Universal Style Transfer

Via feature transforms

Presentation Agenda

What is style transfer ?

Previous works

2 works, Defects, Trade offs

Proposed Algorithm

Reconstruction decoder,
WCT (whitening, coloring),
Multi level coarse to fine stylization

Further Applications

Masking, Textures

Our Results

What is “Universal Style Transfer?”



Previous Works

Optimization based method

- Minimizes style & content loss
- Can generalize
- Very slow style transfer

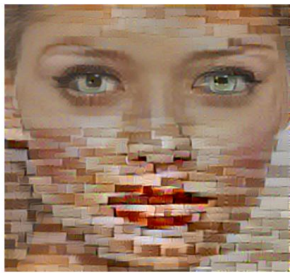
Trained Feed Forward method

- It will adjust mean & variance
- Faster style transfer
- Limited to fixed number of styles

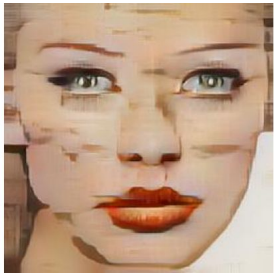
Trade off between Generalization, Quality & Efficiency

Previous Works

Optimization based method

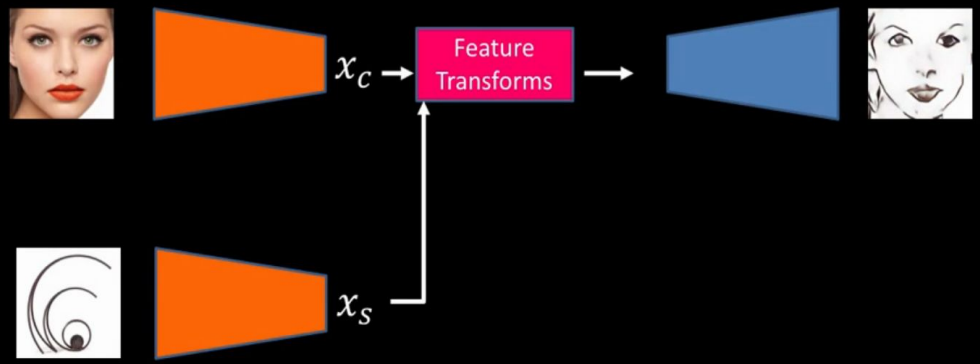
Trained Feed Forward method

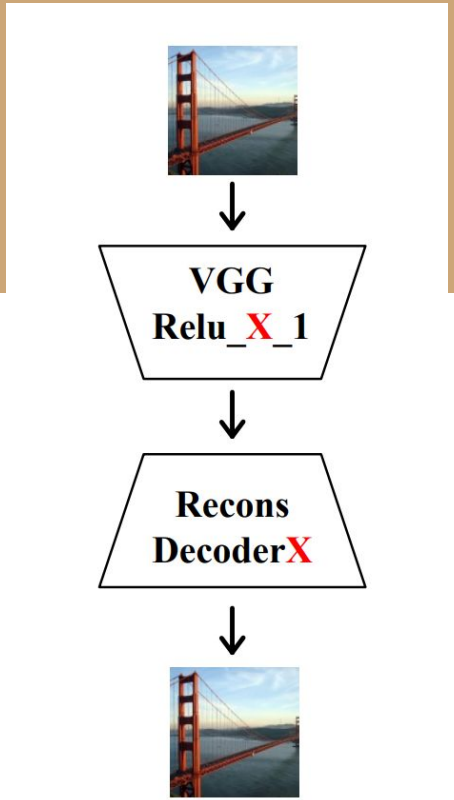


Proposed Algorithm

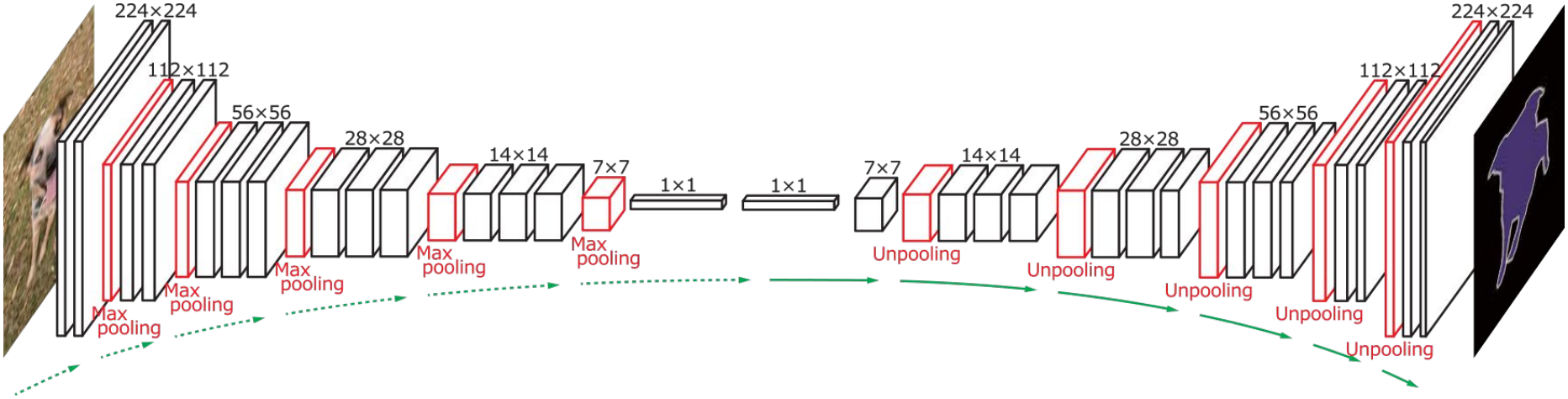
- Reconstruction decoder
- Whitening & Coloring Transform
- Multi level stylization


x_c
Feature Transforms
x_s

Reconstruction Decoder



VGG Architecture



- Network is trained on Microsoft COCO dataset.
- Transfer Learning used in our implementation.
- Encoder & Decoder weights are obtained from author's published repository.

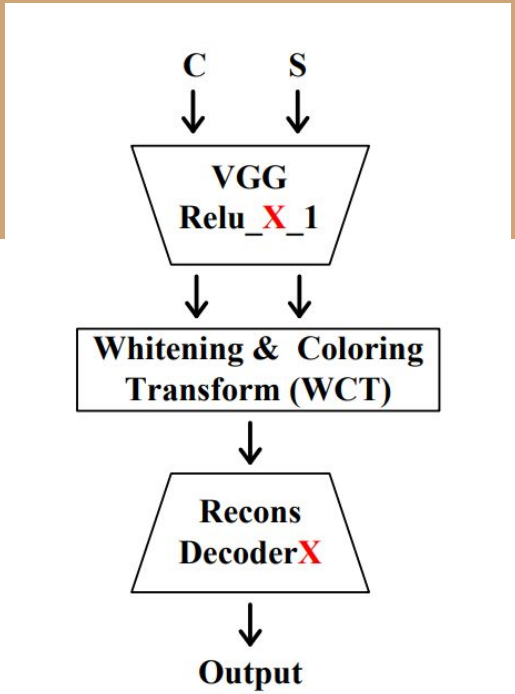
Loss function

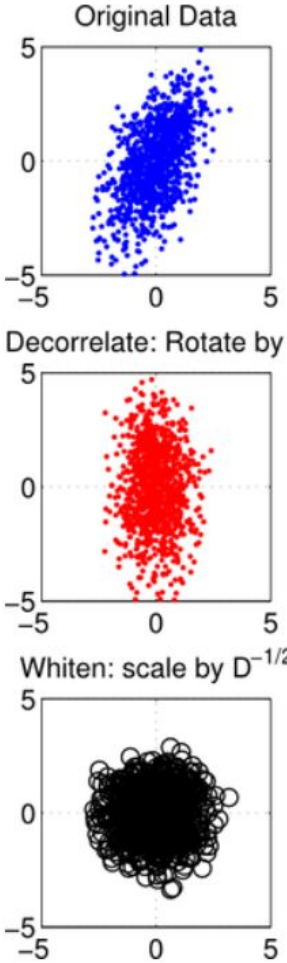
$$L = \|I_o - I_i\|_2^2 + \lambda \|\Phi(I_o) - \Phi(I_i)\|_2^2$$

Loss of network = Reconstruction Loss + Feature Loss

- I(0) = Reconstructed Output Image
- I(i) = Input Image
- Phi = Features extracted from image
- Lambda = Weight to balance the two loss

Whitening & Coloring Transforms





Whitening

$$f_c\, f_c^\top = E_c D_c E_c^\top. \quad \text{and} \quad (\hat{f}_c \hat{f}_c^{\,\top} = I)$$

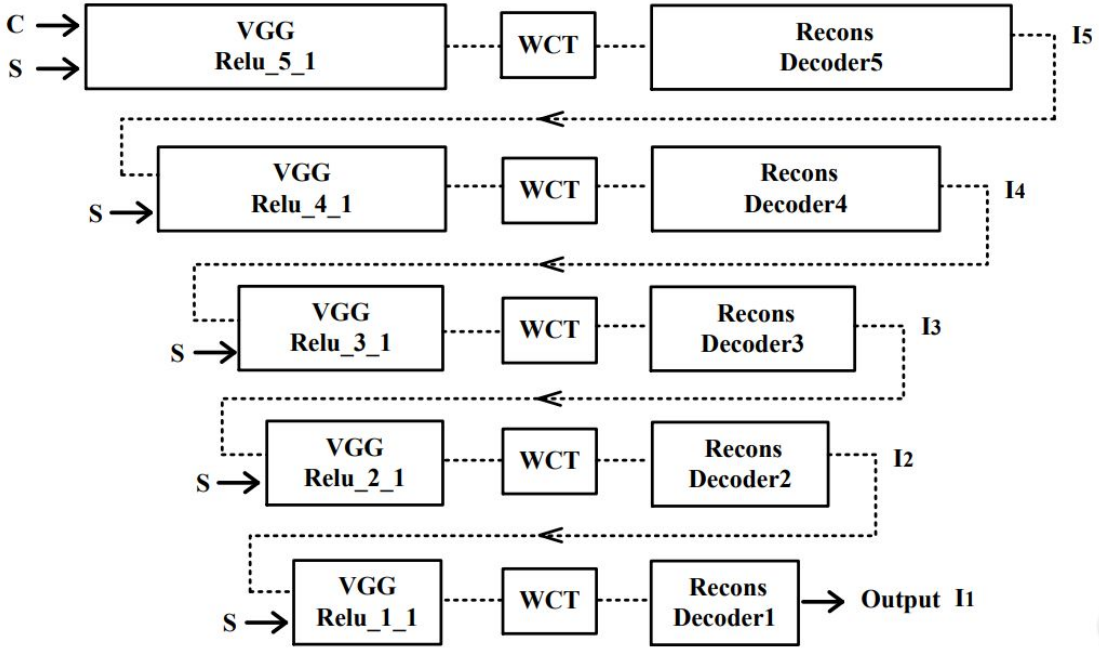
$$\hat{f}_c = E_c\, D_c^{-\frac{1}{2}}\, E_c^\top\, f_c\,,$$

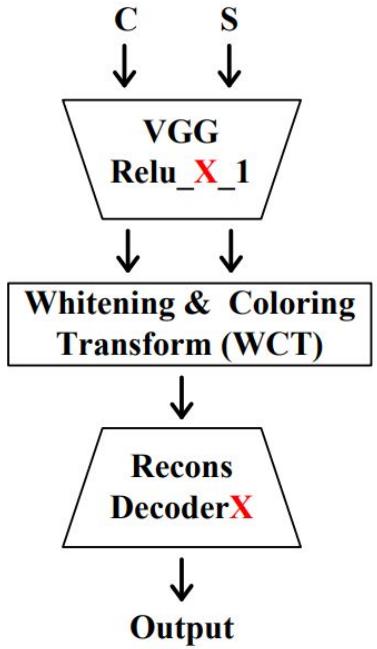
Coloring

$$(\hat{f_{cs}}\ \hat{f_{cs}}^{\top} = f_s\ f_s^{\top}) \qquad \hat{f_{cs}} = \hat{f_{cs}} + m_s$$

$$\hat{f_{cs}} = E_s\ D_s^{\frac{1}{2}}\ E_s^{\top}\ \hat{f}_c$$

Multi Level Stylization





Multi Level Stylization

(a) I_5

(b) I_4

(c) I_1

Multi Level Stylization

(a) Style

(b) Relu_1_1

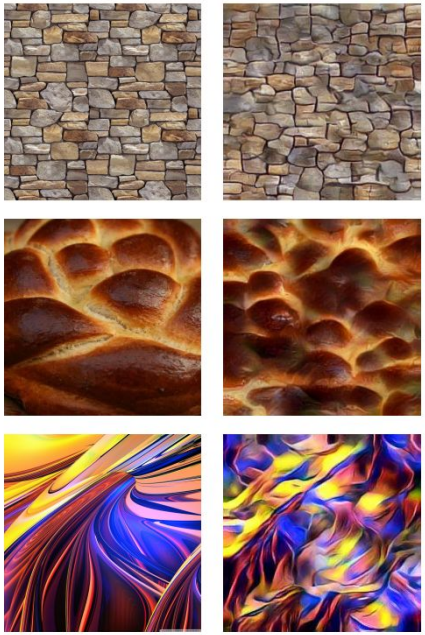
(c) Relu_2_1

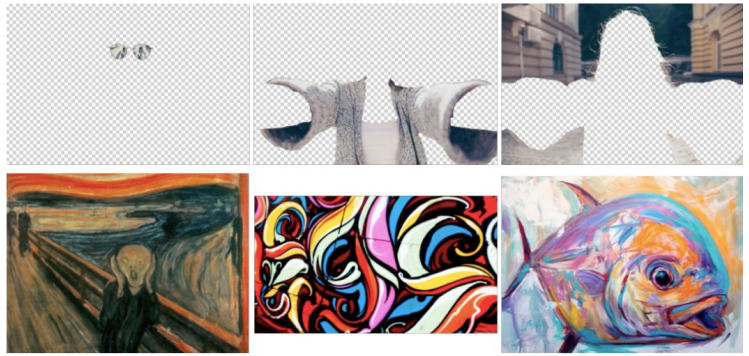
(d) Relu_3_1

(e) Relu_4_1

(f) Relu_5_1

Textures & Masks





Our Work

Style Transfer

Content Image

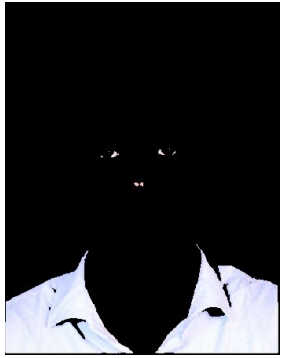
+

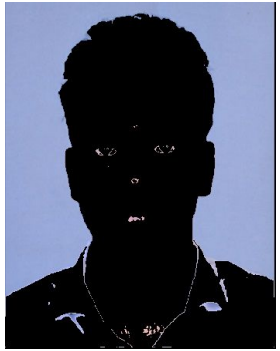
Style Image

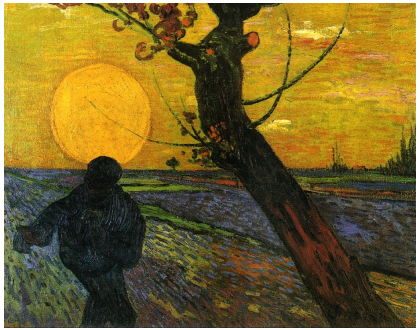
=

Style Transfer

Masking



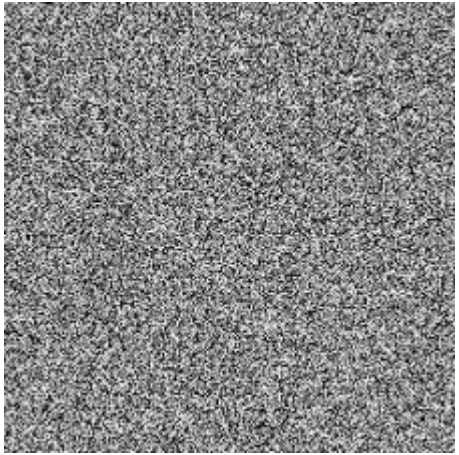




Original Content Image

Masked Output

Texture Transformation



Noise Image

Style Image

Texture Image

Thank you,

Goutham 20172063
Sathis 20172092
Sai Charan 20172086
Sai Alekhya 20172102