IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Computational Imaging Project

Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis

Team 5

SOMMAIRE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

- **Data-driven 2D Images Synthesis**

Input A Input B

Input: example image

- Style image: describe the building blocks of the image
- Content image: constraint the layout

Content A + Style B

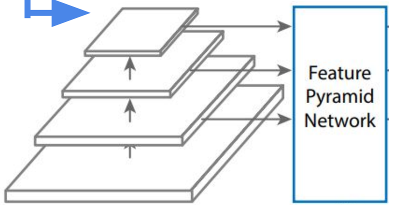
Content B + Style A

Generative Markov Random Field (MRF) Models

+

Discriminatively Trained Deep Convolutional Neural Networks (dCNNs)

control the image layout at an abstract level



- **MRF-based image synthesis**

[Assumption]
The most relevant statistical dependencies in an image are present at a local level, and use the k×k pixel patches to learn the local distribution.

[Key limitation]
- ❖ Difficulty of learning the distribution of plausible image patches from example data
- ❖ Stitch and blend mismatched local fragments

[What we need]
A powerful scheme for interpolating and adapting images from very sparse example sets of sample patches.

- **dCNN-based image synthesis**

Invariance ability
Generalization ability

→ Discriminatively trained deep Convolutional Neural Networks →

Recognize complex image features
Modeling non-linear deformations
Modeling appearance variations

❓ **[A problem]**
dCNN gradually compresses image information on multiple pooling layers into a very rough representation

❗ **[Need to do]**
Reproduce the correct neural coding statistics in the synthesis image

[Solutions]

- ➢ Use VGG to represent image in higher-level.
- ➢ Control the feature layout by penalizing the difference between the high-level neural coding of the synthesized image and the content image.
- ➢ Match the feature of the style image and the synthesized image.
- ➢ Use Gram matrix regularization.

[However] Strict local plausibility is still difficult. The constraint of spatial layout is too weak.

- **MRF + dCNN**

Locally correlated information

Translational invariance

[In this paper]
★ Gram matrix is replaced by MRF regularizer.
★ An additional energy term to model the Markovian consistency in the upper layer of dCNN.
★ Use the EM algorithm for MRF optimization.

Improve local plausibility of the feature layouts

MRF

dCNN

Match semantically related image portions without user annotations

Match and adapt local features with considerable variability

[MRF-based image synthesis]

MRF models suffer from a significant limitation: Local image statistics is usually not sufficient for capturing complex image layouts at a global scale.
Multi-resolution synthesis provides some improvement (and the authors adapt this in their method).
However, a principled solution requires additional high-level constraints. These can be either explicitly provided by the user, or learned from non-local image statistics. Long range correlations have also been modeled by spatial LTSM neural networks.

[Image synthesis with neural networks]

- ➢ Zeiler et al. introduce a deconvolutional network to back-project neuron activations to pixels.
- ➢ Mahendran and Vedaldi reconstruct images from the neural encoding in intermediate layers.
- ➢ Gauthier et al. extend GAN by a Laplacian pyramid.
- ➢ Denton et al. extend the model to a conditional setting, limited to generating faces.

WGN → backprop to update

Objective:
Synthesize an image x with its template guided by a content image x_c and having the textures from a style image x_s.

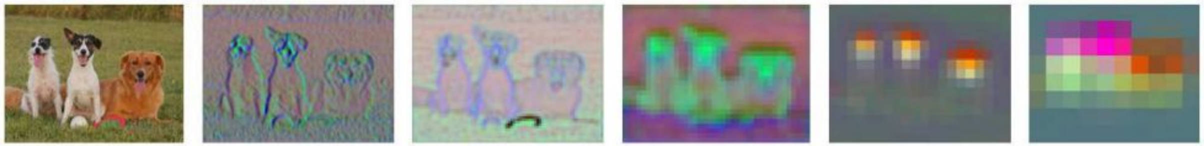
→ ***Optimization problem*** *which minimizes a loss measuring the differences between the* *template of x and* x_c *and between the* *textures of x and* x_s.

Prerequisites: Deep convolutional network
→ *Needed for later explanation.*

- translational invariant
- its information is locally correlated
- higher feature becomes more invariant under in-class variation

approximate **Markovian consistency** properties

Bonus: consider the 3x3 **patch** (or any other sizes) of the feature maps ("**neural patch**"), when going deeper in the convnet, the level of discrimination of this patch gets higher.


input image relu2_1 relu3_1 relu4_1 relu5_1 pool5

Credited to Figure 2 in the original paper

Idea:

Template guidance means that $\boldsymbol{x}$ looks "globally" (excluding meso-structures and also micro details such as pixel color, local edges, etc.) similar to the content image $\boldsymbol{x_c}$.

→ the high-level features of x are constrained to that of the content image → minimize the distance between these features.

Content loss:

$$E_c(\Phi(\mathbf{x}), \Phi(\mathbf{x}_c)) = ||\Phi(\mathbf{x}) - \Phi(\mathbf{x}_c)||^2$$

feature map of x at a certain layer in the network

Idea:

x having the same textures as the style image $\boldsymbol{x_s}$, *without replicating accurate pixel value* (micro details) → style loss on **feature domain.**
Same texture = inherits **patterns** + **meso-structures** of $\boldsymbol{x_s}$ → "locally" similar

→ *Replace **loss** performed on a whole feature map = **loss** on each patch of the feature map.*
→ *result more* visually plausible → **MRF style loss** $E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2$

Loss = distance between a **neural patch of *x*** and its **"best matching" patch $\boldsymbol{\Psi_{NN(i)}(.)}$**

Why? → (approximate) Markovian consistency: each **neural patch of *x*** must be "**linked**" to the **most relevant neural patch of $\boldsymbol{x_s}$**.
How? → normalized cross-correlation: $NN(i) := \arg\min_{j=1,\ldots,m_s} \frac{\Psi_i(\Phi(\mathbf{x})) \cdot \Psi_j(\Phi(\mathbf{x}_s))}{|\Psi_i(\Phi(\mathbf{x}))| \cdot |\Psi_j(\Phi(\mathbf{x}_s))|}$ → **Maximize** (not minimize) this one → this can be done by an additional convolutional layer.

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Idea:
The model has to make the ***x*** smooth and natural → Why? In CNN, feature maps are **downsampled** → **information loss** → a noisy and unnatural reconstruction.

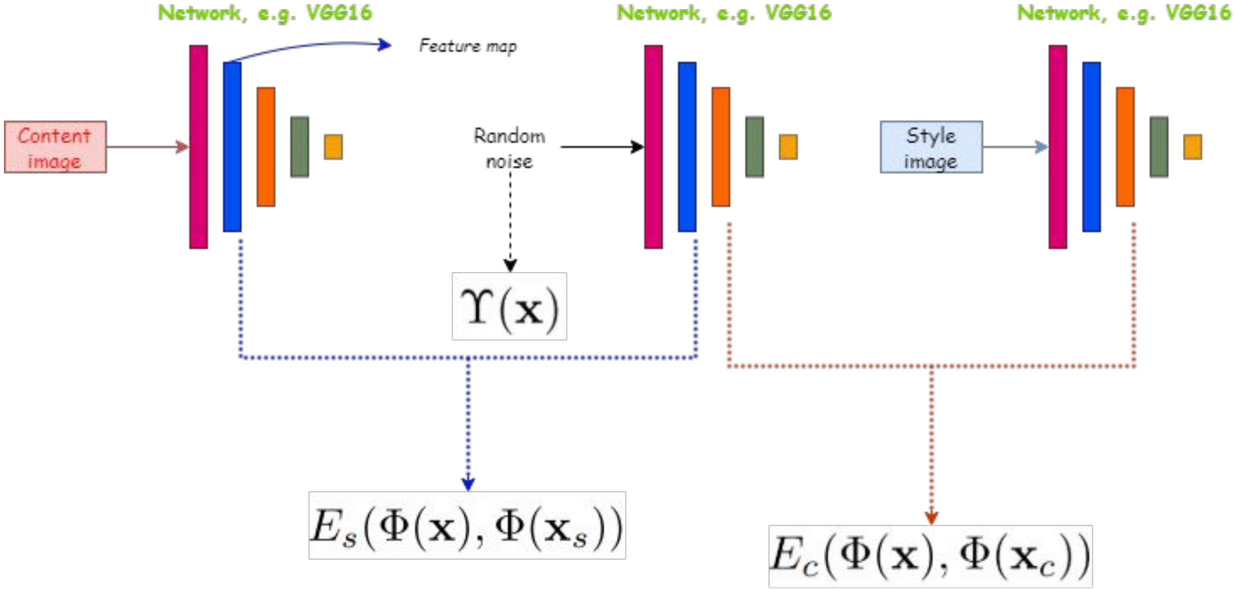
→ Here comes the smoothness regularization term to penalize the image

$$\Upsilon(\mathbf{x}) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)$$

→ an MRF prior uses a **smaller neighborhood** compared to the MRF style prior

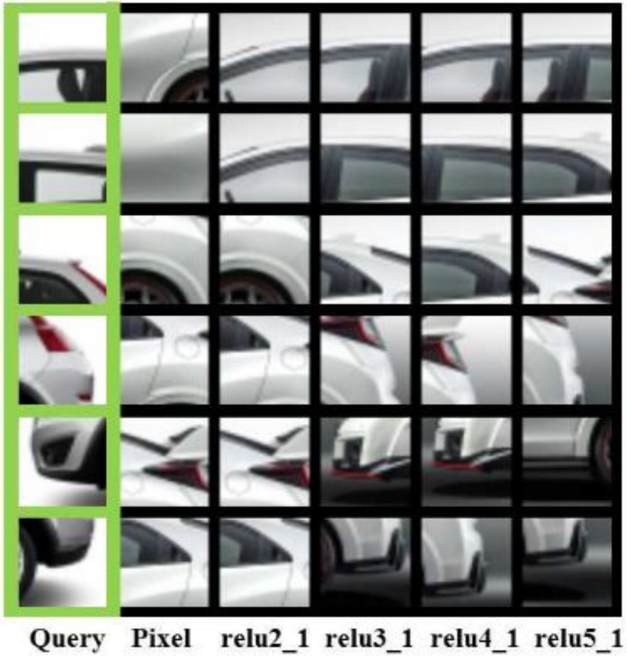
2.4 Final objective/loss function

$$\mathbf{x} = \arg\min_{x} E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) + \alpha_1 E_c(\Phi(\mathbf{x}), \Phi(\mathbf{x}_c)) + \alpha_2 \Upsilon(\mathbf{x})$$



Visualization for better understanding of the objective function.

Example: Matching two different car images.



The matching results from *relu3_1* and *relu4_1* are the best

Minimizing the equation:

$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2$$

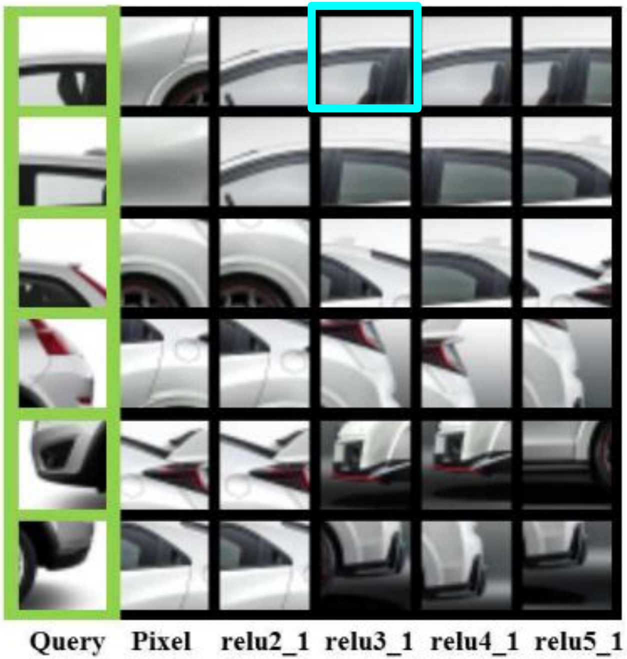
A linear blending operation for overlapping patches

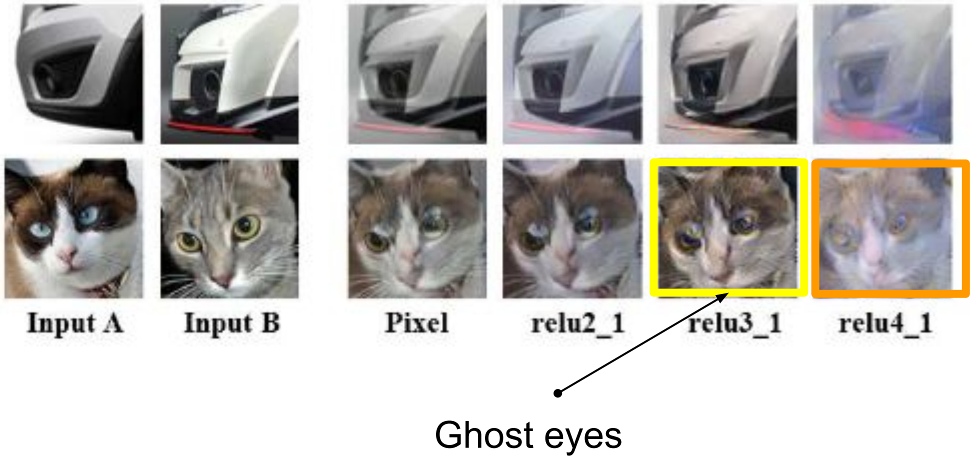
Pixel space vs feature space blending comparison

Input A Input B Pixel relu2_1 relu3_1 relu4_1

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

It is still possible for a dCNN to generate **implausible results**

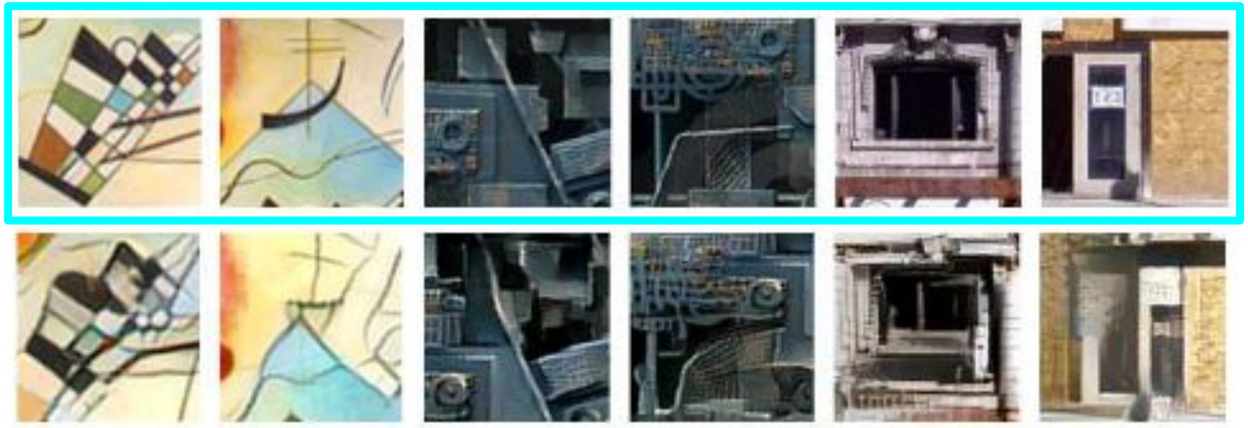


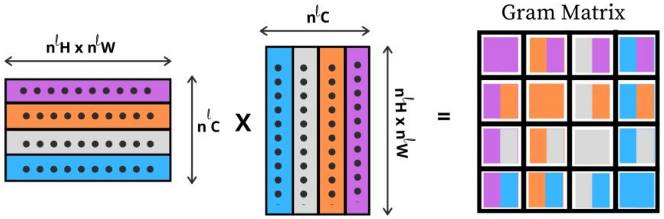


MRF prior reduces the artifacts in synthesized images

MRF prior

Style constraint based on match-ing Gram matrices





Input A Input B

InputC: synthesized image (x) (initialized as random noise)

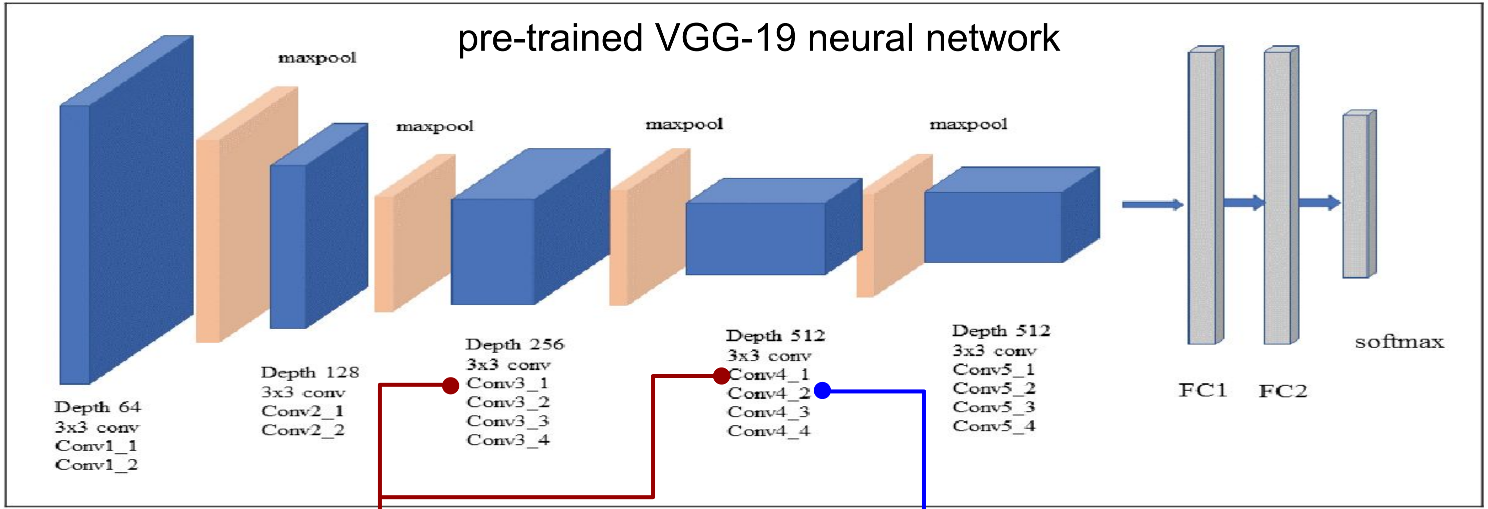


Fig. 3. VGG-19 network architecture

MRF style loss

Content loss

$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2$$

$$+ \ \Upsilon(\mathbf{x})$$

$$E_c(\Phi(\mathbf{x}), \Phi(\mathbf{x}_c)) = ||\Phi(\mathbf{x}) - \Phi(\mathbf{x}_c)||^2$$

update x by minimizing the equation using back-propagation

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

The patch matching:

$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \boxed{\Psi_{NN(i)}(\Phi(\mathbf{x}_s))}||^2$$

- For both layer relu3_1 and relu4_1, the author used 3*3, stride to 1 patch
- The patch matching is implemented as an additional convolutional layer : The best matching of a patch is the filter that gives the maximum response.

Input: synthesized image
(initialized as random noise)

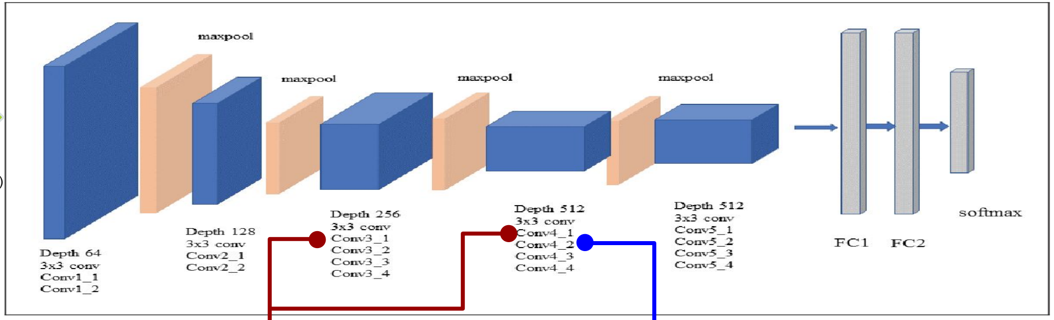


Fig. 3. VGG-19 network architecture

add a convolutional layer
(filters : patches from the style image)

$$E_c(\Phi(\mathbf{x}), \Phi(\mathbf{x}_c)) = ||\Phi(\mathbf{x}) - \Phi(\mathbf{x}_c)||^2$$

$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2$$

3.4 Implementation details

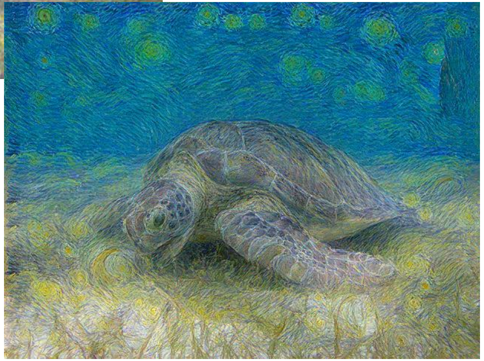
Practice details : A multi-resolution process

1. Scale the synthesized image, the style image and the content image in lower resolution
2. Feeding them into the network and perform 200 iterations
3. Up-sample the previous output
4. Feed the previous results and the reference images into the network and repeat the above steps

Running time : take about three minutes to synthesis an image of size 384*384 with a Titan X GPU.

res1

res2



res3

More details :

Create copies for patches from the style image with different rotations and scales :

- seven scales: {0.85; 0.9; 0.95; 1; 1.05; 1.1; 1.15}
- five rotations: $\{-\frac{\pi}{12}, -\frac{\pi}{24}, 0, \frac{\pi}{24}, \frac{\pi}{12}\}$

Why : To overcome the perspective and scale difference between the style and the content images

However, increasing the number of patches is computational expensive, so the author only used the rotational copies for objects that can deform – for example : faces.

CHAPTER 4 : Results and Discussion

4.1 Results—— stylizing photos by artwork

Style Image | Content Image | Gatys et al | Ours

Gatys et al:

- **eyes look unnatural**
- **lost the characteristic shapes in the original painting**
  partially blends with the content exemplar

MRF:

synthesized more reasonable facial features

- **eyes and nose are faithfully preserved in Picasso's paintings**
- **eyes and mouth are synthesized as simple shapes**
  hair as regions of dark color

CHAPTER 4 : Results and Discussion

4.1 Results——photorealistic synthesis

Input style

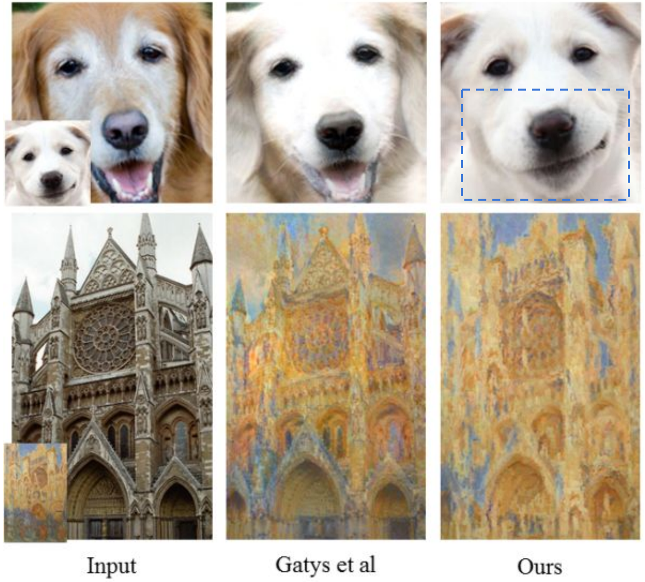
Input content

Gatys et al

Ours

a good match => produces more reasonable results
mis-matching => deviates from the content image
no matching => replaces with texture synthesis

- only works if the content image can be re-assembled by the MRFs in the style image
- not as sharp as the original image



loses important features

This paper mainly combines the discriminative power of deep neural networks with classical MRF-based texture synthesis to develop a method for style transfer between images, which permits transferring photo-realistic styles with some plausibility.

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Thank you for your attention

Team 5