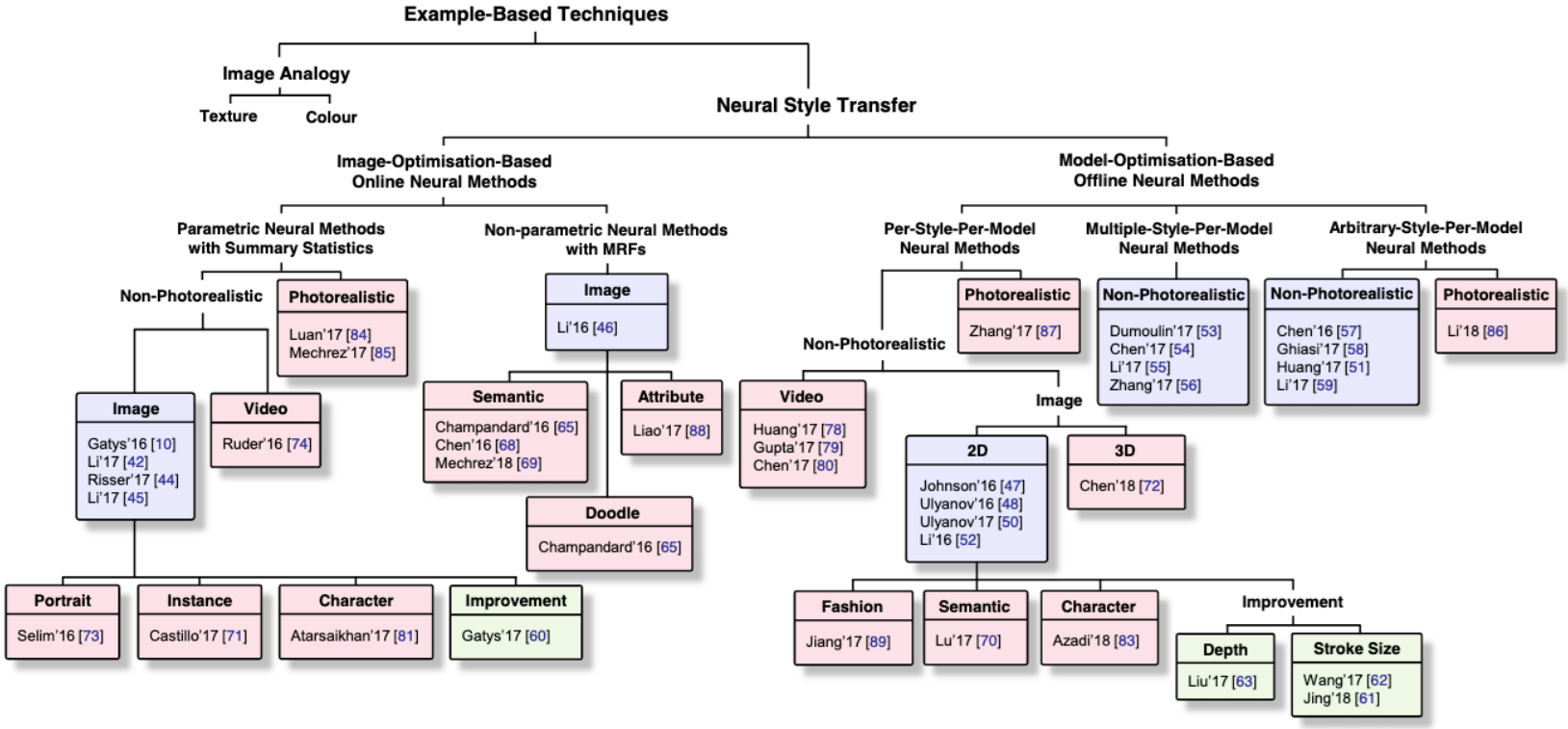
FastPhoto Style



Style Prediction



What are other style transfer branches?



References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pages 2414–2423. IEEE, 2016.

[3] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In European Conference on Computer Vision (pp. 694-711). October 2016, Springer, Cham.

[4] X. Huang, S. Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization, in ICCV 2017.

[5] G. Ghiasi, H. Lee and M. Kudlur, Exploring the structure of a real-time, arbitrary neural artistic stylization network. in British Machine Vision Conference 2017, London, 2017.

[6] Li, Yijun, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang and Jan Kautz. “A Closed-Form Solution to Photorealistic Image Stylization.” ECCV (2018).

[7] Jing, Yongcheng, Yezhou Yang, Zunlei Feng, Jingwen Ye and Mingli Song. “Neural Style Transfer: A Review.” IEEE transactions on visualization and computer graphics (2018)