# NEURAL NETWORK IMAGE STYLE TRANSFER METHODS

GRIGORYEV ILYA

# ARTICLES

- **Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer** 11.04.2017
  *Xin Wang, Geoffrey Oxholm, Da Zhang, Yuan-Fang Wang*
  *University of California, Santa Barbara, CA*
  *Adobe Research, San Francisco, CA*

- **Deformable Style Transfer** 24.03.2020
  *Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, Gregory Shakhnarovich*
  *Toyota Technological Institute at Chicago*
  *University of Chicago*

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Problems of current style transfer methods

Our method

Ulyanov

Johnson

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Gatys

Johnson

Ulyanov

Our method

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- The current feed-forward networks are trained on a specific resolution of the style image, so deviating from that resolution (bigger or smaller) results in a scale mismatch

- Current networks often fail to capture small, intricate textures, like brushwork, of many kinds of artworks on high-resolution images.

## Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Learn both coarse, large-scale texture distortion and fine, exquisite brushwork of an artistic style by utilizing multiple scales of a style image

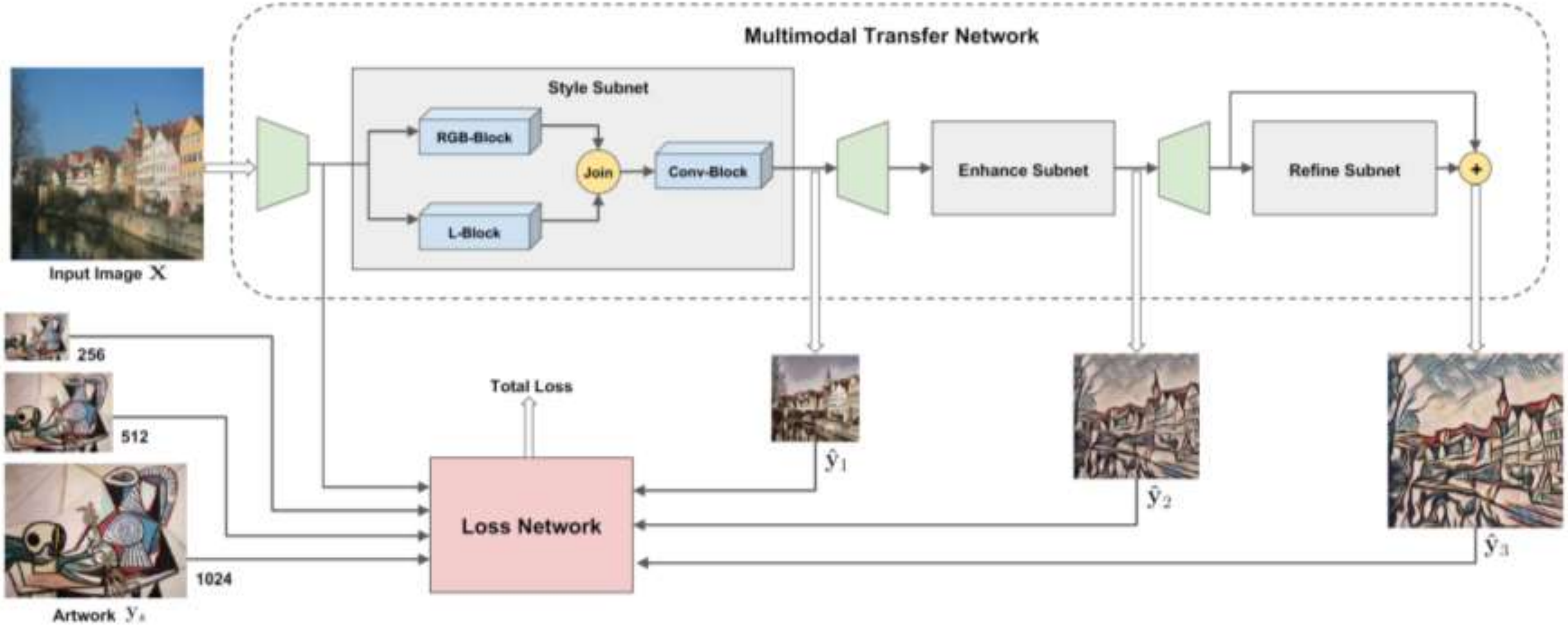
- Our hierarchical training scheme and end-to-end CNN network architecture allow us to combine multiple models into one network to handle increasingly larger image sizes

- Instead of taking only RGB color channels into consideration, our network utilizes representations of both color and luminance channels for style transfer

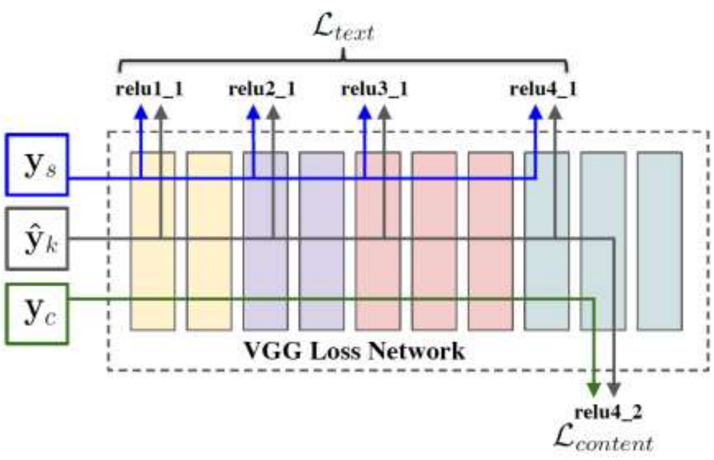
## Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Johnson proposed a feed-forward network for both fast style transfer and super-resolution using the perceptual losses defined in Gatys
- Ulyanov shows that replacing spatial batch normalization in the feed-forward network with instance normalization can significantly improve the quality of generated images for fast style transfer

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer



# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer




- Content Loss

$$\mathrm{L}_{content}(\hat{y}_k, y_c, l) = \sum_{i=1}^{N_l} \left\| F_i^l(\hat{y}_k) - F_i^l(y_c) \right\|_2^2$$

- Texture or Style Loss

$$G_{ij}^l(x) = \langle F_i^l(x), F_j^l(x) \rangle$$

$$\mathrm{L}_{style}(\hat{y}_k, y_s) = \sum_{l \in L} \left\| G^l(\hat{y}_k) - G^l(y_s) \right\|_2^2$$

## Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Finally, the stylization loss for each output $\hat{y}_k$ from the MT network is defined as a weighted sum of the content loss and the texture loss

$$L_S(\hat{y}_k, y_c, y_s) = \alpha L_{content}(\hat{y}_k, y_c) + \beta L_{style}(\hat{y}_k, y_s)$$

**Hierarchical Stylization Loss Function**

$$L_S^k(\hat{y}_k, y_c^k, y_s^k) = \alpha L_{content}(\hat{y}_k, y_c^k) + \beta L_{style}(\hat{y}_k, y_s^k)$$

**Total Loss Function**

$$L_H = \sum_{k=1}^{K} \lambda_k L_S^k(\hat{y}_k, y_c^k, y_s^k)$$

## Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Each subnet denoted by $\Theta_k$ is trained to minimize the parallel weighted stylization losses that are computed from the latter outputs $\hat{y}_i$ (i ≥ k)

$$\Theta_k = \underset{\Theta_k}{\operatorname{argmin}} E_{x\sim\chi}\left[\sum_{i\geq k}^{K} \lambda_i L_S^i(f\left(\bigcup_{j=1}^{i} \Theta_j\ , x\right), y_c^i, y_s^i)\right]$$

**Even though all those subnets are designed for different purposes, they are not totally independent!**

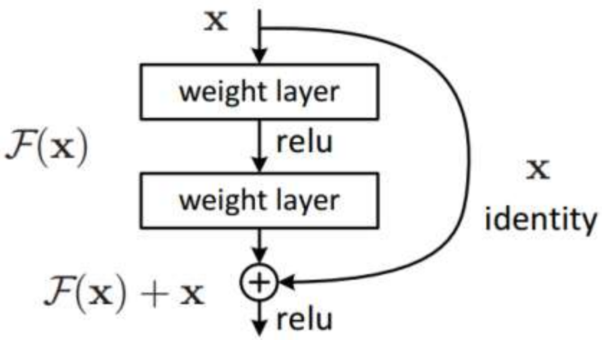
## Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

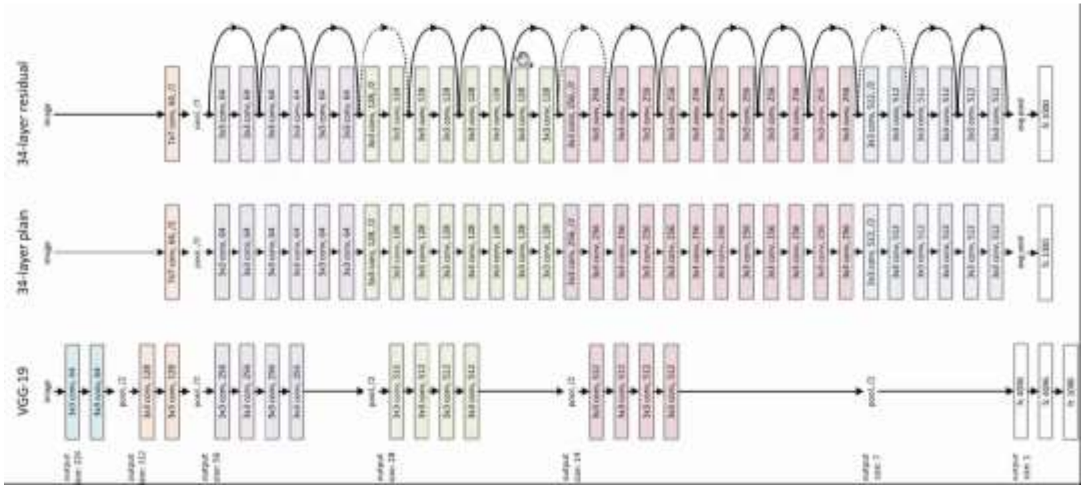
- RGB-Block and L-Block
  1. three convolutional layers (9×9, 3×3, 3×3)
  2. three residual blocks

- Conv-Block
  1. three residual blocks
  2. two resize-convolution layers for upsampling
  3. 3×3 convolutional layer to obtain the output image

- All non-residual convolutional layers are followed by instance normalization and ReLU

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer



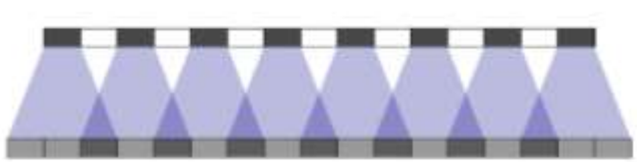



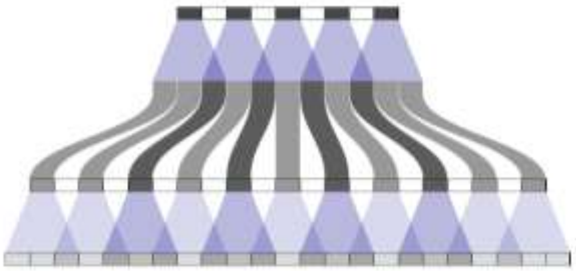
**Single Residual Block**

**ResNet architectures**

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Deconvolution and Checkerboard Artifacts



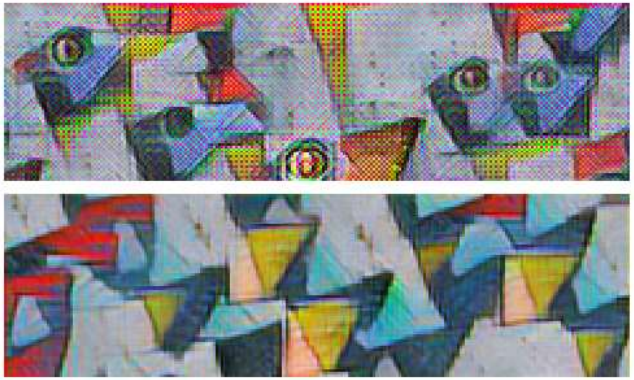


- Using nearest-neighbor interpolation or bilinear interpolation and then do a convolutional layer (resize-convolution)

Deconv in last two layers
Other layers use resize-convolution
*Artifacts of frequency 2 and 4*

Deconv only in last layer
Other layers use resize-convolution
*Artifacts of frequency 2*

All layers use resize-convolution
*No artifacts*



Using deconvolution.
*Heavy checkerboard artifacts.*

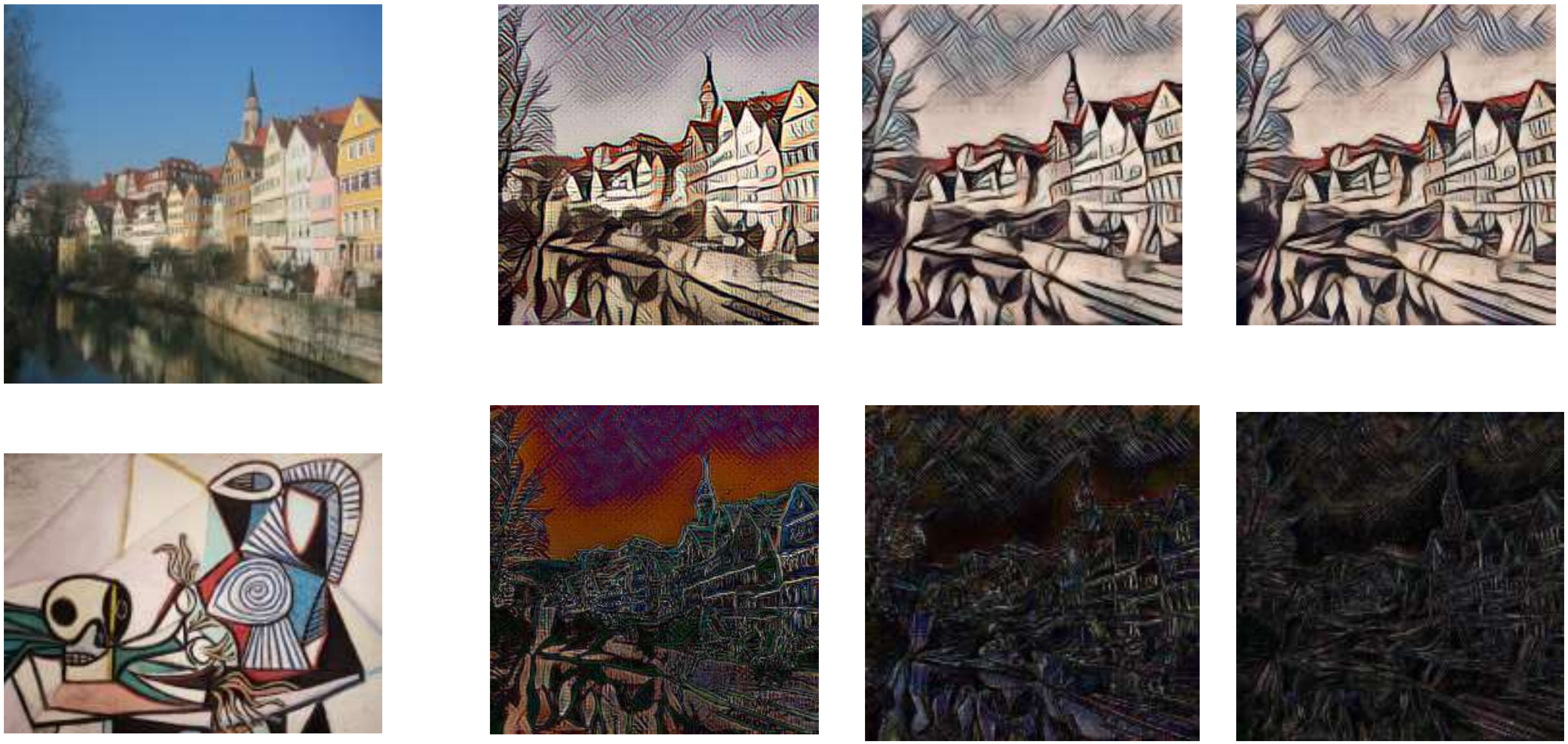
Using resize-convolution.
*No checkerboard artifacts.*

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

- Enhance Subnet
    1. convolutional layer for downsampling
    2. the same structure as the style subnet
    3. resize-convolution layer for upsampling

- Refine Subnet
    1. three convolutional layers
    2. three residual blocks
    3. two resize-convolution layers
    4. one last convolutional layer to obtain the final output

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer



# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Style

Singular Transfer (style size 256)

Singular Transfer (style size 1024)

Multimodal Transfer

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Style

Singular Transfer (style size 256)

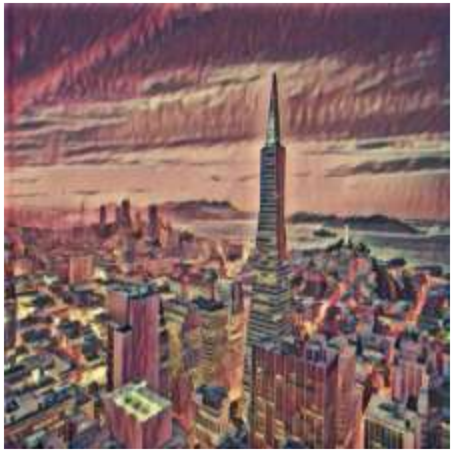
Singular Transfer (style size 1024)

Multimodal Transfer

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Style

Singular Transfer (style size 256)



Singular Transfer (style size 1024)

Multimodal Transfer

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

Style

Singular Transfer (style size 256)

Singular Transfer (style size 1024)

Multimodal Transfer

# Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

First style

Second style

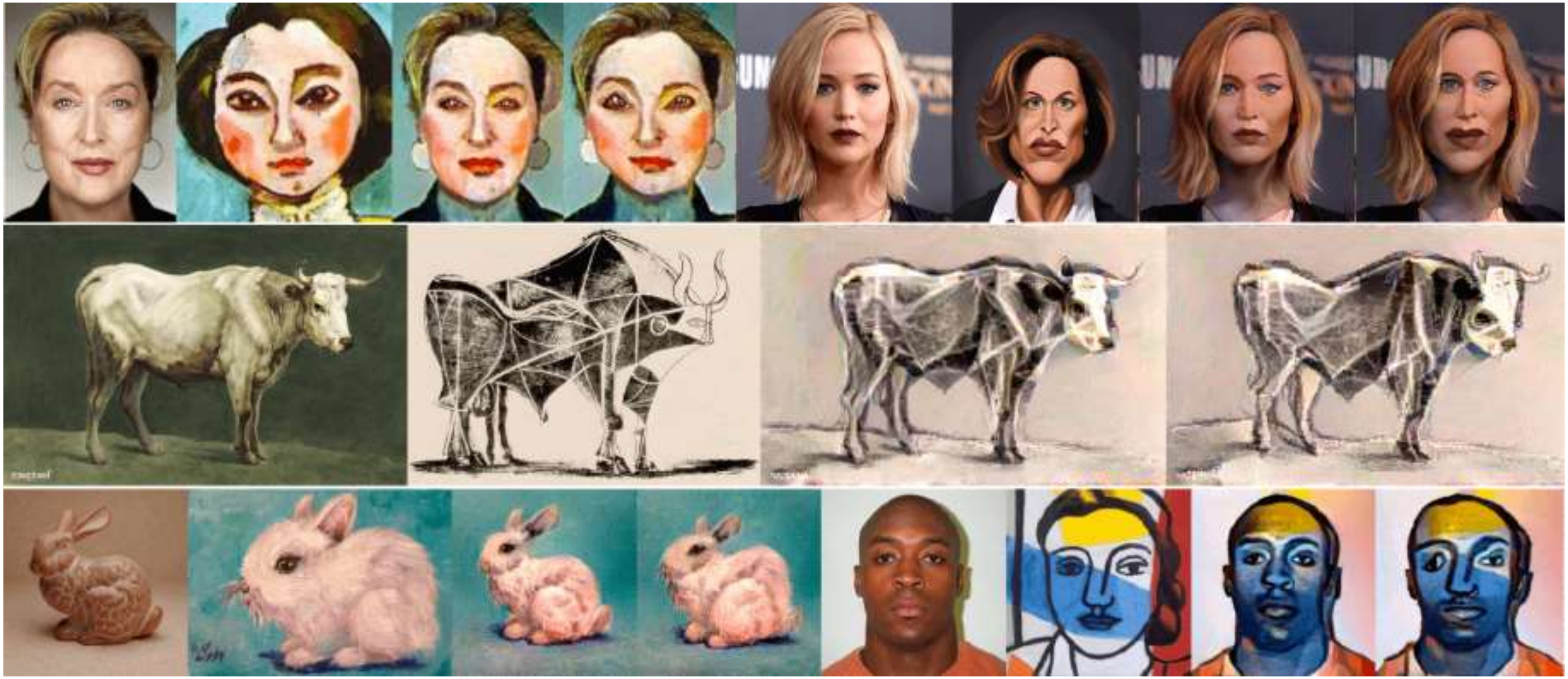
Content

# **Deformable Style Transfer** 24.03.2020

*Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, Gregory Shakhnarovich*
*Toyota Technological Institute at Chicago*
*University of Chicago*

# Deformable Style Transfer

Here proposed deformable style transfer (DST), an optimization-based approach that integrates texture and geometry style transfer.



# Deformable Style Transfer

Pablo Picasso

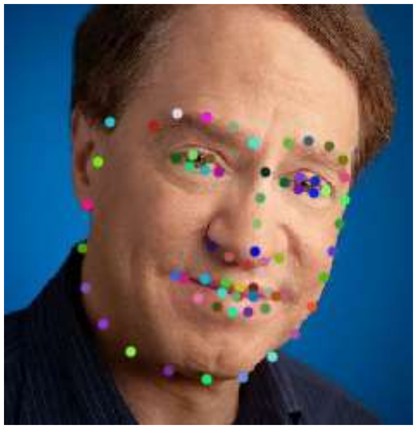
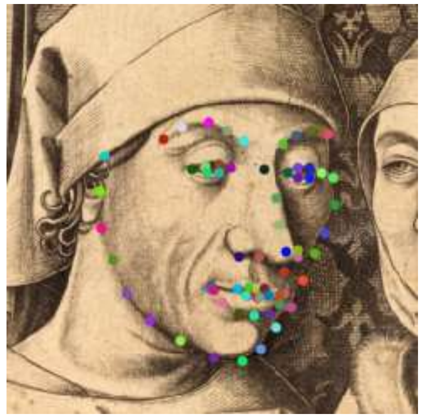
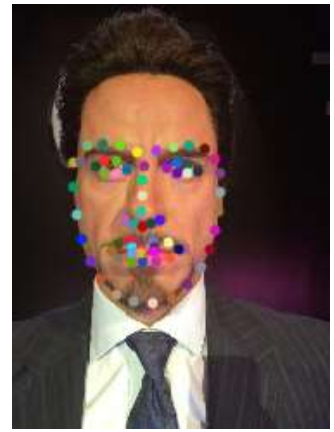
Salvador Dali

## Deformable Style Transfer

- **The key idea** in our approach is to consider a spatial deformation of the content image that would bring it into a spatial alignment with the style image. This deformation is guided by a set of matching keypoints, chosen to maximize the feature similarity between paired keypoints across the two images

# Deformable Style Transfer

## Correspondences

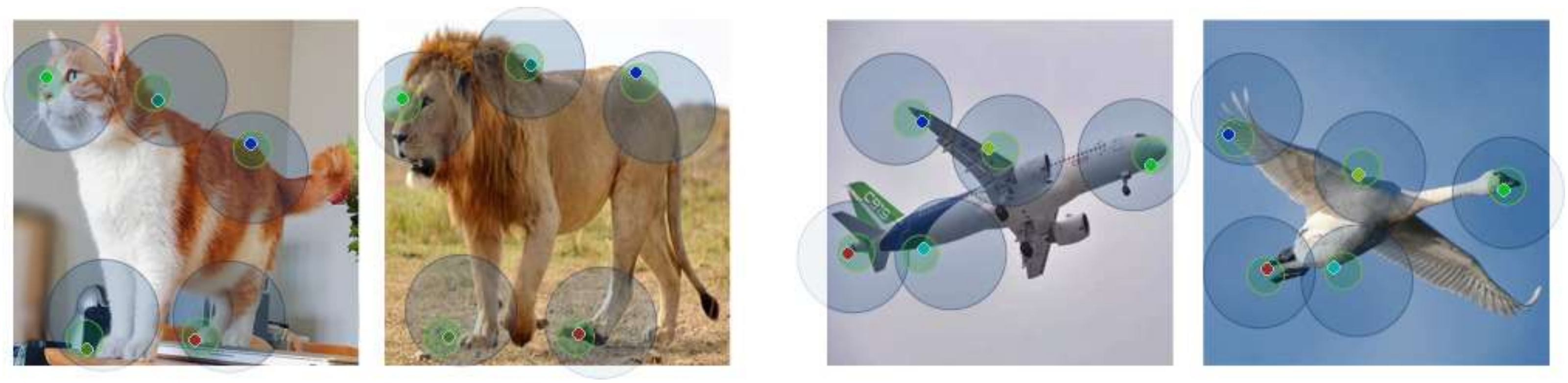







# Deformable Style Transfer

- Neural Best-Buddies (NBB)

NBB finds a sparse set of correspondences between two images that could be from different domains or semantic categories. It utilizes the hierarchies of deep features of a pre-trained CNN, that is, the characteristic that deeper layers extract high-level semantically meaningful and spatially invariant features and shallow layers encode low-level features such as edge and color features. Starting from the deepest layer, NBB searches for pairs of correspondences that are mutual nearest neighbors, filters them based on activation values, and percolates them through the hierarchy to narrow down the search region at each level. At the end of the algorithm, it clusters the set of pixel-level correspondences into k spatial clusters and returns k keypoint pairs.

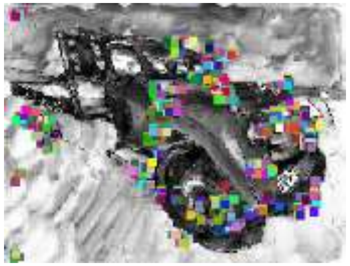


## Deformable Style Transfer

- A thin-plate spline interpolation can extend set of displacements to a full displacement field specifying how to deform every pixel in the output image.

- There is a modified NBB in DST.

# Deformable Style Transfer




# Deformable Style Transfer

- We specify an image deformation by a set of source keypoints $P = \{p_1, \ldots, p_k\}$ and the associated 2D displacement vectors $\theta = \{\theta_1, \ldots, \theta_k\}$. The $\theta_i$ specify for each source keypoint $p_i$ the destination coordinates $p_i + \theta_i$.

- The input to DST consists of a style image $I_s$, a content image $I_c$, and aligned keypoint pairs $P$ (source) and $P'$ (target).

- DST optimizes the stylization parameters (usually the pixels of the output image) $X$ and the deformation parameters $\theta$. The final output is the warped stylized image $W(X, \theta)$.

## Deformable Style Transfer

- DST framework can be used with any optimization-based style transfer method with a content loss and a style loss. In this work DST demonstrated with two such methods: Gatys and Kolkin.

- The style loss of DST is composed of two terms:

$$L_{style}(I_s, X) + L_{style}(I_s, W(X, \theta))$$

## Deformable Style Transfer

- Deformation Loss Term

Given the set of $k$ source keypoints $P$ and the matching target points $P'$, we define the deformation loss as

$$L_{warp}(P, P', \theta) = \frac{1}{k}\sum_{i=1}^{k} \|(p_i + \theta_i) - p_i'\|_2$$

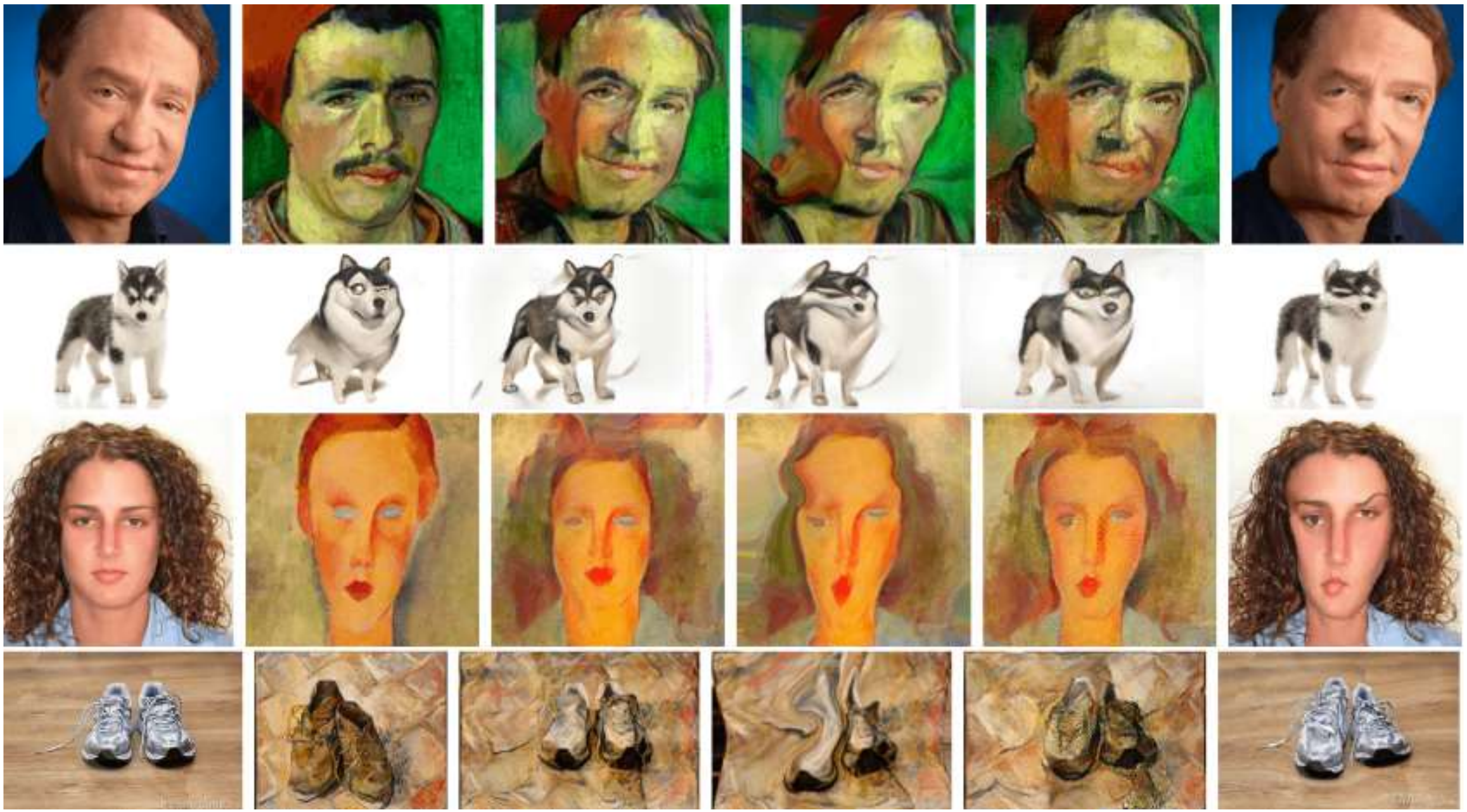
Total variation norm of the 2D warp field f normalized by its size:

$$R_{TV}(f) = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} \left\|f_{i+1,j} - f_{i,j}\right\|_1 + \left\|f_{i,j+1} - f_{i,j}\right\|_1$$

Joint optimization:

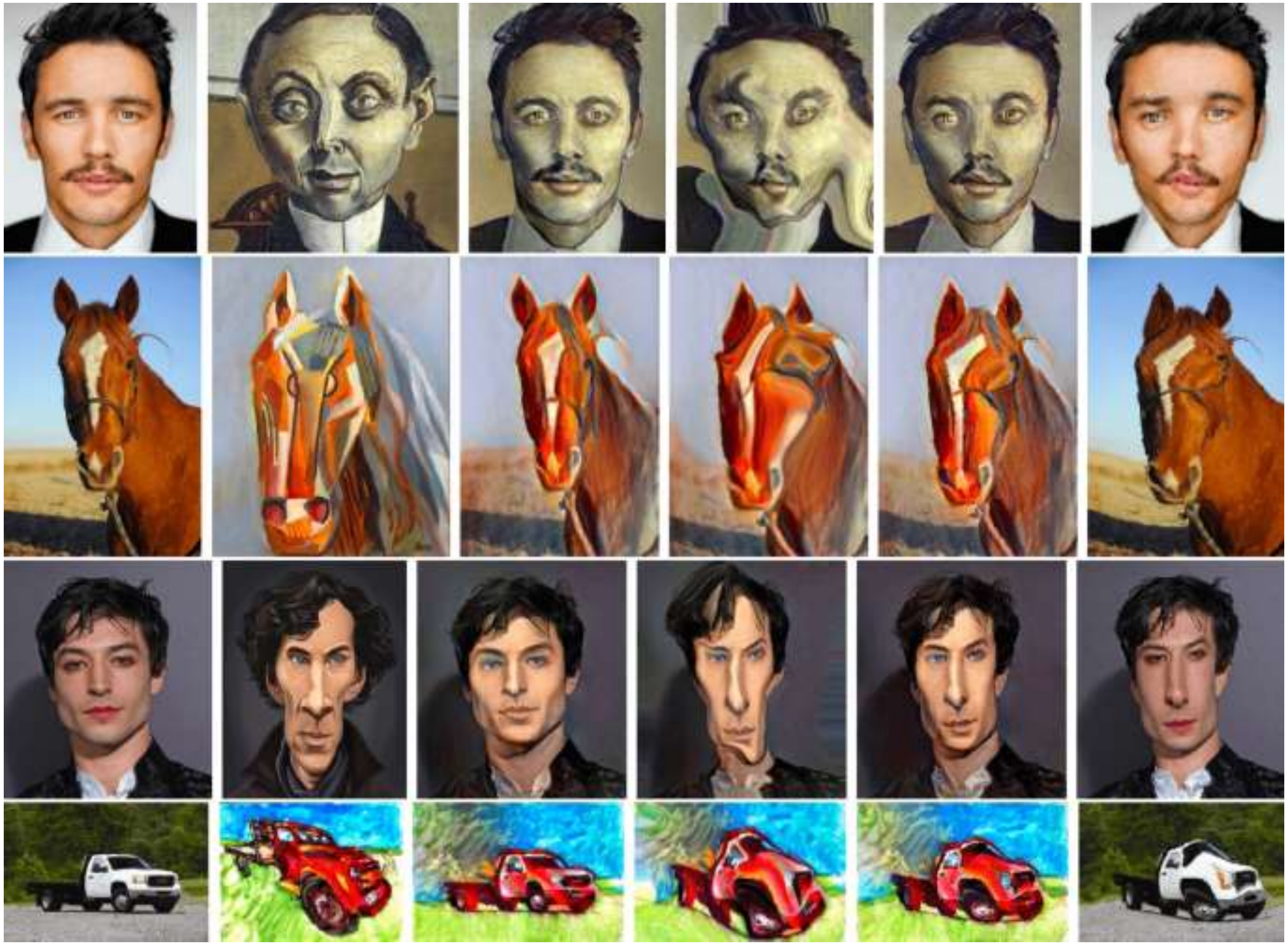
$$L(X, \theta, I_c, I_s, P) = \alpha L_{content}(I_c, X) + L_{style}(I_s, X) + L_{style}(I_s, W(X, \theta)) + \beta L_{warp}(P, P', \theta) + \gamma R_{TV}(f_\theta)$$

# Deformable Style Transfer



Content Style Gatys Gatys naive warp DST DST without ST

# Deformable Style Transfer



Content Style STROTSS STROTSS naive warp DST DST without ST

# Deformable Style Transfer

Content

Style

DST output

WarpGAN output