Image Inpainting

Yan Zhao

May 21, 2018

Contents：

Contents：

2 Image Quality Assessment

- PSNR
- SSIM

Contents：

3 Classical Methods to Image Inpainting

- Examplar-based Inpainting
- OpenCV

Contents：

4 Deep Learning to Image Inpainting

- Datasets
- Methods in Recent Papers

Methods in Recent Papers：

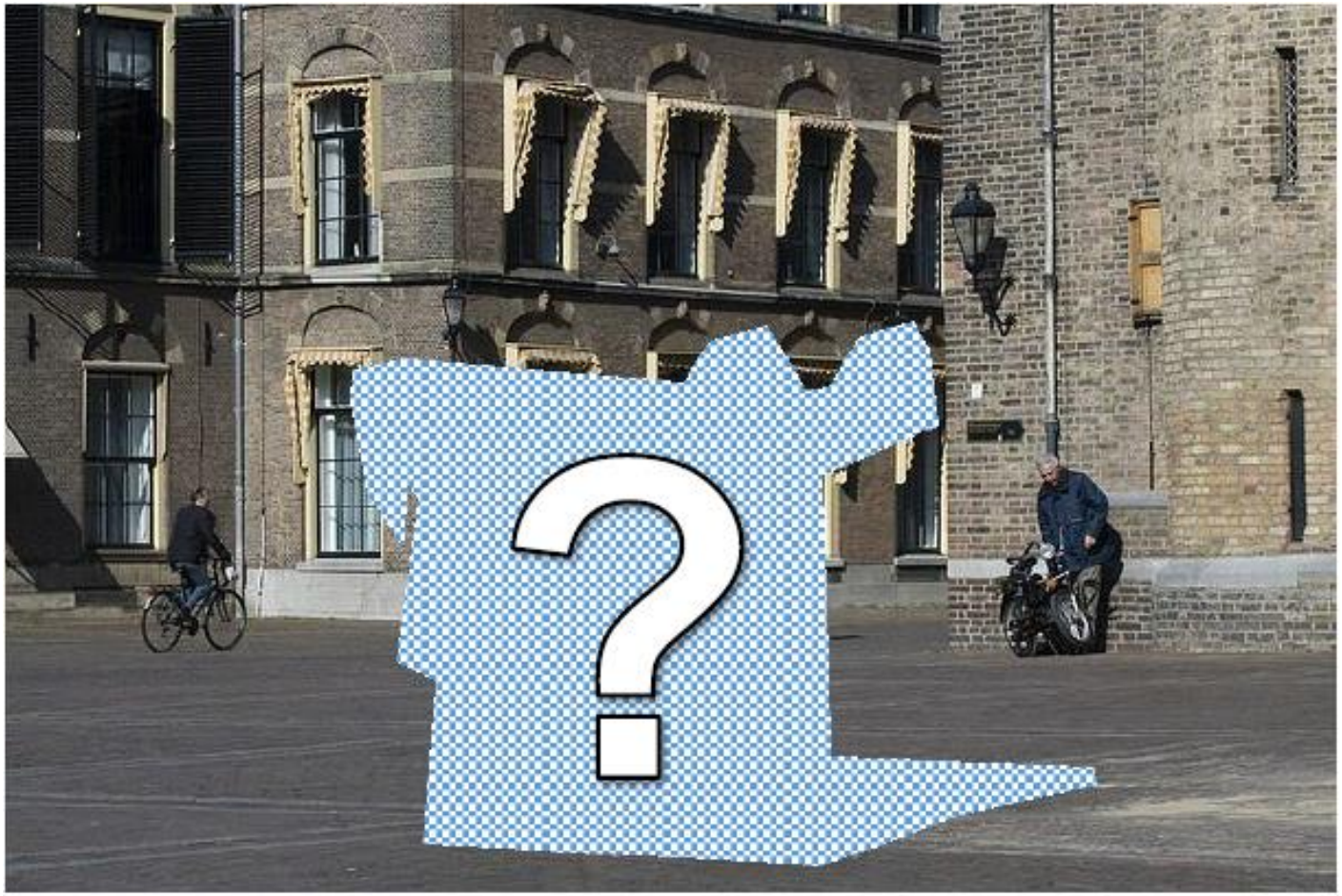
- Globally and Locally Completion
- Semantic Image Inpainting
- High-Resolution Image Inpainting
- Content Encoder

If your photo is damaged, what would you do?

The image completion problem

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

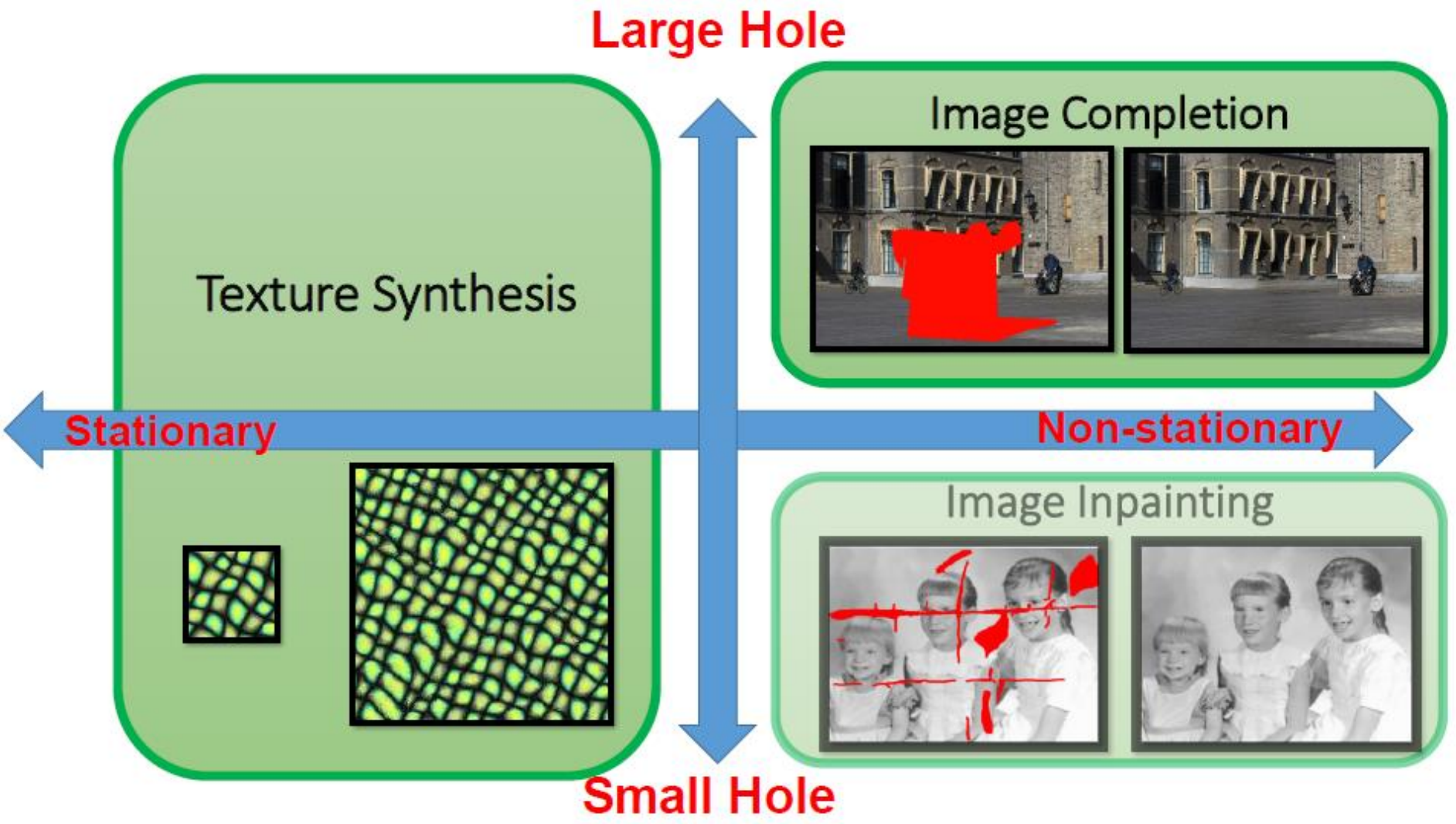
The image completion problem



https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

The image completion problem

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014



https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

What is Image Inpainting ?

Original and restored image

Inpainting is the process of reconstructing lost or deteriorated parts of images and videos.

https://en.wikipedia.org/wiki/Inpainting

What is Image Inpainting ?

https://www.youtube.com/watch?v=1F-6iRrgh1s

Objective: Make attentive inpainter

IF BACKGROUND OF TARGET REMOVING OBJECT IS SIMPLE, EXISTING METHOD WORKS FINE

HOWEVER, IF BACKGROUND OF TARGET REMOVING OBJECT IS COMPLEX, BETTER NEED ANOTHER METHOD

https://www.slideshare.net/ssuser73ec8f/semantic-image-inpainting

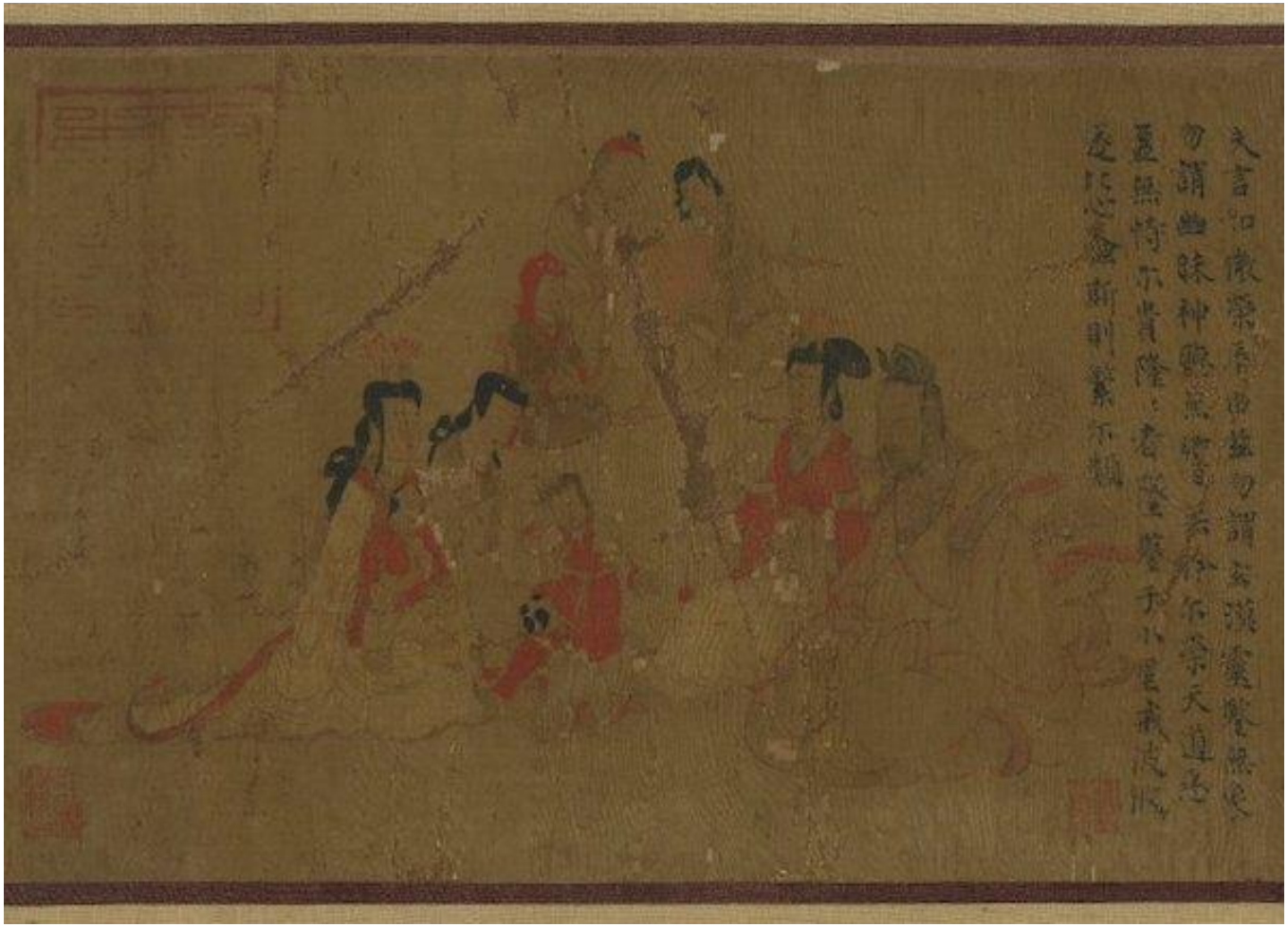
What is Image Inpainting ?

- In the museum world： in the case of a valuable painting, this task would be carried out by a skilled art conservator or art restorer.

- In the digital world： inpainting (also known as *image interpolation* or *video interpolation*) refers to the application of sophisticated algorithms to replace lost or corrupted parts of the image data (mainly small regions or to remove small defects).

https://en.wikipedia.org/wiki/Inpainting

In the museum word：

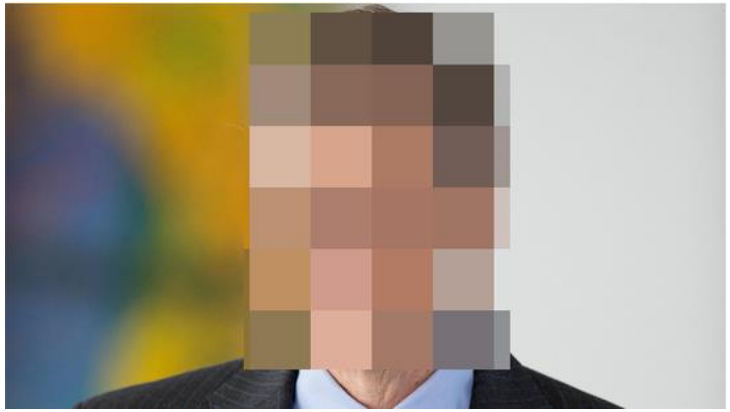


http://p0.ifengimg.com/haina/

In the digital word：

Photo repair

De-watermarking



De-masking

Target removal

Image Inpainting Methods

Inpainting is rooted in the restoration of images. Traditionally, inpainting has been done by professional restorers. The underlying methodology of their work is as follows:

- The global picture determines how to fill in the gap. The purpose of inpainting is to restore the unity of the work.
- The structure of the gap surroundings is supposed to be continued into the gap. Contour lines that arrive at the gap boundary are prolonged into the gap.
- The different regions inside a gap, as defined by the contour lines, are filled with colors matching for those of its boundary.
- The small details are painted, i.e. “texture” is added.

https://en.wikipedia.org/wiki/Inpainting

Image Inpainting Methods

- Structural inpainting
- Textural inpainting
- Combined Structural and textural inpainting

https://en.wikipedia.org/wiki/Inpainting

Structural inpainting

- Use geometric approaches for filling in the missing information in the region which should be inpainted.
- These algorithms focus on the consistency of the geometric structure.

https://en.wikipedia.org/wiki/Inpainting

Textural inpainting

- All the structural inpainting methods are not able to restore texture.

- Texture has a repetitive pattern which means that a missing portion cannot be restored by continuing the level lines into the gap.

https://en.wikipedia.org/wiki/Inpainting

Combined Structural and textural inpainting

- Perform texture and structure filling in regions of missing image information.

- The boundaries between image regions accumulate structural information which is a complex phenomenon. This is the result when blending different textures together.

- The state of the art inpainting method attempts to combine structural and textural inpainting.

https://en.wikipedia.org/wiki/Inpainting

Applications of Image Inpainting

- **Photo repair**
- De-watermarking
- De-mosaicing
- Object removal
- De-blurring

http://www.baike.com/wiki/

Photo repair

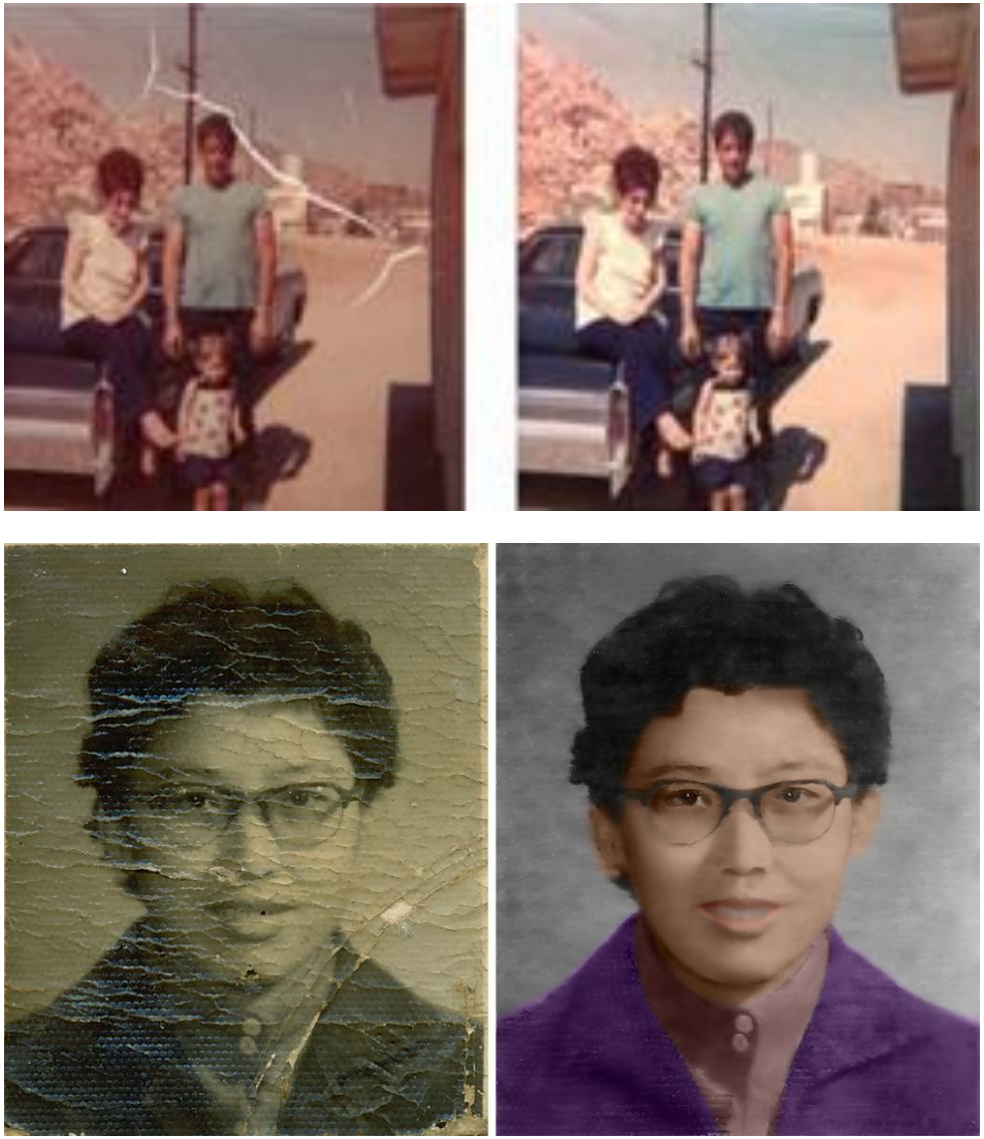


Photo repair - Adobe Photoshop

Resized

Imitation stamp

Original image

Repaired image

https://jingyan.baidu.com/article/d2b1d102bc14865c7e37d4cf.html

Applications of Image Inpainting

- Photo repair
- **De-watermarking**
- De-mosaicing
- Object removal
- De-blurring

http://www.baike.com/wiki/

De-watermarking – Adobe Photoshop

http://www.16xx8.com/photoshop/jiaocheng/2014/135267.html

De-watermarking – OpenCV (*Open Source Computer Vision*)

- A library of programming functions mainly aimed at real-time computer vision
- OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

OpenCV 's application areas include:

2D and 3D feature toolkits	Egomotion estimation
Facial recognition system	Gesture recognition
Human–computer interaction (HCI)	Augmented reality
Mobile robotics	Motion understanding
Object identification	Segmentation and recognition
Structure from motion (SFM)	Motion tracking

https://en.wikipedia.org/wiki/OpenCV

De-watermarking – OpenCV

Original image

Masked image

SIFT
FLANN

Inpaint

Inpainted image

https://blog.csdn.net/yhmabcdef/article/details/75214020

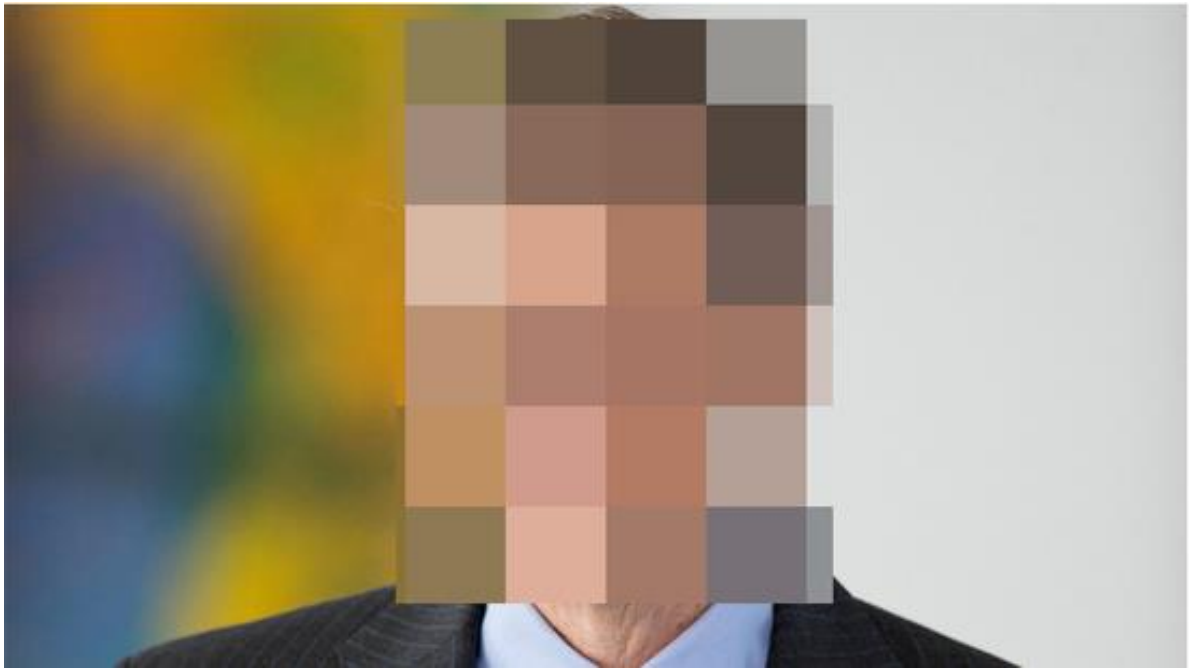
Applications of Image Inpainting

- Photo repair
- De-watermarking
- **De-mosaicing**
- Object removal
- De-blurring

http://www.baike.com/wiki/

De-mosaicing

In the Internet era of increasingly sensitive personal information, Internet users are generally used to cover sensitive information on photos with “fuzzy” or “mosaic”.



http://www.pc6.com/edu/100603.html

De-mosaicing VS Auto-mosaicing

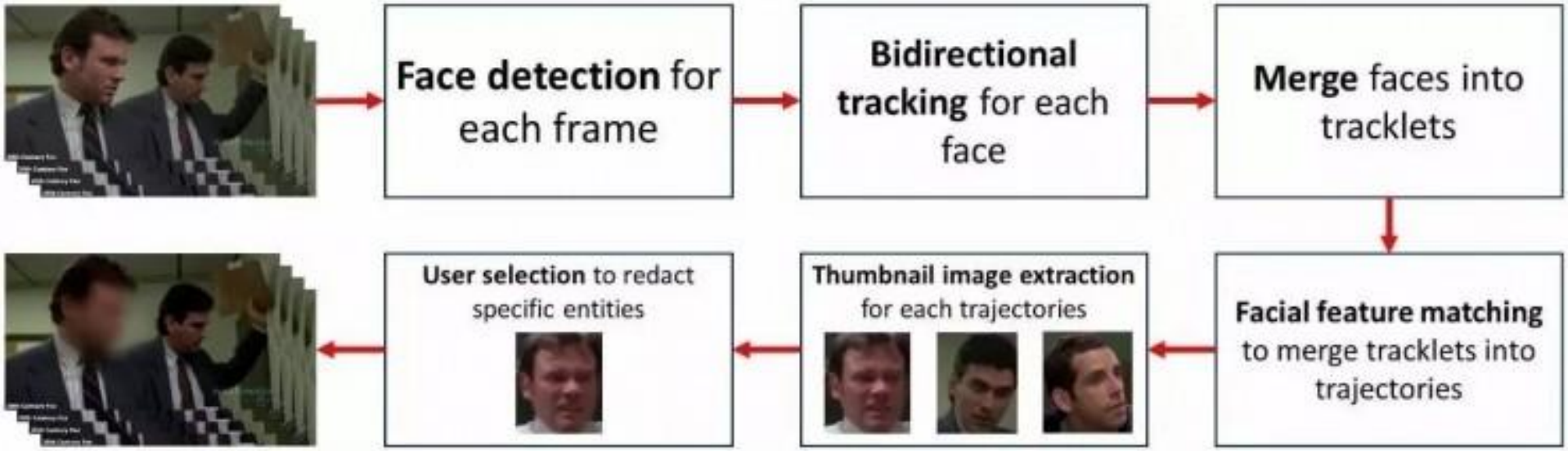
https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

Microsoft: Auto-mosaicing

- Original video frame
- When the character appears, the user selects a person to mosaic
- Masked video frame

https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

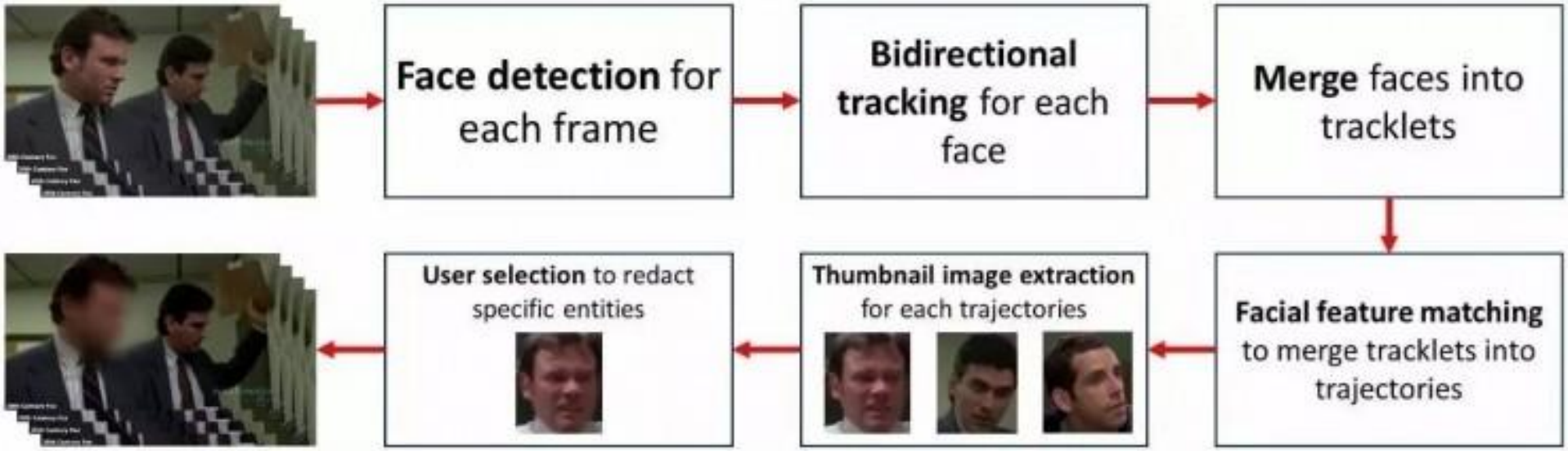
Microsoft: Auto-mosaicing



- Find out where all the faces in the video are
- Connect all the faces of the same person.

https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

Microsoft: Auto-mosaicing



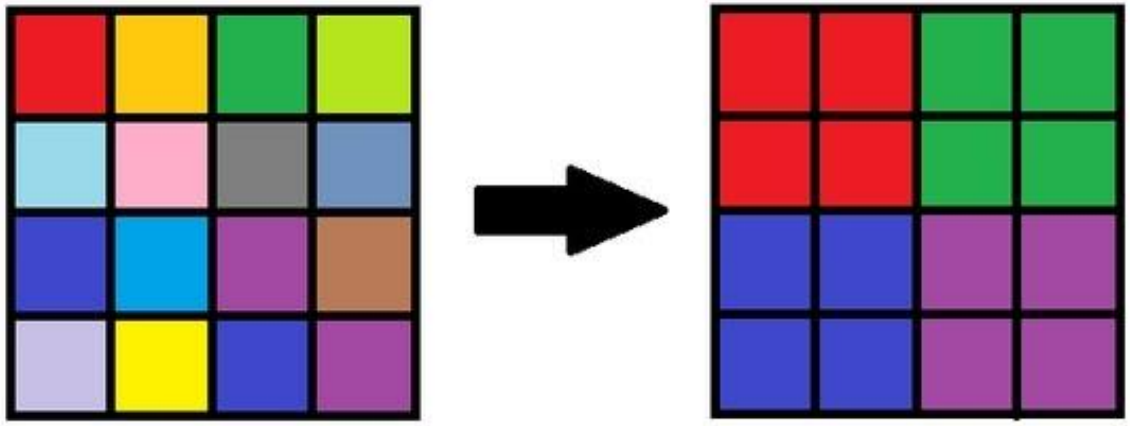
- Face detection
- Face tracking
- Face recognition

https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

De-mosaicing

In general, mosaic is irreversible loss information, it is difficult to restore.

Mosaicing is a process of reducing the amount of information that has been lost. It is equivalent to sampling the image signal at a lower frequency than the original data.



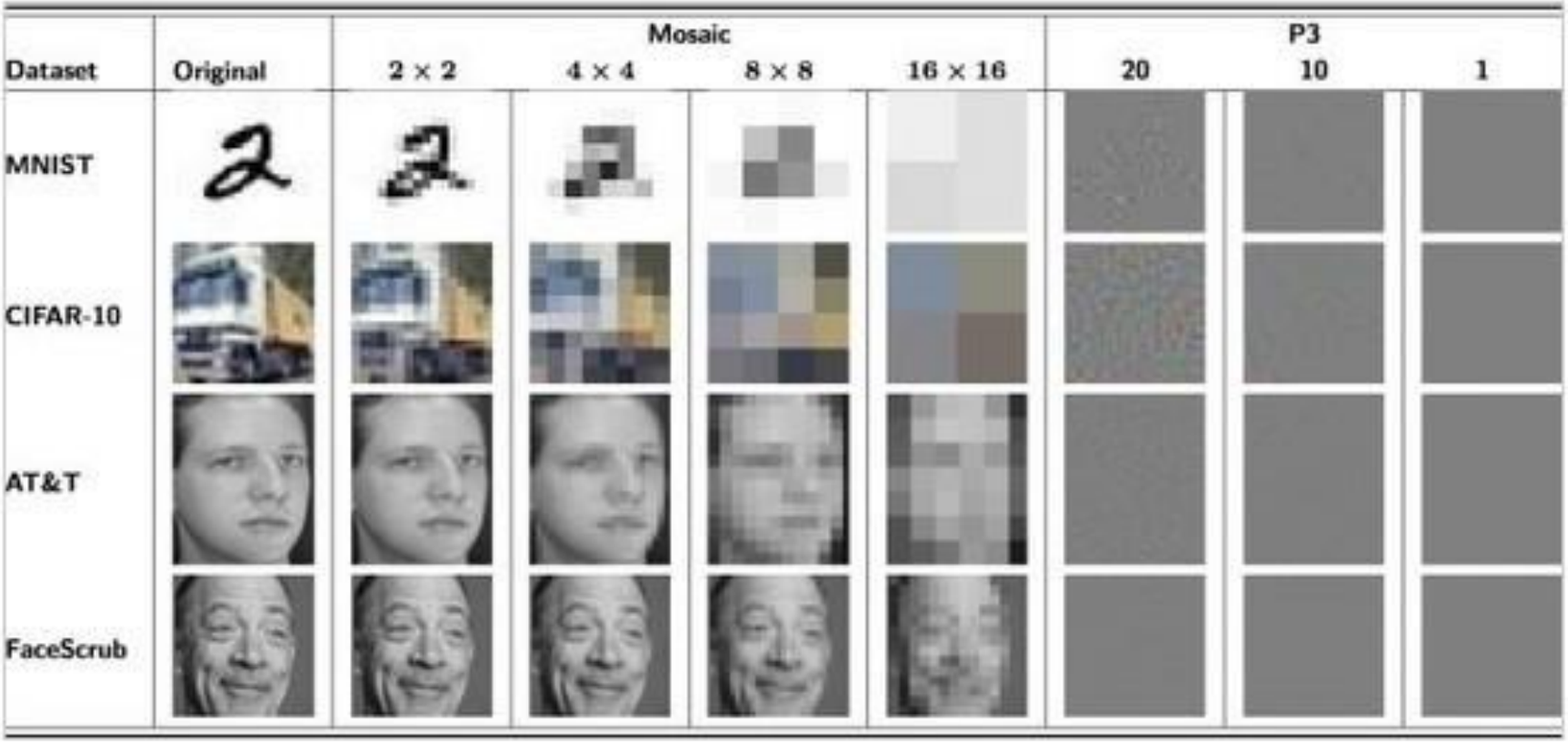
https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

De-mosaicing

- Generalized mosaics can be of two broad categories: large mosaic blurs and small mosaic blurs.
- Bulk mosaic blurring is often more difficult to handle. And small mosaic blur is easier than it is.

https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

De-mosaicing- Torch

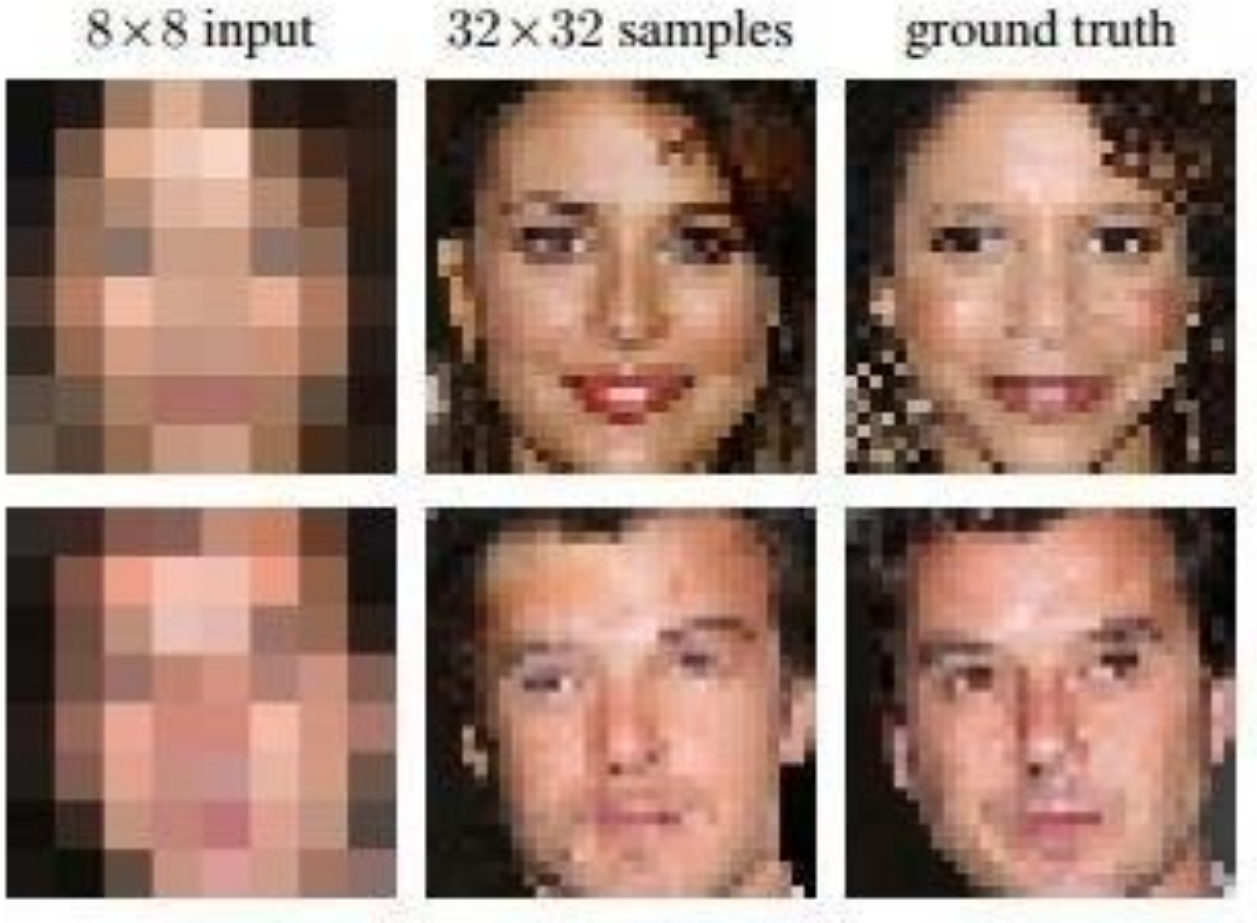


https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

Google: De-mosaicing

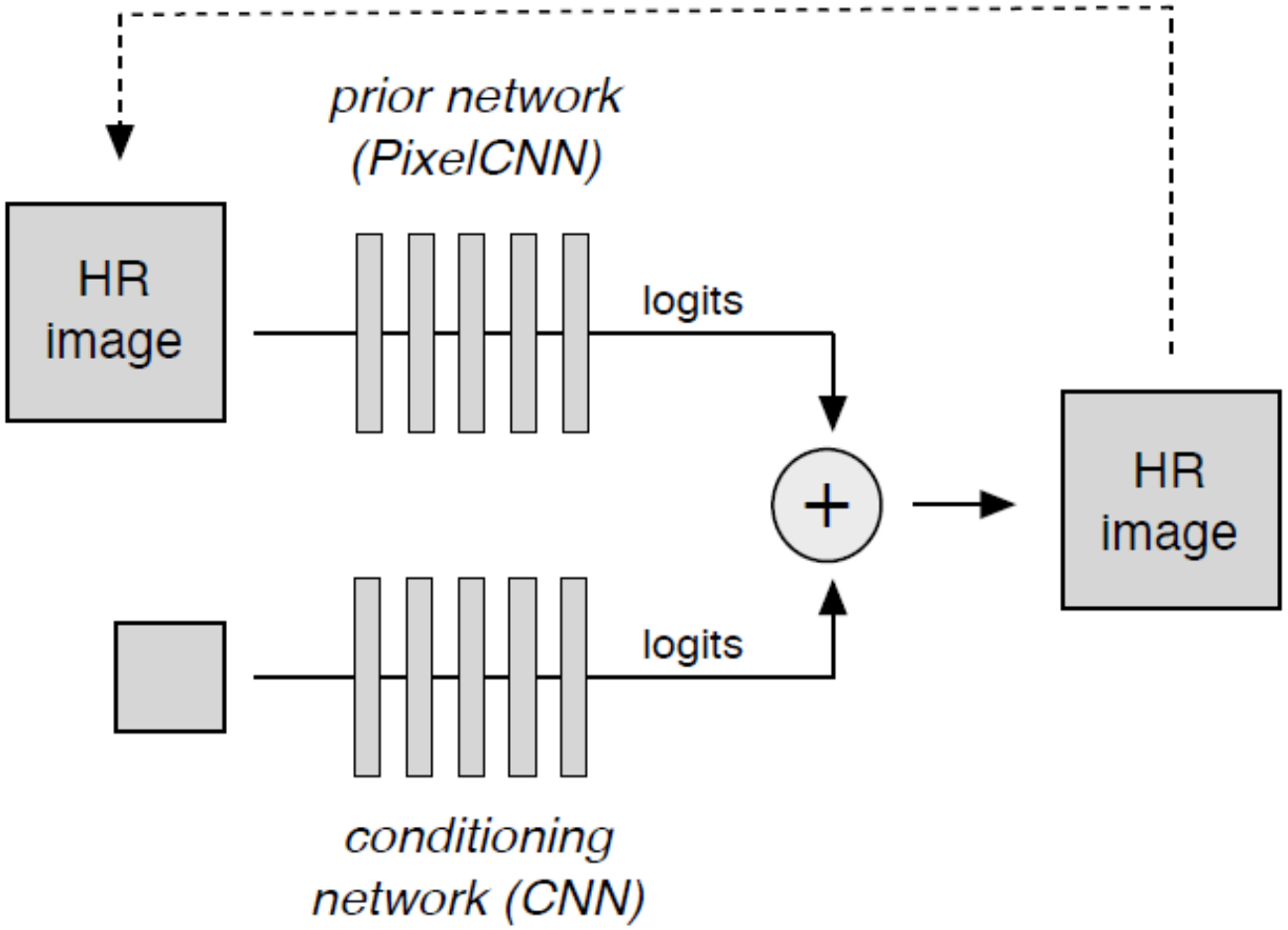
https://www.leiphone.com/news/201702/AcL7VAhJtdMhiwP5.html

Google: De-mosaicing



Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

Network

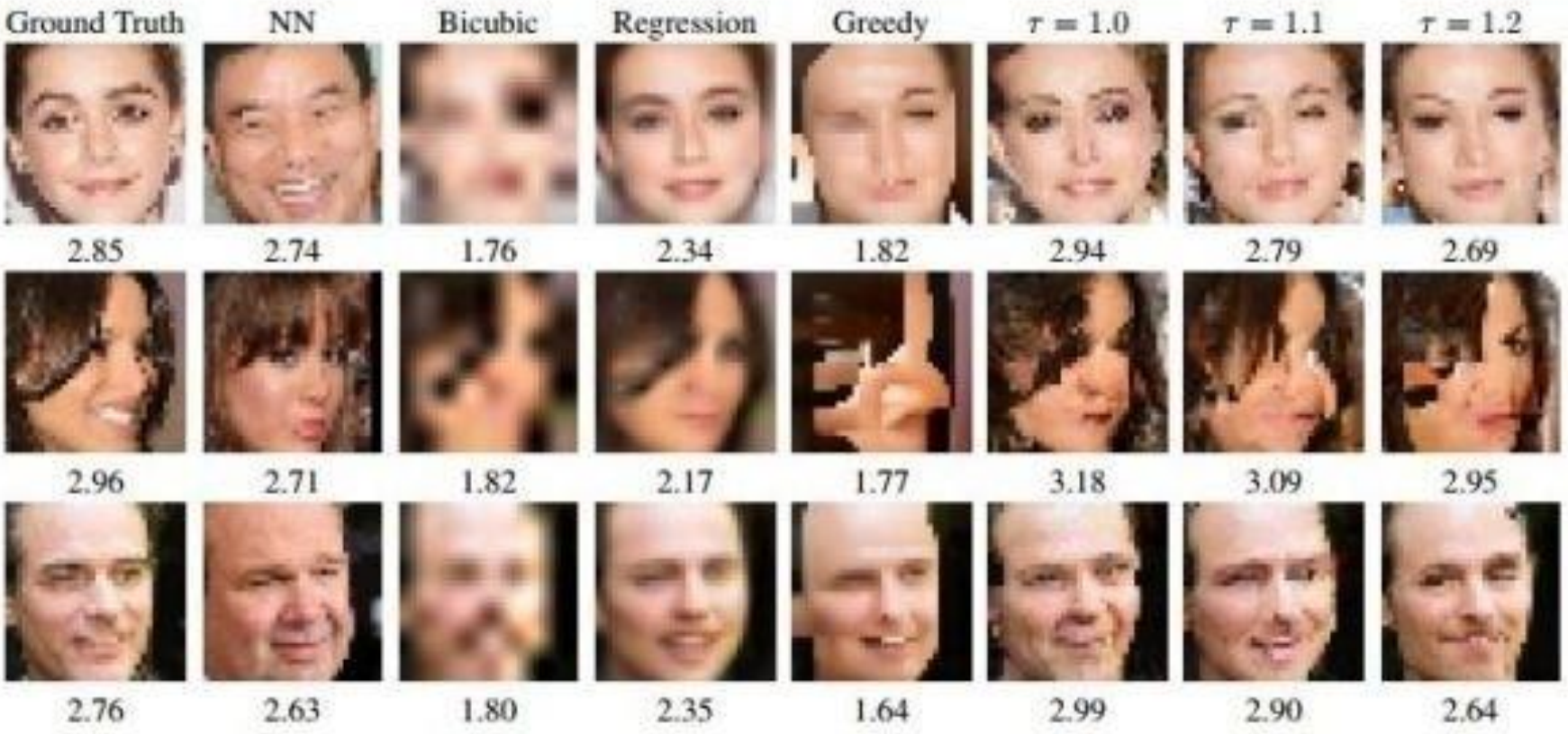


Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

Google: De-mosaicing

Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

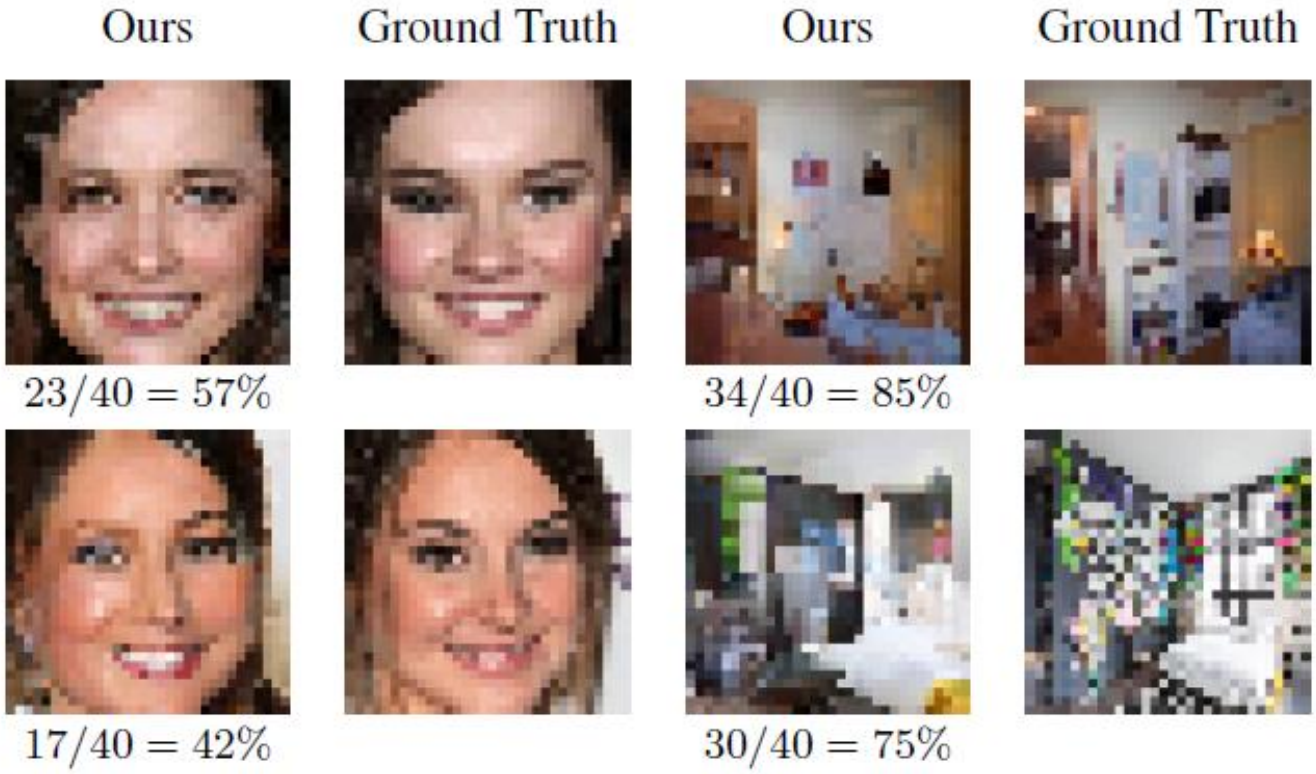
Google: De-mosaicing



Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

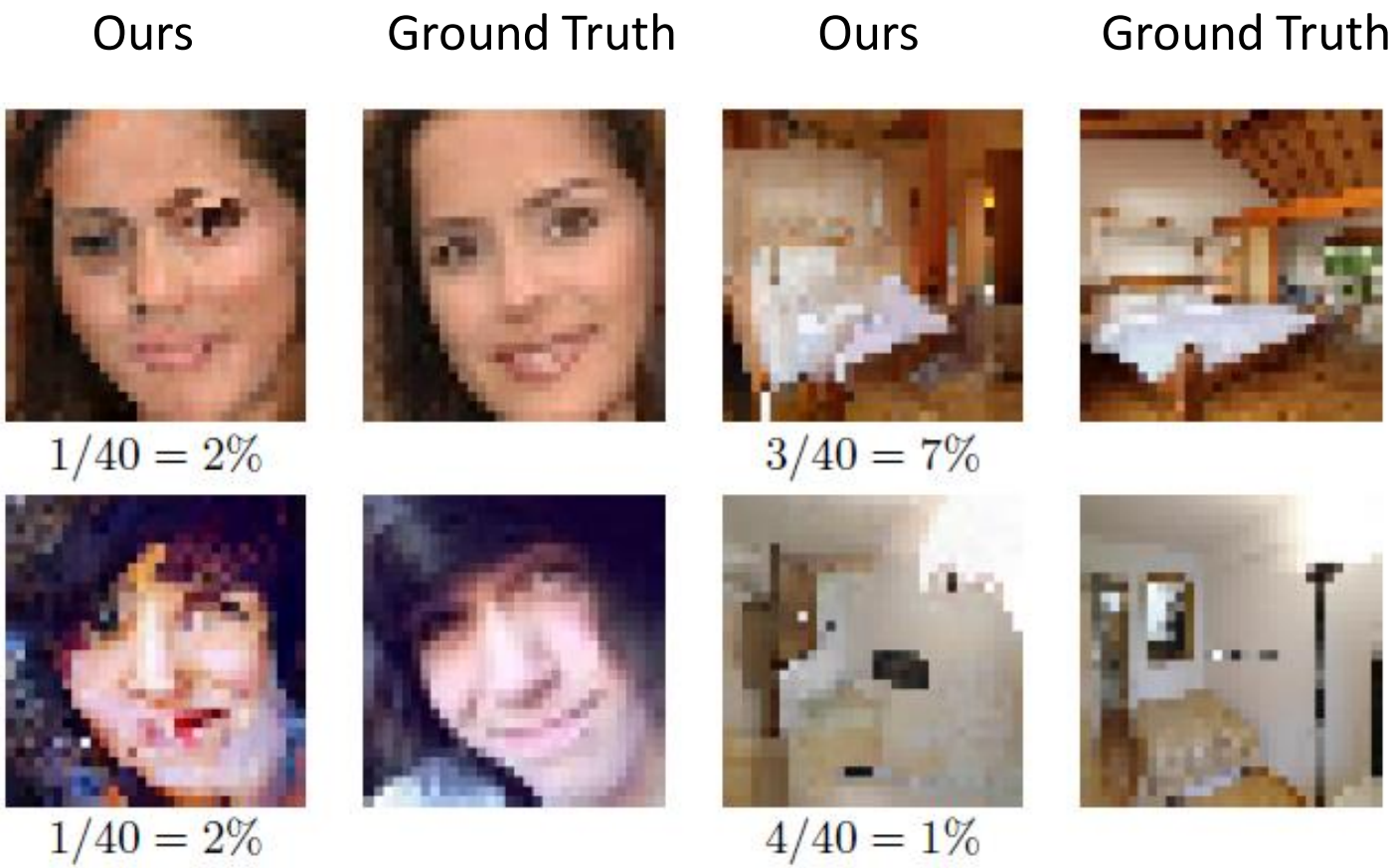
Samples images that performed best and worst in human ratings:

- The best and worst rated images in the human study. The fractions below the images denote how many times a person choose that image over the ground truth.



Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

Samples images that performed best and worst in human ratings:



Ryan Dahl , Mohammad Norouzi, Jonathon Shlens. Pixel Recursive Super Resolution

Applications of Image Inpainting

- Photo repair
- De-watermarking
- De-mosaicing
- ➢ **Object removal**
- De-blurring

http://www.baike.com/wiki/

Object removal

Remove unwanted targets

http://www.baike.com/wiki/

Object removal

- Idea: Remove object(s) from digital photographs, and then fill the hole with information extracted from the surrounding area.

- Result: Filled region should look “reasonable” to the human eyes.

https://www.slideshare.net/christojoy/image-inpainting-72607276

Example

Before

After

https://www.slideshare.net/christojoy/image-inpainting-72607276

Approach

Texture Synthesis

- Idea:
  - Sample color values of the surrounding area
  - Generate textures with sampling result to fill the hole
- Advantage:
  - Cheap and effective
  - No blur or other degradation
- Disadvantage:
  - May lose linear structure and composite textures

https://www.slideshare.net/christojoy/image-inpainting-72607276

Object removal

From the left (including objects, such as cars and other objects) to the right (after removing the target)

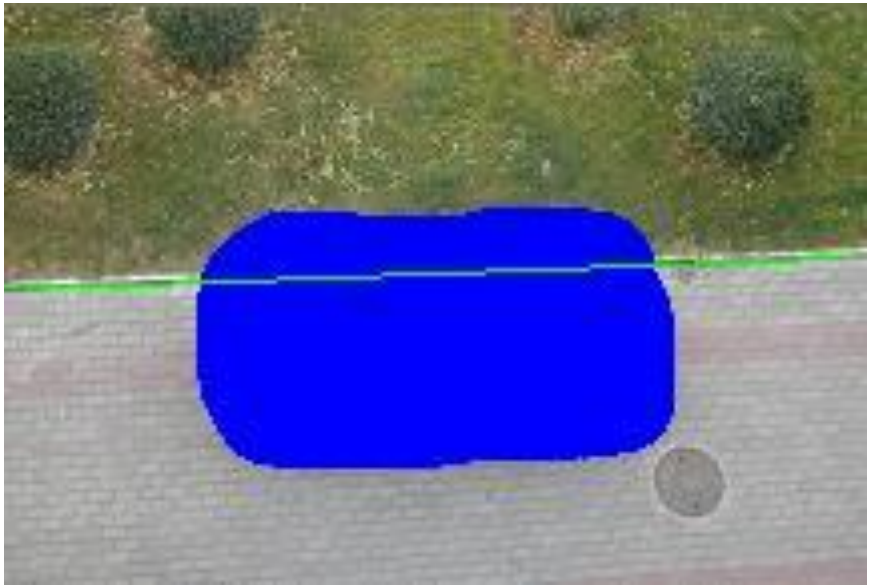
http://blog.sina.com.cn/s/blog_c190cb710102vslx.html

Object removal

- Erase the area in the image that you want to remove or repair

http://blog.sina.com.cn/s/blog_c190cb710102vslx.html

Object removal



- Draw contour guides to force the position of the contour to be repaired or automatically generated by the system.
- Smartly fills the area to be repaired, making the picture look intact and naturally flowing.

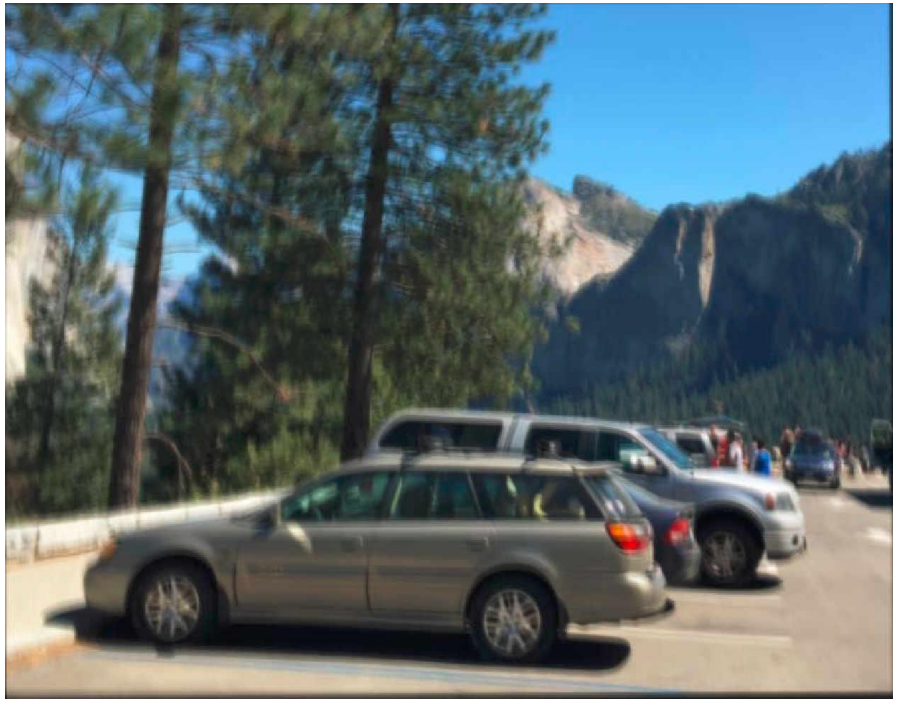
http://blog.sina.com.cn/s/blog_c190cb710102vslx.html

Applications of Image Inpainting

- Photo repair
- De-watermarking
- De-mosaicing
- Object removal
- ➢ **De-blurring**

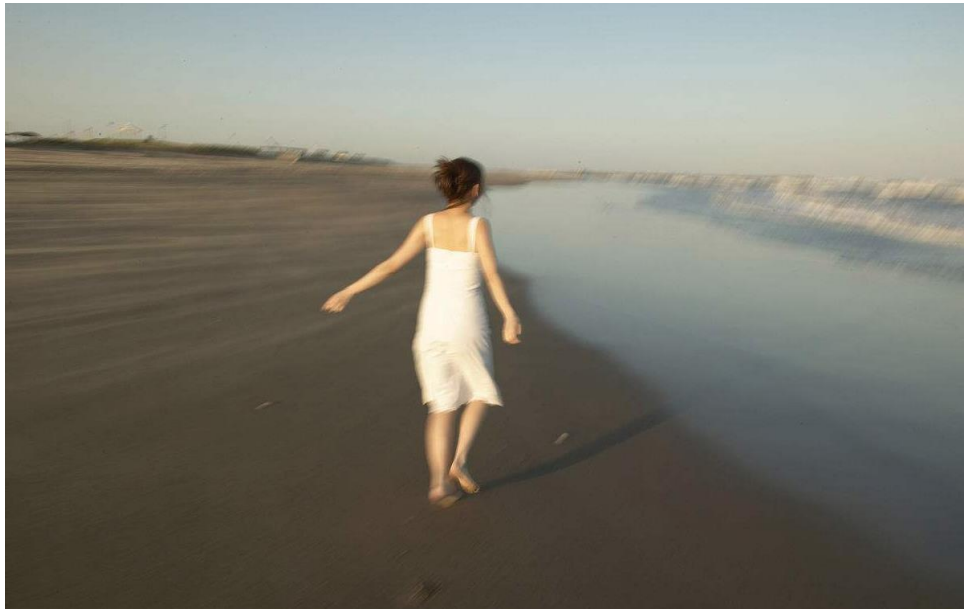
http://www.baike.com/wiki/

Deblurring



Lerenhan Li, Jinshan Pan, Wei-Sheng Lai, Changxin Gao, Nong Sang, Ming-Hsuan Yang. Learning a Discriminative Prior for Blind Image Deblurring. In CVPR 2018.

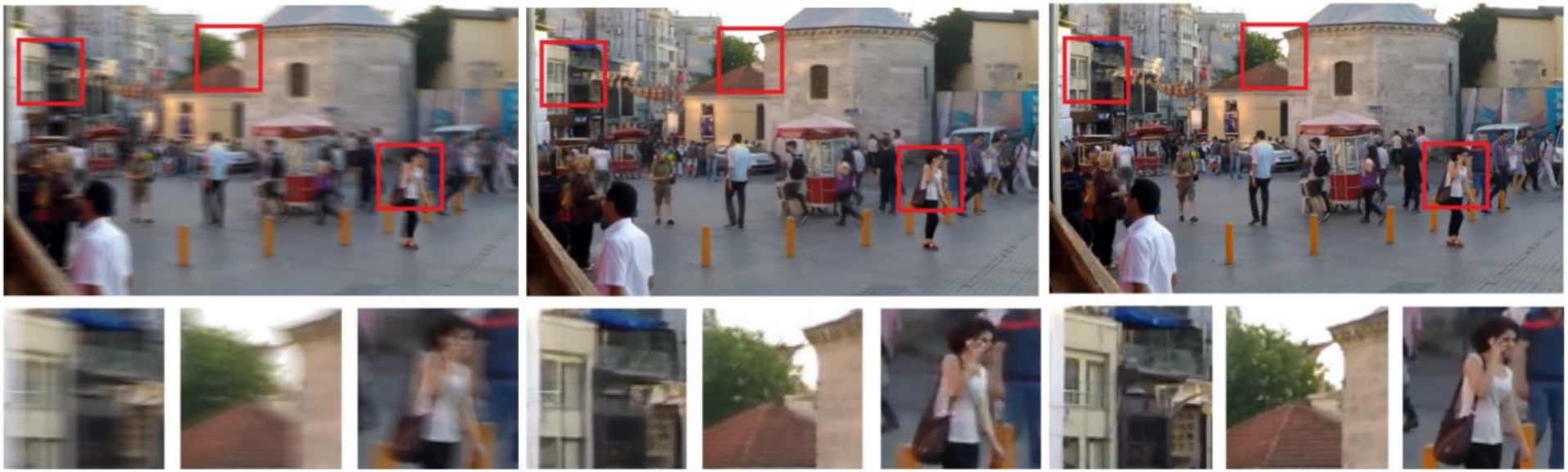
Motion blur



- Motion blur is a static scene or a series of pictures such as a movie or an object moving rapidly in an animation causing a noticeable blurry dragging trace

http://baike.baidu.com/

DeblurGAN

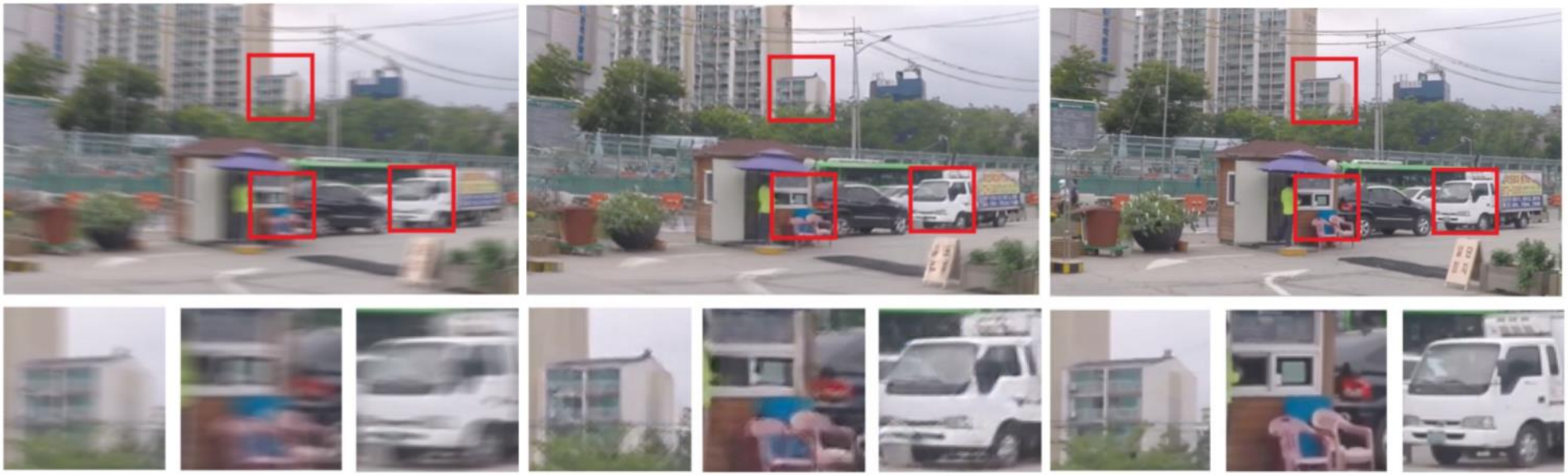


- An end-to-end learned method for motion deblurring. The learning is based on a conditional GAN and the content loss
- Achieves state-of-the-art performance both in the structural similarity measure and visual appearance.

DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

DeblurGAN

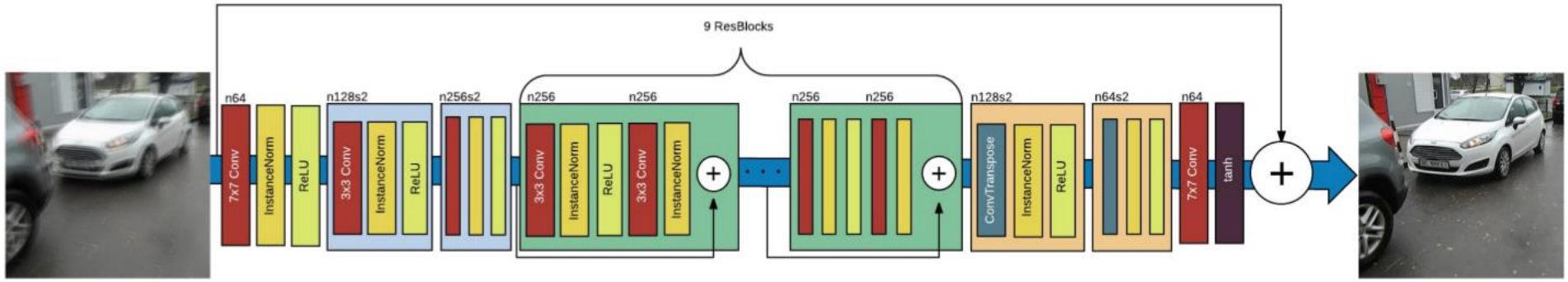
Blurred DeblurGAN Ground Truth



GoPro images processed by DeblurGAN

DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

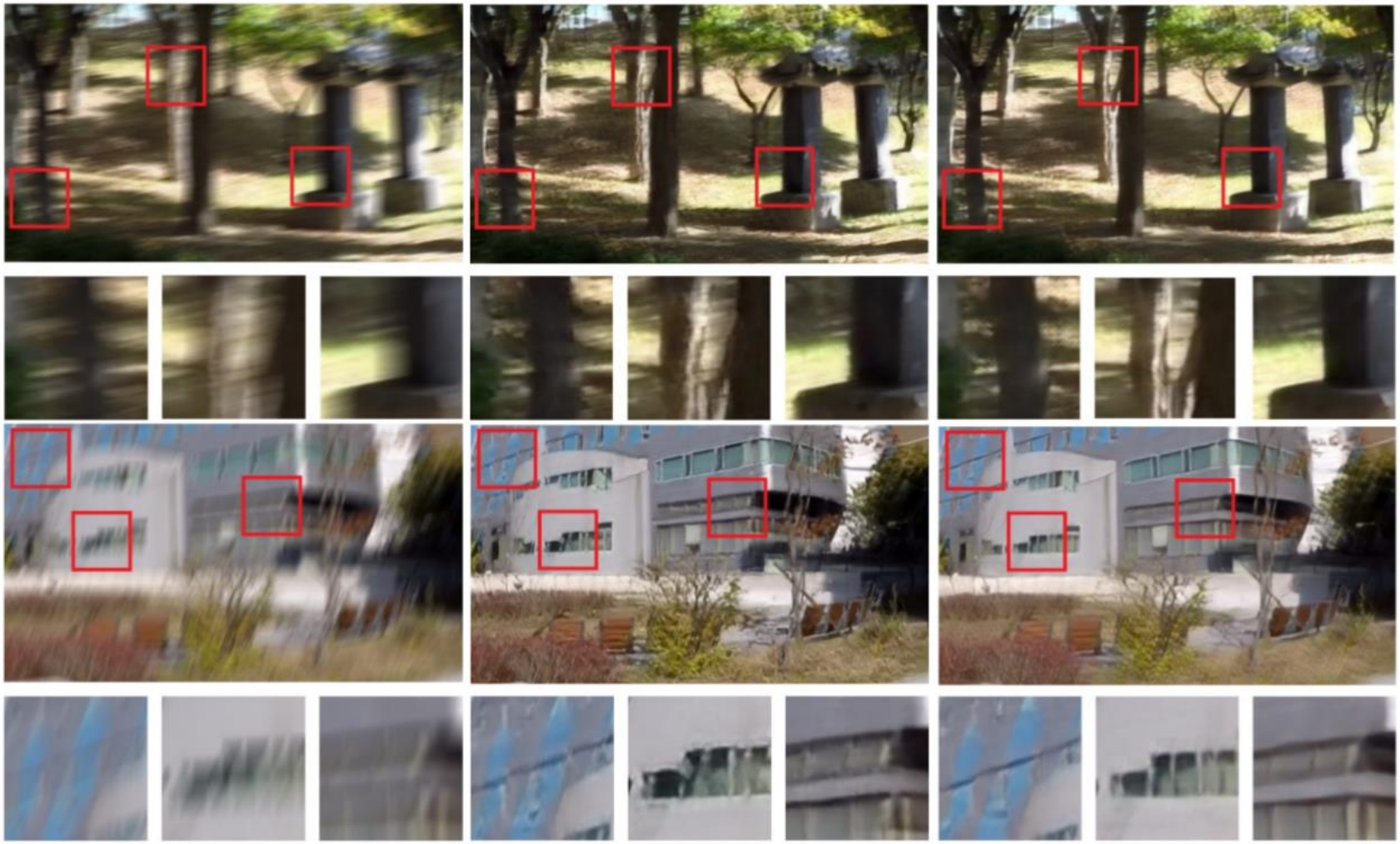
DeblurGAN -generator architecture



- Two strided convolution blocks with stride ½
- Nine residual blocks
- Two transposed convolution blocks
- Each ResBlock consists of a convolution layer, instance normalization layer, and ReLU activation

DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

Blurred Nah et al. [1] DeblurGAN



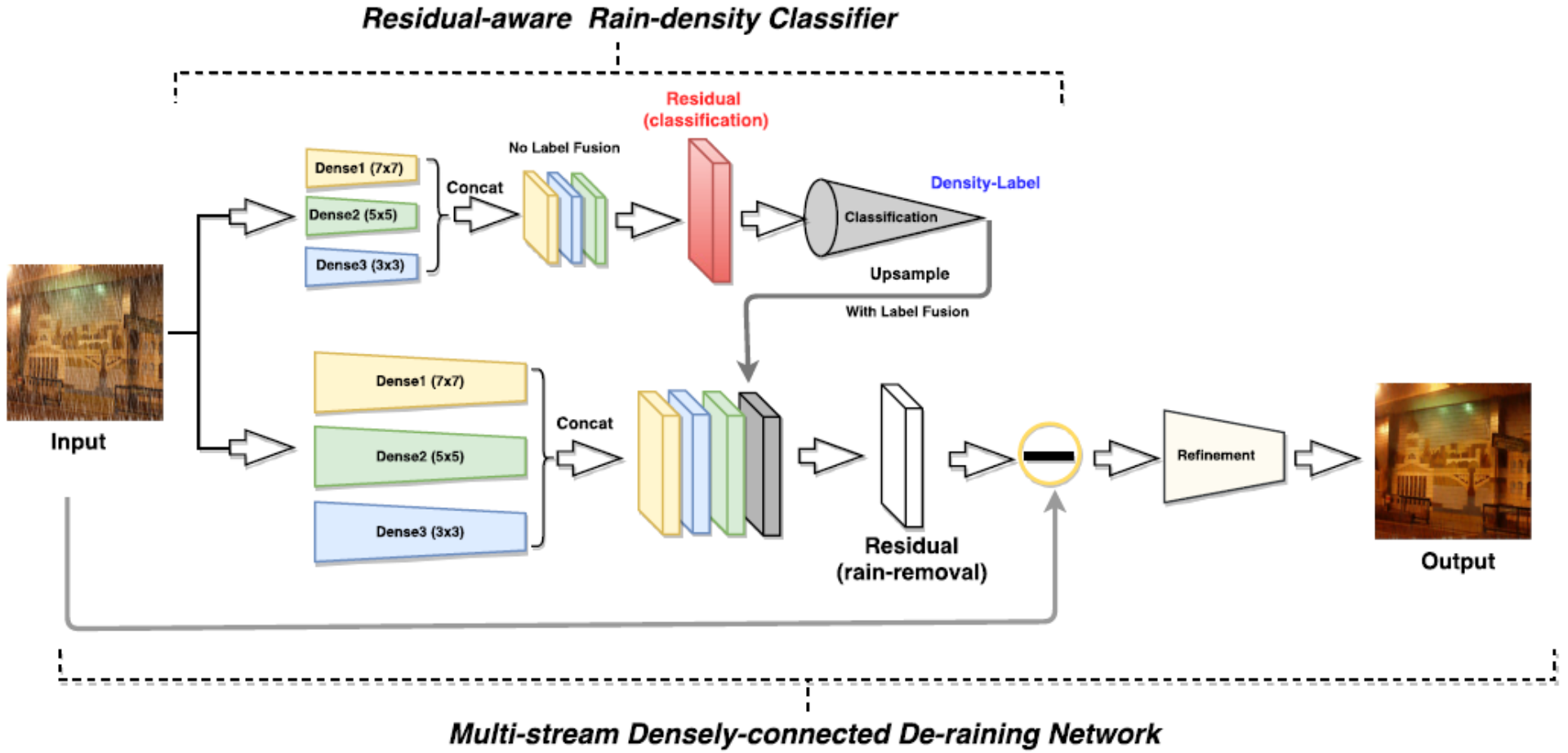
S. Nah, T. Hyun, K. Kyoung, and M. Lee. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring.2016

De-raining


- Single image de-raining
- Density-aware Image De-raining method using a Multi stream Dense Network (DID-MDN)
- Rain-density classification
- Rain streak removal

He Zhang Vishal M. Patel. Density-aware Single Image De-raining using a Multi-stream Dense Network. In CVPR, 2018.

DID-MDN



He Zhang Vishal M. Patel. Density-aware Single Image De-raining using a Multi-stream Dense Network. In CVPR, 2018.

Result

Input De-raining

He Zhang Vishal M. Patel. Density-aware Single Image De-raining using a Multi-stream Dense Network. In CVPR, 2018.

Image Quality Assessment

Image Quality Assessment

Image quality (often **Image Quality Assessment, IQA**) is a characteristic of an image that measures the perceived image degradation (typically, compared to an ideal or perfect image.

- Full-reference (FR) methods
- Reduced-reference (RR) methods
- No-reference (NR) methods

https://en.wikipedia.org/wiki/Image_quality

Image quality metrics

- Full-reference metrics

  - Peak Signal to Noise Ratio (PSNR)
  - Structural Similarity (SSIM)
  - Visual Information Fidelity (VIF)

https://en.wikipedia.org/wiki/Image_quality

Peak Signal to Noise Ratio (PSNR)

- **PSNR**, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

- Many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

PSNR is most easily defined via the mean squared error (MSE)
Given a noise-free $m \times n$ monochrome image I and its noisy approximation K, MSE is defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$$

The PSNR (in dB) is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^{\ 2}}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

MAX_I (=255) is the maximum possible pixel value.

https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

Structural Similarity (SSIM)

- SSIM is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos.
- SSIM is used for measuring the similarity between two images.
- SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE).

https://en.wikipedia.org/wiki/Structural_similarity

The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size $\mathrm{N}\times\mathrm{N}$ is:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$$

with:

- μ_x the average of x; μ_y the average of y;
- σ_x^2 the variance of x; σ_y^2 the variance of y;
- σ_{xy} the covariance of x and y;
- $c_1=(k_1L)^2, c_2=(k_2L)^2$ two variables to stabilize the division with weak denominator;
- L the dynamic range of the pixel-values;
- $k_1=0.01$ and $k_2=0.03$ by default

The SSIM index satisfies the condition of symmetry: $SSIM(x,y)=SSIM(y,x)$

https://en.wikipedia.org/wiki/Structural_similarity

PSNR & SSIM

PSNR:

- Mostly used, but its value does not reflect people's subjective feeling well.
- Value range : 20～40 dB
- The larger the value, the better the video quality.

SSIM :

- Complicated calculation, and its value can better reflect the subjective feelings of the human eye.
- Value range : 0～1
- The larger the value, the better the video quality.

https://blog.csdn.net/leixiaohua1020/article/details/11694369

3 Classical Methods to Image Inpainting

Inpainting: context and issues

Inpainting corresponds to filling holes (i.e. missing areas) in images

Let be an image I *defined as*

$$I : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$$

$n = 2$ for a 2D image

$m = 3$ for a (R,G,B) image

Let be a degradation operator M

$$M : \Omega \mapsto \{0,1\}$$

$$M(x) = \begin{cases} 0, & \text{if } x \in U \\ 1, & \text{otherwise} \end{cases}$$

$\Omega = S \cup U$

S being the known part of I

U the unknown part of I

Let F *the observed image:*

$$F = M \circ I$$

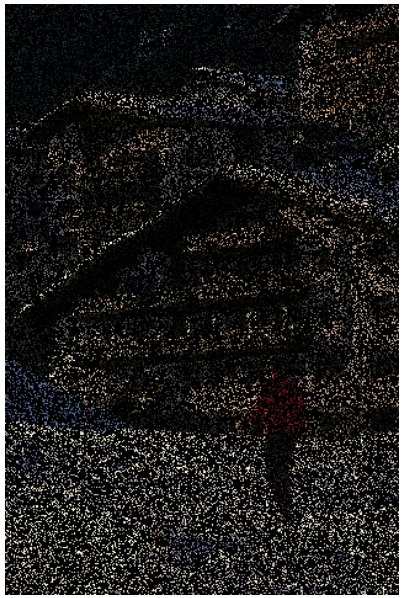
$\circ$ is the Hadamard product

M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRPAH 2000, 2000*.

Inpainting: context and issues

Different configurations according to the definition of M:

Original image



80% of the pixels have been removed.

Sparsity and low-rank methods

damaged portions in black, scratches

Diffusion-based methods

Object removal

Examplar-based methods

https://www.slideshare.net/OlivierLeMeur/o-le-meur-examplarbased-inpainting

Examplar-based Inpainting:

Texture synthesis

Examplar-based inpainting methods rely on the assumption that the known part of the image provides a good dictionary which could be used efficiently to restore the unknown part (Efros and Leung, 1999).

The recovered texture is therefore learned from similar regions.

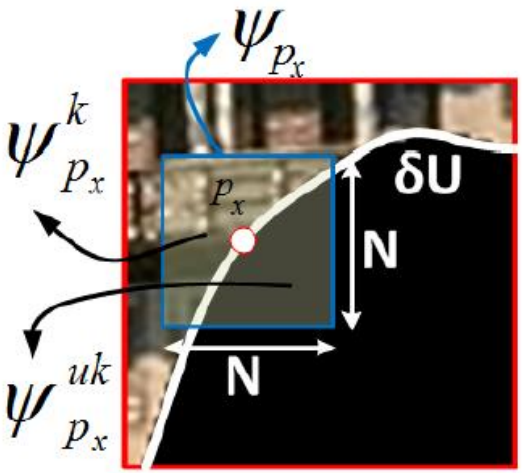
- This can be done simply by sampling, copying or combining patches from the known part of the image;

Template Matching

- Patches are then stitched together to fill in the missing area.

A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *CVPR, 1999.*

Notations:



$$\psi_{p_x} = \begin{bmatrix} I_1(0,0) \\ I_1(0,1) \\ \vdots \\ I_M(N-2,N-2) \\ I_M(N-1,N-1) \end{bmatrix}$$

- a patch ψ_{p_x} is a discretized $N \times N$ neighborhood centered on the pixel p_x. This patch can be vectorized in a raster-scan order as a mN^2-dimensional vector;
- $\psi^{uk}_{p_x}$ denotes the unknown pixels of the patch;
- $\psi^{k}_{p_x}$ denotes its known pixels;
- $\psi_{p_{x(i)}}$ denotes the i^{th} nearest neighbour of ψ_{p_x};
- δU is the front line;

https://www.slideshare.net/OlivierLeMeur/o-le-meur-examplarbased-inpainting

Examplar-based Inpainting:

Criminisi et al.'s algorithm

Criminisi et al. (Criminisi et al., 2004) has brought a new momentum to inpainting applications and methods. They proposed a new method based on two sequential stages:

- Filling order computation
- Texture synthesis

A. Criminisi, P. P´erez, and K. Toyama. Region filling and object removal by examplar-based image inpainting. *IEEE Trans. On Image Processing, 13:1200–1212, 2004.*

Filling order computation:

$$P(p_x) = C(p_x) \times D(p_x)$$

Confidence term

$$C(p_x) = \frac{\sum_{q \in \psi_{p_x}^k} C(q)}{|\psi_{p_x}|}$$

Where $|\Psi_{p_x}|$ is the area of Ψ_{p_x}

Data term

$$D(p_x) = \frac{|\nabla I^{\perp}(p_x) \cdot n_{p_x}|}{\alpha}$$

Where α is the a normalization constant in order to ensure that $D(p_x)$ is in the range 0 to 1

https://www.slideshare.net/OlivierLeMeur/o-le-meur-examplarbased-inpainting

Texture synthesis:

A template matching is performed within a local neighborhood:

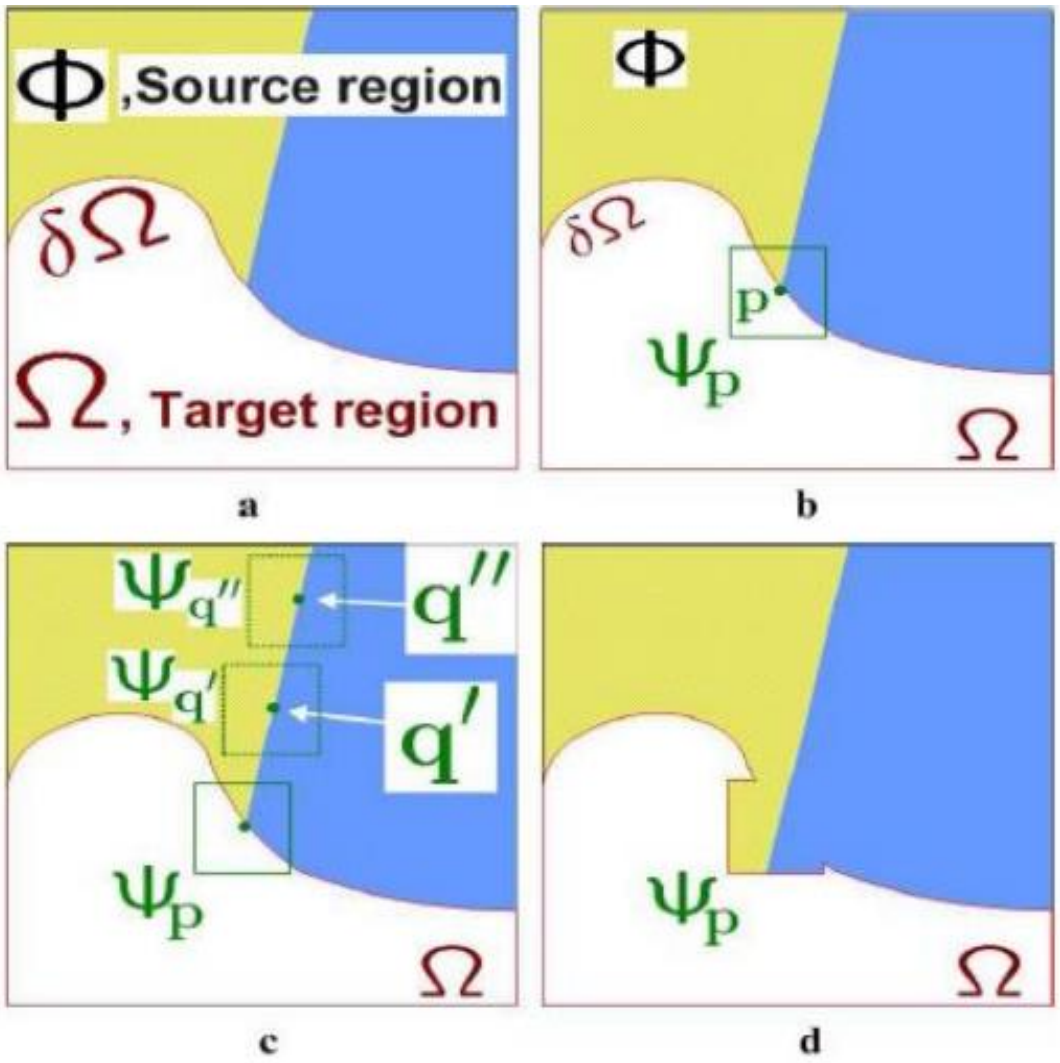
$$p_y = \arg\min_{q \in W} d(\Psi^k_{p_q}, \Psi^k_{p_{x^*}})$$

- $W \subseteq S$ *is the window search;*
- $\Psi^k_{p_{x^*}}$ are the known pixels of the patch $\Psi_{p_{x^*}}$ with the highest priority;
- $\Psi^k_{p_y}$ are the known pixels of the nearest patch neighbor;
- $d(a, b)$ is the sum of squared differences between patches a and b.

The pixels of the patch $\Psi^{uk}_{p_y}$ *are then copied into the unknown pixels* of the patch $\Psi_{p_{x^*}}$

https://www.slideshare.net/OlivierLeMeur/o-le-meur-examplarbased-inpainting

Texture synthesis



(a) Original image

(b) Repaired patch

(c) Match and search

(d) Copy or synthesis

A Criminisi, P Perez and K Toyama: Region Filling and Object Removal by Exemplar-Based Inpainting. IEEE Transactions on Image Processing(TIP),2004.

Some examples:

Inpainted pictures with (Criminisi et al., 2004)'s method:

https://www.slideshare.net/OlivierLeMeur/o-le-meur-examplarbased-inpainting

Development

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, Dan B Goldman.
PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing
ACM Transactions on Graphics (Proc. SIGGRAPH), 2009.

[2] Connelly Barnes, Eli Shechtman, Dan B Goldman, Adam Finkelstein.
The generalized PatchMatch correspondence algorithm.
In Proceedings of European Conference on Computer Vision (ECCV), 2010.

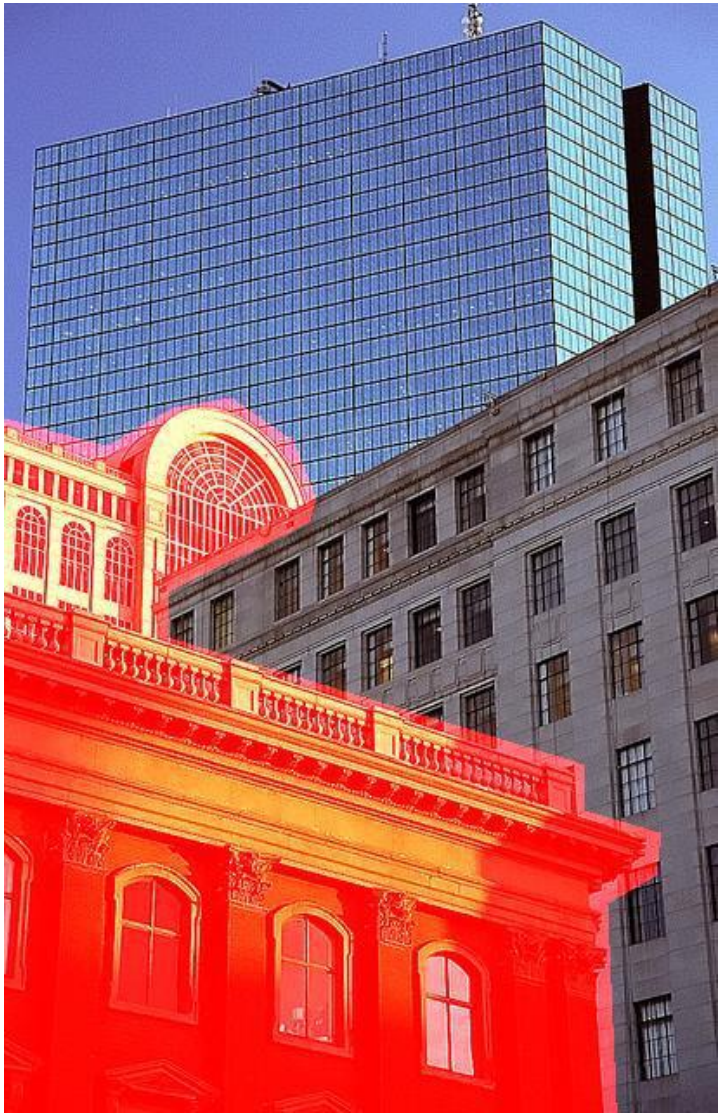
[3] S Darabi, E Shechtman, C Barnes and et al.
Image melding: combining inconsistent images using patch-based synthesis.

[4] K He and J Sun. **Image completion approaches using the statistics of similar patches.**
IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2014.

[5] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, Johannes Kopf.
Image Completion using Planar Structure Guidance.
In SIGGRAPH, 2014.

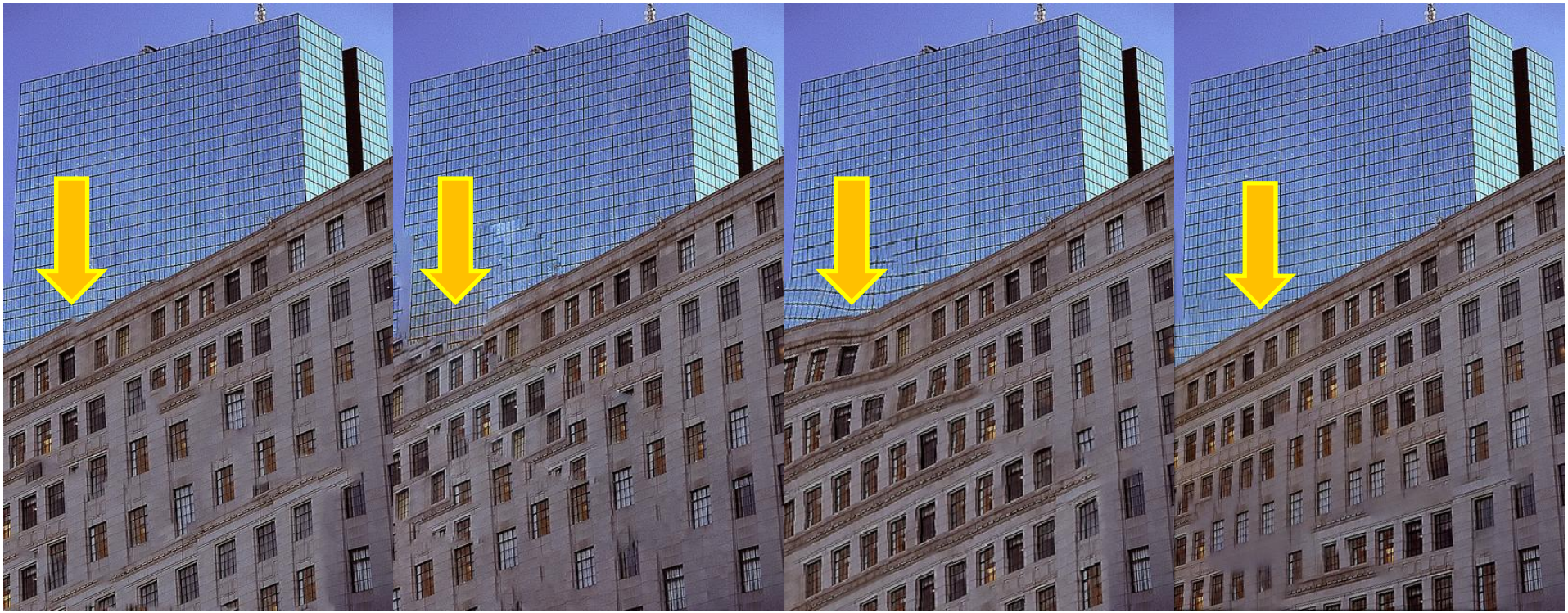
https://mp.weixin.qq.com/s/lPhg-pvC-6GofMuxTF6DnQ

Input and hole



https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Result



Photoshop CAF Patch Match [Barnes et al. 2009]

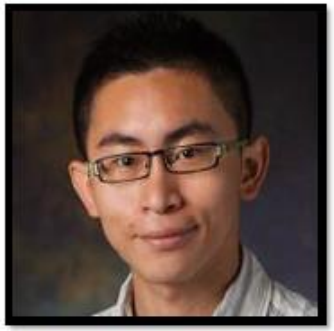
Statistics of Patch Offsets [He and Sun 2014]

Image Melding [Darabi et al. 2012]

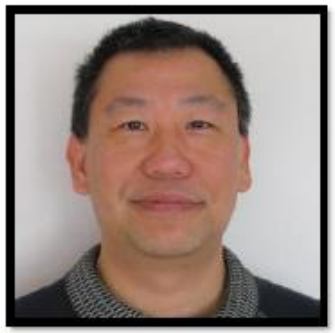
Planar Structure Guidance [Huang et al. 2014]

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

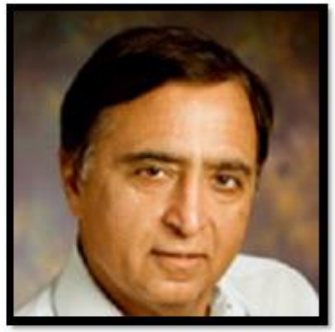
Image Completion using Planar Structure Guidance



Jia-Bin Huang



Sing Bing Kang



Narendra Ahuja

Johannes Kopf

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Microsoft Research

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion

Input image

Region to fill

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Photoshop Content Aware Fill
[Wexler et al. 2007] [Barnes et al. 2009]

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Statistics of Patch Offsets
[He and Sun ECCV 2012]

Translational patches are not sufficient!

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Melding
[Darabiet al. 2012]

Searching patch transformation space is HARD!

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Planar Structure Guidance

Use mid-level information

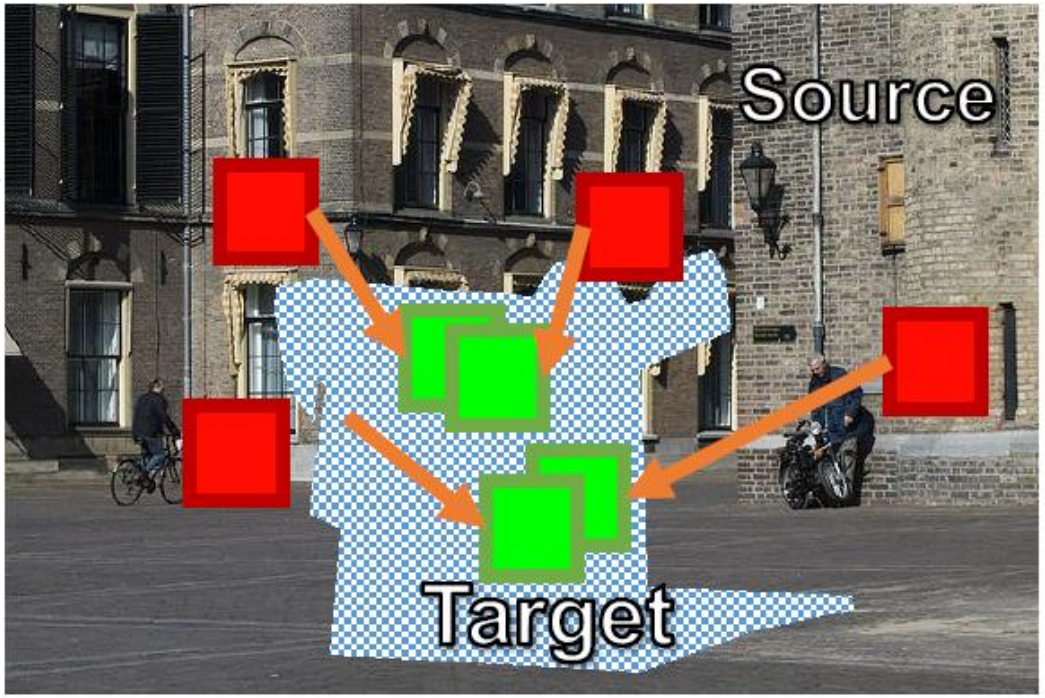
https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion using Planar Structure Guidance

- Based on non-parametric framework of [Wexler et al. 2007]
- The basic form $\min_{\{t_i, s_i\}} \sum E_{color}(t_i, s_i)$
- Patch matching cost

$E_{color}(t_i, s_i) = ||$  $-$  $||_1$

- Patch Match [Barnes et al. 2009] for efficient nearest neighbor field search.

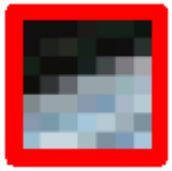


https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion using Planar Structure Guidance

- Low-level processing is not sufficient
- Idea: guide the completion using mid-level information!
- Transformation of source patches

- Translational (Photoshop CAF)



- Rotation and scale (Image Melding)

- Projective transform (Planar Structure Guidance)

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion using Planar Structure Guidance

Mid-Level cues for guidance

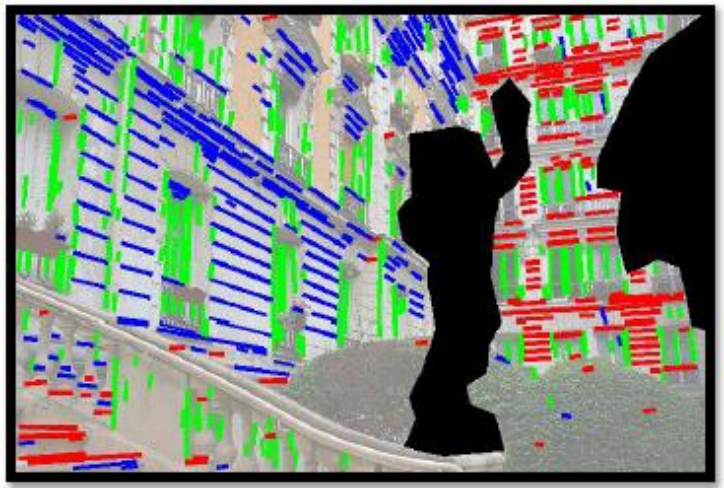
Input and Hole

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

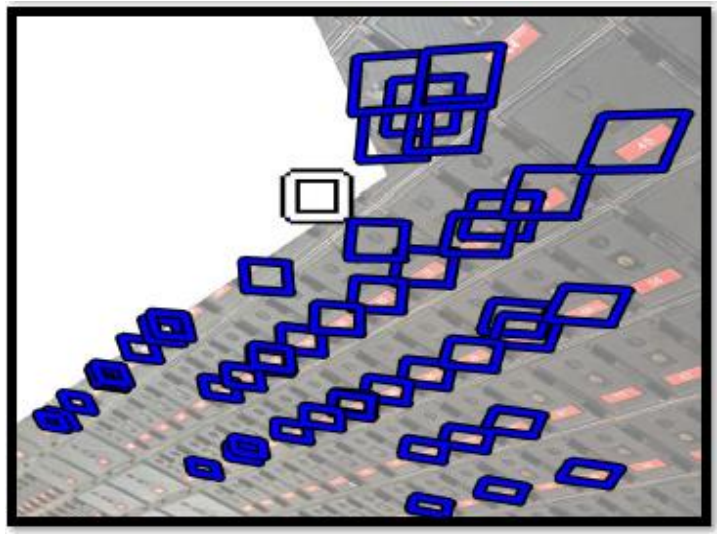
Image Completion using Planar Structure Guidance

Mid-Level cues for guidance

- Planes:
  Cues for how to deform patches
  Plane detection: vanishing point detection and posterior probability



- Regularity:
  Cues for where to sample patches
  Regularity detection:
  Feature detection and matching
  Mode detection



https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion using Planar Structure Guidance

Result

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Result Comparison

Photoshop Content Aware Fill

Statistics of Patch Offsets

Image Melding

Planar Structure Guidance

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Image Completion using Planar Structure Guidance

Regularity detection

With planar guidance only

Plane and regularity

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

What if there are no structures?

- Reduces to baseline image completion algorithms which use translational patches (same as Photoshop content-aware fill)

$$\min_{\{\mathbf{t_i},\mathbf{s_i}\}} \sum \mathbf{E}_{\text{color}}(\mathbf{t_i},\mathbf{s_i}) + \mathbf{E}_{\text{guide}}(\mathbf{t_i},\mathbf{s_i})$$

- Works well with man-made scenes
- Backward-compatible with natural scenes

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

What if there are no structures?

Input and Mask

Completion

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

What if there are no structures?

Input and Mask

Completion

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

What if there are no structures?

Input and Mask

Completion

https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Project Website

- http://bit.ly/planarcompletion

- Full comparison of 110 images (85 urban scenes + 25 natural scenes)

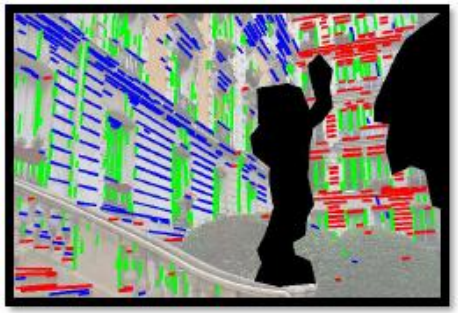
- Code

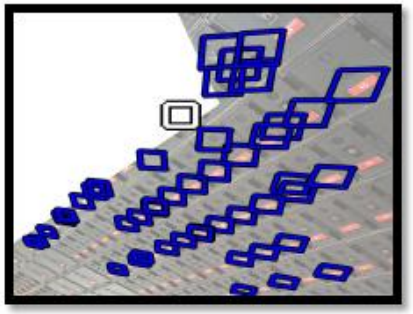
https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

Conclusions

- Mid-level constraints for the win!

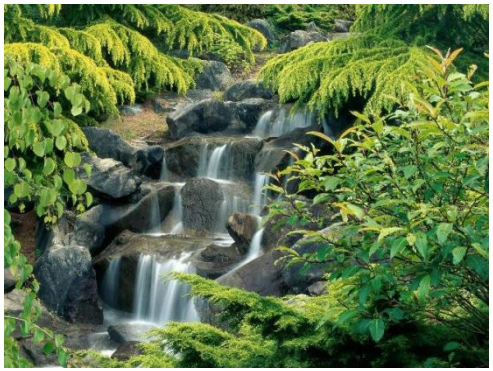
  Our work: planes and regularity





- Great for man-made scenes

- Backward-compatible with natural scenes



https://www.slideshare.net/jbhuang/image-completion-using-planar-structure-guidance-siggraph-2014

OpenCV-Image Inpainting

Function: **Inpaint( )**

```
void inpaint( InputArray src, InputArray inpaintMask,
              OutputArray dst, double inpaintRadius, int flags);
```

- **src**: input single-channel or RGB image
- **inpaintMask**: the mask of image, single-channel image,
  the size is the same as the original image,
  the pixels of the mask are all 1, others are all 0.
- **dst**: repaired image
- **inpaintRadius**: the radius of the neighborhood taken by the repair algorithm and is used to calculate the difference of the current pixel.
- **flags**: repair algorithm, INPAINT_NS and INPAINT_TELEA;

https://www.cnblogs.com/skyfsm/p/6888213.html

Results

Input and Mask

Inpainted

Other parts have changed color!

https://www.cnblogs.com/skyfsm/p/6888213.html

Results

Input and Mask

Inpainted

The repair effect is good!

https://www.cnblogs.com/skyfsm/p/6888213.html

Results

Input and Mask

Inpainted

The complex background is not well repaired!

https://www.cnblogs.com/skyfsm/p/6888213.html

Results

Input and Mask

Inpainted

In order not to affect images outside the area, artificially limit the areas that need to be repaired

https://www.cnblogs.com/skyfsm/p/6888213.html

Results

Input and Mask

Inpainted

The background part of the repair effect is good, but the repair of the edge details is not good!

https://www.cnblogs.com/skyfsm/p/6888213.html

4 Deep Learning to Image Inpainting

Datasets

- CelebA
- MS COCO
- PASCAL VOC
- ImageNet
- Paris Street View
- SVHN
- Car Dataset

Large-scale CelebFaces Attributes (CelebA) Dataset

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Details

- **CelebFaces Attributes Dataset (CelebA)** is a large-scale face attributes dataset with more than **200K** celebrity images, each with **40** attribute annotations.
- Images cover large pose variations and background clutter.
- CelebA has large diversities, large quantities, and rich annotations, including
  - **10,177** number of **identities**
  - **202,599** number of **face images**
  - **5 landmark locations**, **40 binary attributes** annotations per image

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Application

The dataset can be employed as the training and test sets for the following computer vision tasks:

- Face attribute recognition
- Face detection
- Landmark (or facial part) localization

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Sample Images

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Sample Images

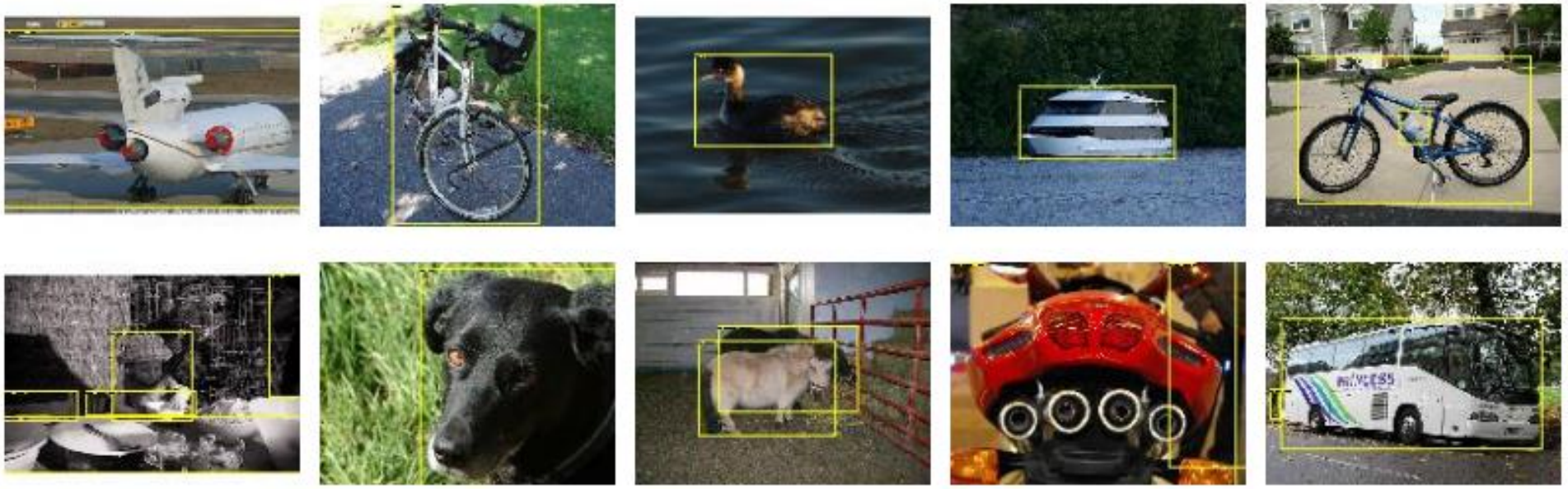
http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

MS COCO

- 102 classes
- 165,482 training images
- 81,208 validation images
- 81,434 test instances

http://host.robots.ox.ac.uk/pascal/VOC/

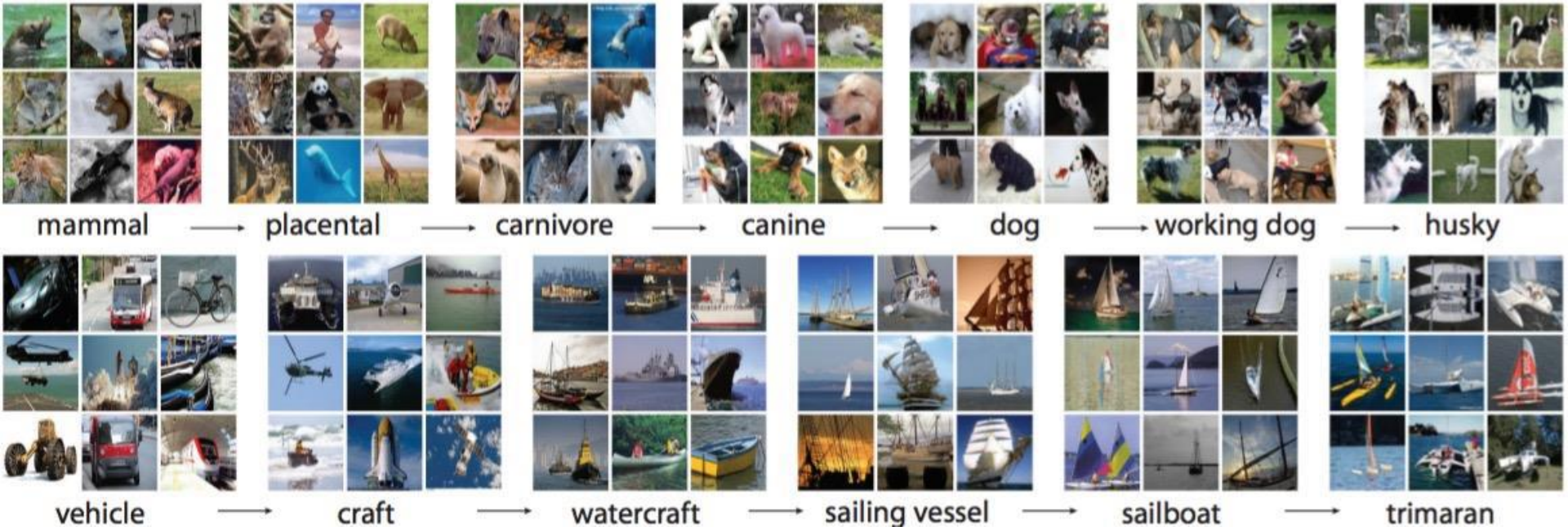
PASCAL VOC 2012



- 20 classes
- 11,530 train/val images
- 27,450 object instances
- 6,929 segmentations

http://host.robots.ox.ac.uk/pascal/VOC/

ImageNet-1000



- 1000 classes
- 1.3M training images
- 100K testing images

http://www.image-net.org/

Paris Street View

- 15,000 images

C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? Communications of the ACM, 58(12):103–110, 2015.

The Street View House Numbers(SVHN)

- 73,257 training images
- 26,032 test images

http://ufldl.stanford.edu/housenumbers/

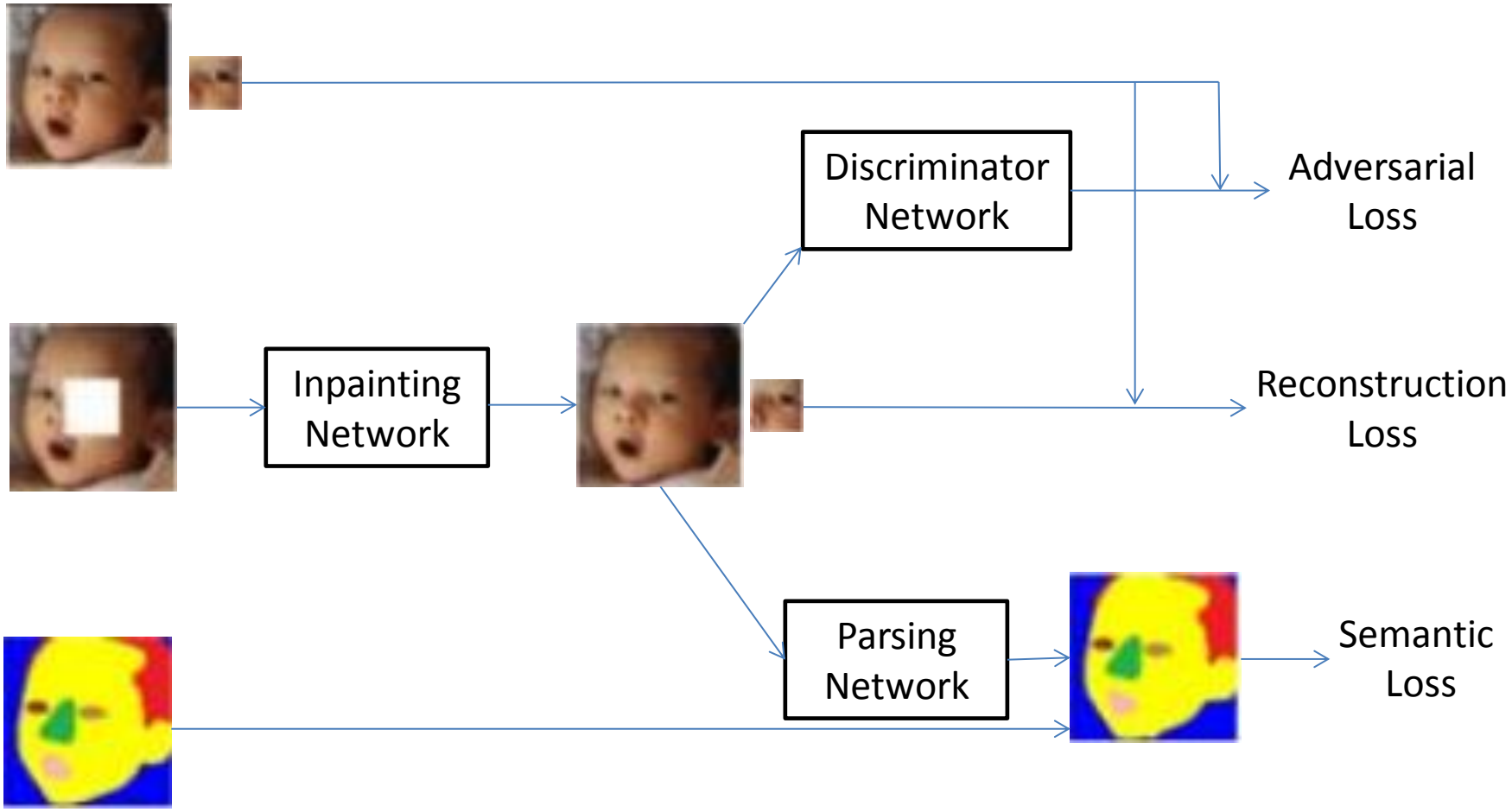
Cars Dataset

- 196 classes of cars
- 8,144 training images
- 8,041 testing images

http://ai.stanford.edu/~jkrause/cars/car_dataset.html

Methods in Recent Papers

Image Inpainting Framework— Deep Learning



https://mp.weixin.qq.com/s/lPhg-pvC-6GofMuxTF6DnQ

- **Globally and Locally Completion**
- Semantic Image Inpainting
- High-Resolution Image Inpainting
- Content Encoder

Globally and Locally Consistent Image Completion

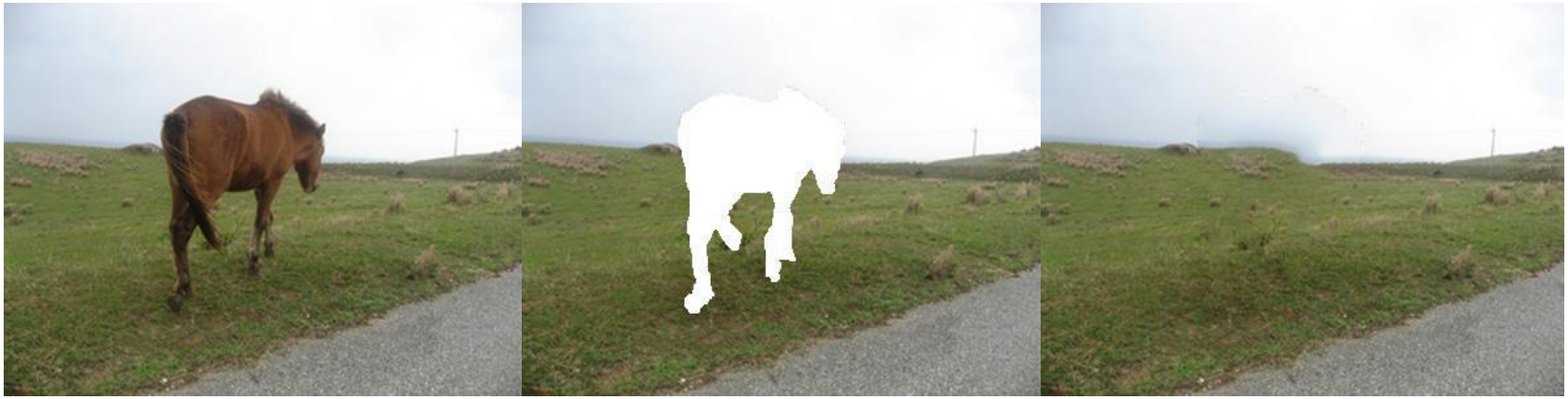
Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa
Waseda University. In SIGGRAPH ,2017.

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Image Completion

- Filling in a part of images
  - Object removal
  - Texture generation for occluded regions
  - …



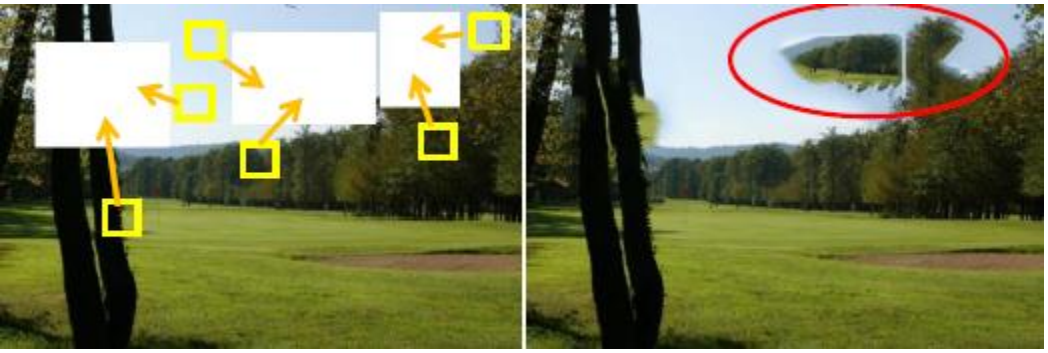
Input Mask Image Completion

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Related Work

- Patch-based inpainting
  - Synthesize texture by collecting small image patches
  - Cannot preserve global structures
  - Cannot generate novel objects



Input Output

Input Output

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Related Work

- Learning-based inpainting
  - Learning inpainting with GAN
  - Fixed image resolution(128 × 128 pixels)
  - Fixed mask position and size (center, 64 × 64 pixels)
  - Tends to generate texture that is inconsistent with an input image

Input Output Input Output

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Our Method

- Novel network for globally and locally consistent image completion
  - Completion network that is able to inpaint arbitrary regions
  - Adversarial training with two auxiliary networks
  - Can generate novel objects




Input Output Input Output

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion




Input

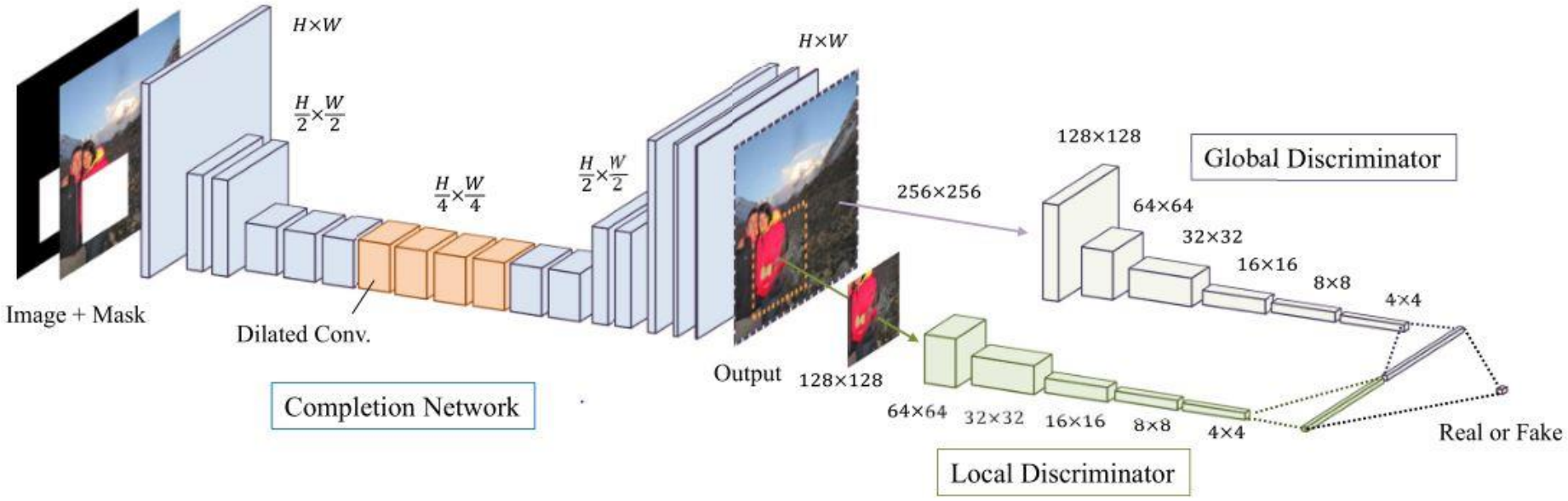
Completion

Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. In SIGGRAPH ,2017

Demo

http://hi.cs.waseda.ac.jp/~iizuka/projects/completion/ja/

Overview of architecture

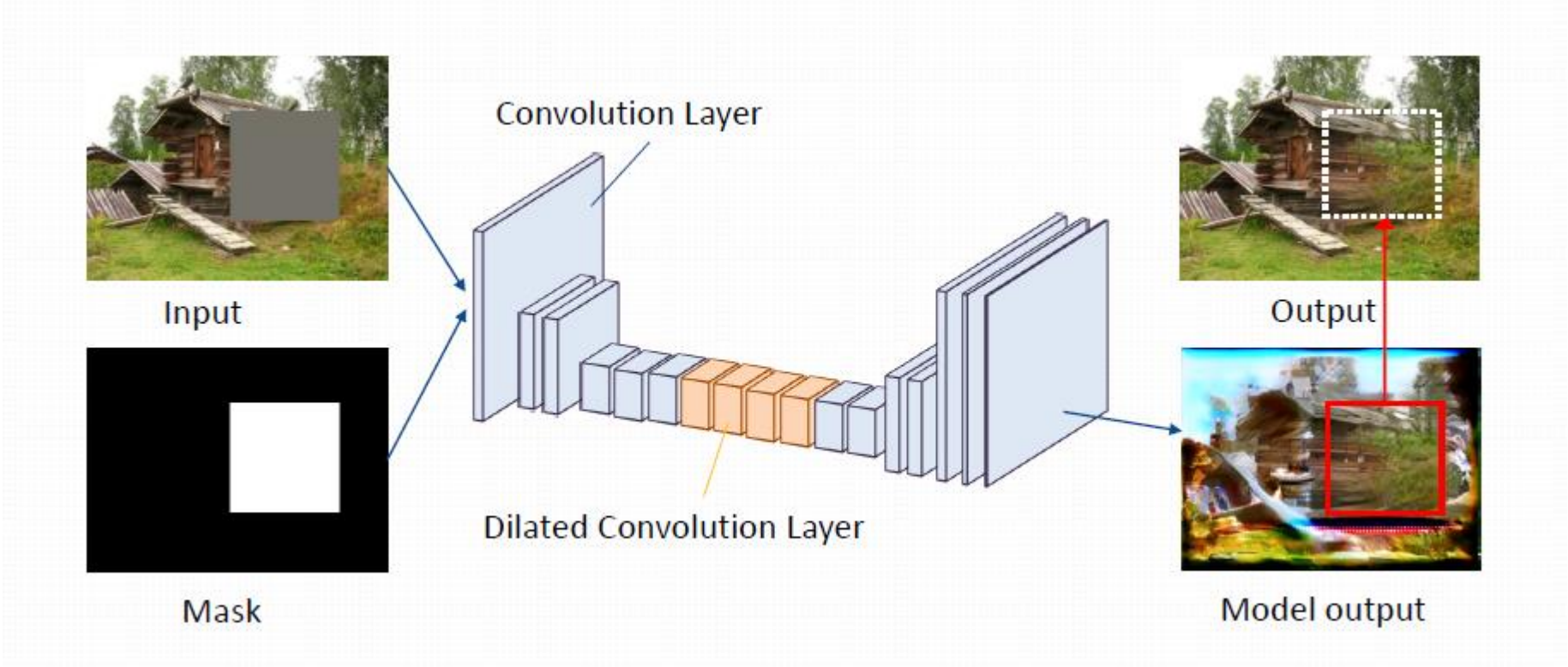


GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Completion Network

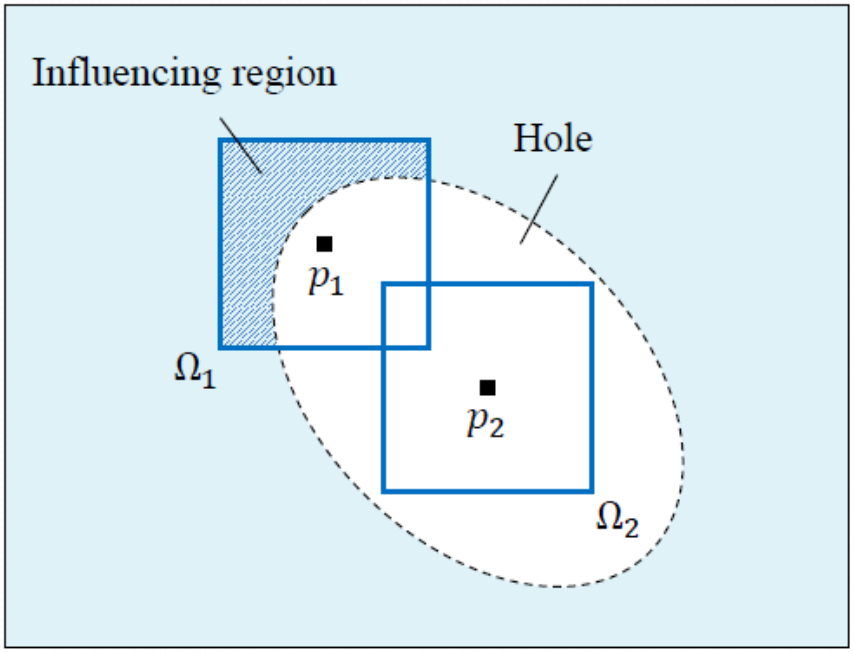
- Composed of 17 convolutional layers
- Output an image filled in holes



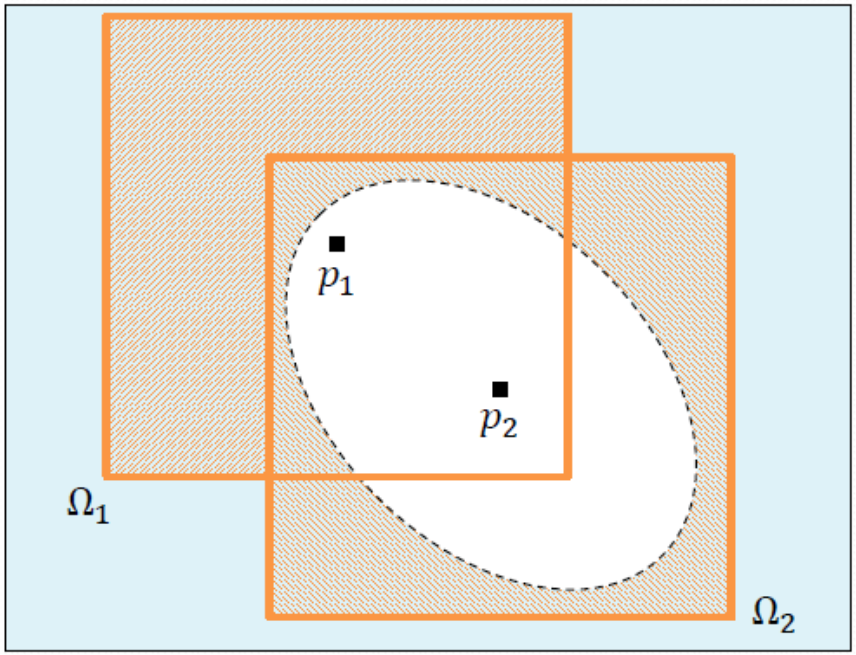
GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Importance of Spatial Support



p_1 can be computed
p_2 cannot be computed



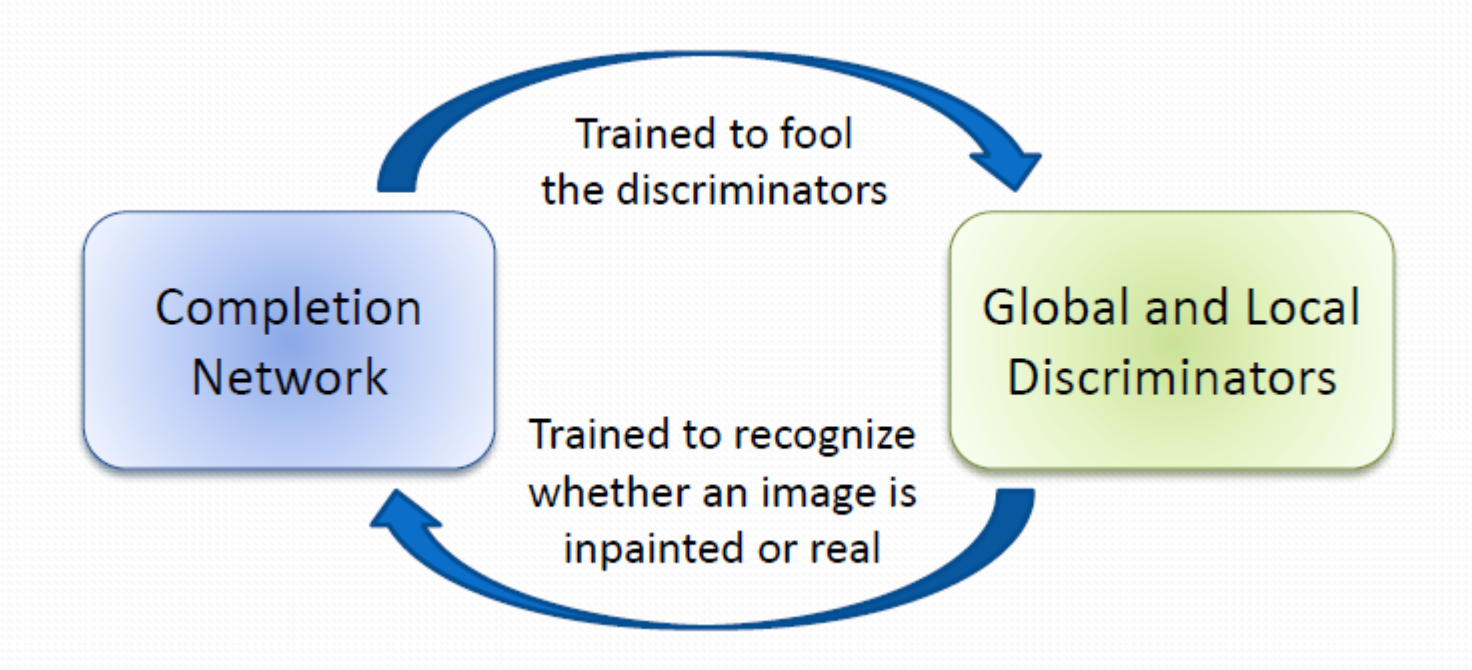
Both can be computed

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Training

- Alternately update the completion network and discriminators
  - Based on Generative Adversarial Networks
  - MSE(Mean Squared Error)



GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Loss function

- MSE loss：

$$L(x, M_c) = \| M_c \odot (C(x, M_c) - x) \|^2$$

- GAN optimization：

$$\min_C \max_D \mathbb{E}[\log D(x, M_d) + \log(1 - D(C(x, M_c), M_c)]$$

- Combined optimization：

$$\min_C \max_D \mathbb{E}[L(x, M_c) + \alpha \log D(x, M_d) + \alpha \log(1 - D(C(x, M_c), M_c))]$$

Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. In SIGGRAPH ,2017

Algorithm 1 Training procedure of the image completion network.

while iterations $t < T_{train}$ **do**
  Sample a minibatch of images x from training data.
  Generate masks M_c with random holes for each image x in the minibatch.
  if $t < T_C$ **then**
    Update the completion network C with the weighted MSE loss (Eq. (2)) using (x, M_c).
  else
    Generate masks M_d with random holes for each image x in the minibatch.
    Update the discriminators D with the binary cross entropy loss with both $(C(x, M_c), M_c)$ and (x, M_d).
    if $t > T_C + T_D$ **then**
      Update the completion network C with the joint loss gradients (Eq. (5)) using (x, M_c), and D.
    end if
  end if
end while

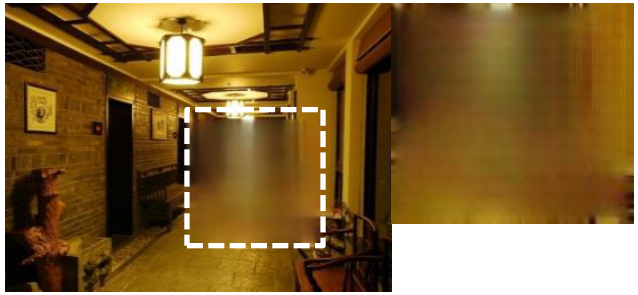
Comparison of training with different discriminator configurations

(a) Input (b) Weighted MSE (c) Global D (d) Local D (e) Full method

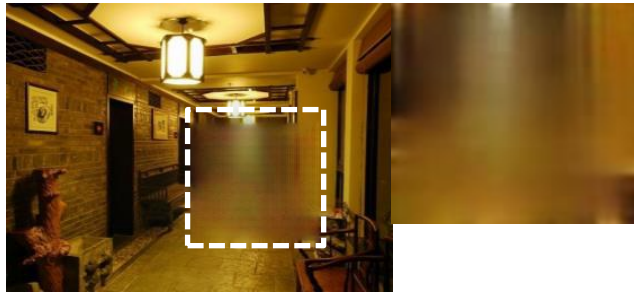
Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. In SIGGRAPH ,2017

Training with Different Discriminator Configurations

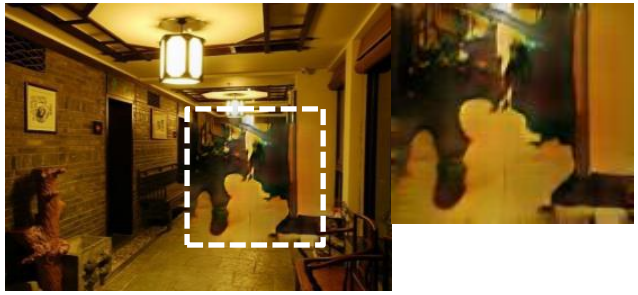
Input



Mean Squared Error(MSE)



MSE + Global discriminator



MSE + Local discriminator

Full method

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

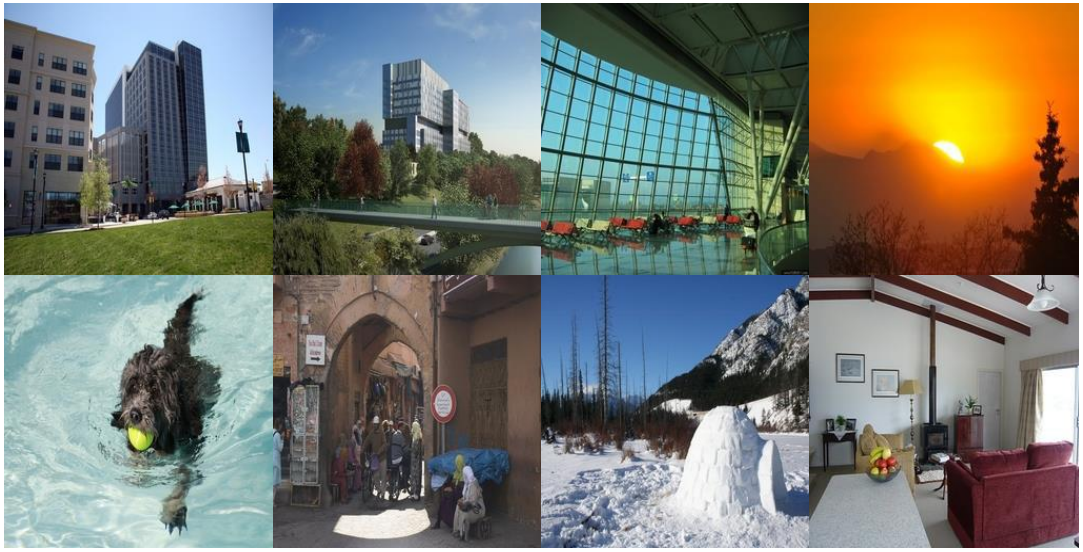
Post-Processing

(a) Input (b) Output (c) Post-processed

Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. In SIGGRAPH ,2017

Dataset

- Places2 dataset
  - About 8 million images with various scenes
  - Randomly generate a hole for training



Places2 dataset

Input Ground truth

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Result: Image Completion

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Result: Object Removal

GitHub https://github.com/satoshiiizuka/siggraph2017_inpainting

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Comparison

Comparison with Photoshop CS6 Content Aware Fill (PatchMatch [Barnes et al. 2009]), [Darabi et al. 2012], [Huang et al. 2014], and [Pathak et al. 2016] on a diverse set of scenes using random masks.

- Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing.. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 28, 3 (2009), 24:1–24:11.

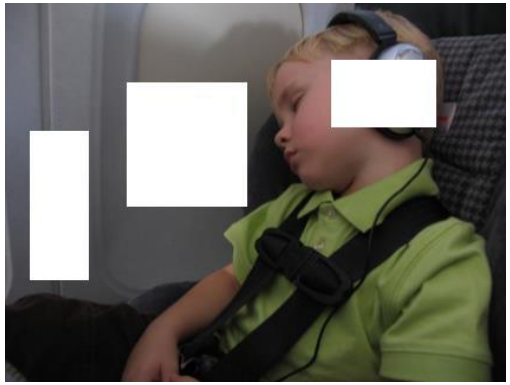
- Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image Melding: Combining Inconsistent Images using Patch-based Synthesis.. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 31, 4 (2012), 82:1–82:10.

- Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image Completion Using Planar Structure Guidance.. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 33, 4 (2014), 129:1-129:10.

- Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context Encoders: Feature Learning by Inpainting. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.

http://hi.cs.waseda.ac.jp/~iizuka/projects/completion/en/

Comparison



Input

Patch Match
[Barnes et al. 2009]

Image Melding
[Darabi et al. 2012]

Content Encoder
[Pathak et al. 2016]

Ours

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Comparison

Input

Patch Match
[Barnes et al. 2009]

Image Melding
[Darabi et al. 2012]

Content Encoder
[Pathak et al. 2016]

Ours

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Application to Specific Dataset

- Fine-tuning the model using a specific dataset
  - Achieves more complicated inpainting

- Face dataset
  - 200,000 training images
  - 2499 test images

Large-scale Celeb Faces Attributes Dataset(CelebA)

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Result: Face Completion

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Result: Removing Sunglasses

Original Input Output

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Failure Case

Input Ours Ground truth

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

Conclusion

- Novel network for globally and locally consistent image completion
  - Completion network for arbitrary region inpainting
  - Adversarial training with global and local discriminators

https://www.slideshare.net/siizuka/siggraph-2017-globally-and-locally-consistent-image-completion

- Globally and Locally Completion
- **Semantic Image Inpainting**
- High-Resolution Image Inpainting
- Content Encoders

Semantic Image Inpainting with Deep Generative Models

Raymond A. Yeh*, Chen Chen*, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Do

In CVPR 2017

[Image from KDEF dataset]

GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Motivation

Task: Semantic image inpainting (filling large missing regions)

- ill-posed task
- requires strong prior knowledge on the data
- extracting information from only a single image produces unsatisfactory results

Contributions:

- deep generative models produce missing content by conditioning on available data
- inpainting as constrained optimization problem using context and prior loss

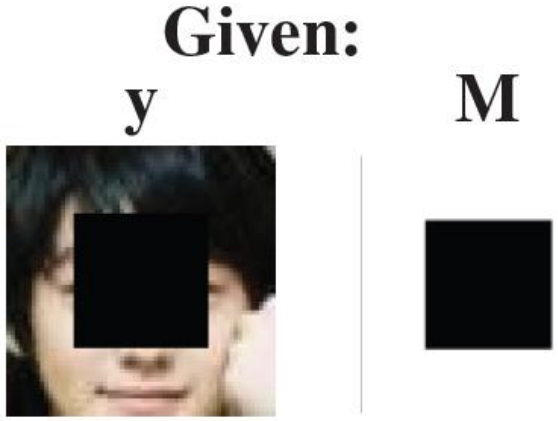
GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Introduction

Problem Formulation:

- Corrupted image: **y**
- Binary mask: **M**
- Task: predict uncorrupted version $\hat{x}$



Baselines:

- Total Variation and Low Rank assume smoothness in the pixel space
- Context Encoder is a deep model which treats inpainting as a regression problem

GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

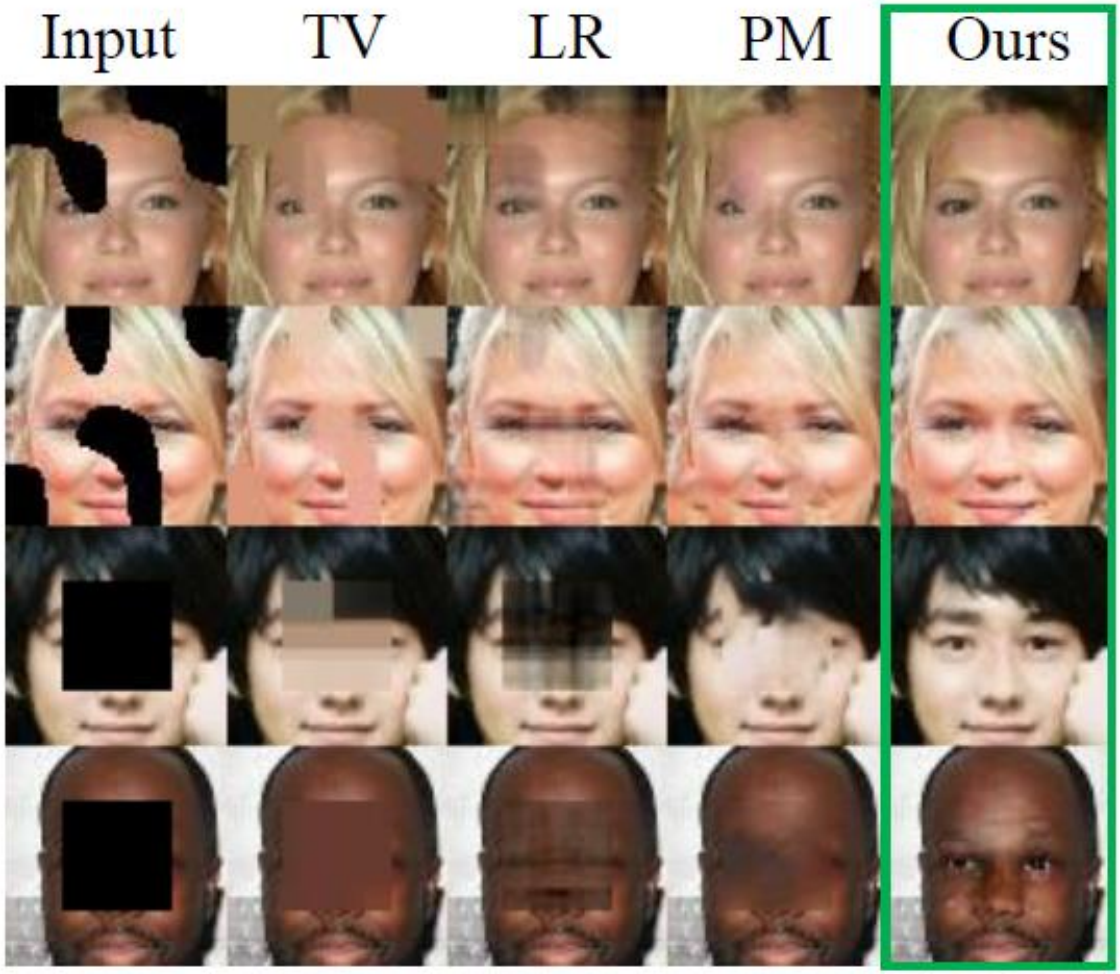


Figure 1. Semantic inpainting results by TV, LR, PM and our method. Holes are marked by black color.

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Introduction

- DCGAN-Based
- Not end-to-end
  - Train generator first (uncorrupted data)
  - Find $\hat{z}$ for inpainting

GitHub https://github.com/moodoki/semantic_image_inpainting

https://www.slideshare.net/ssuser73ec8f/semantic-image-inpainting

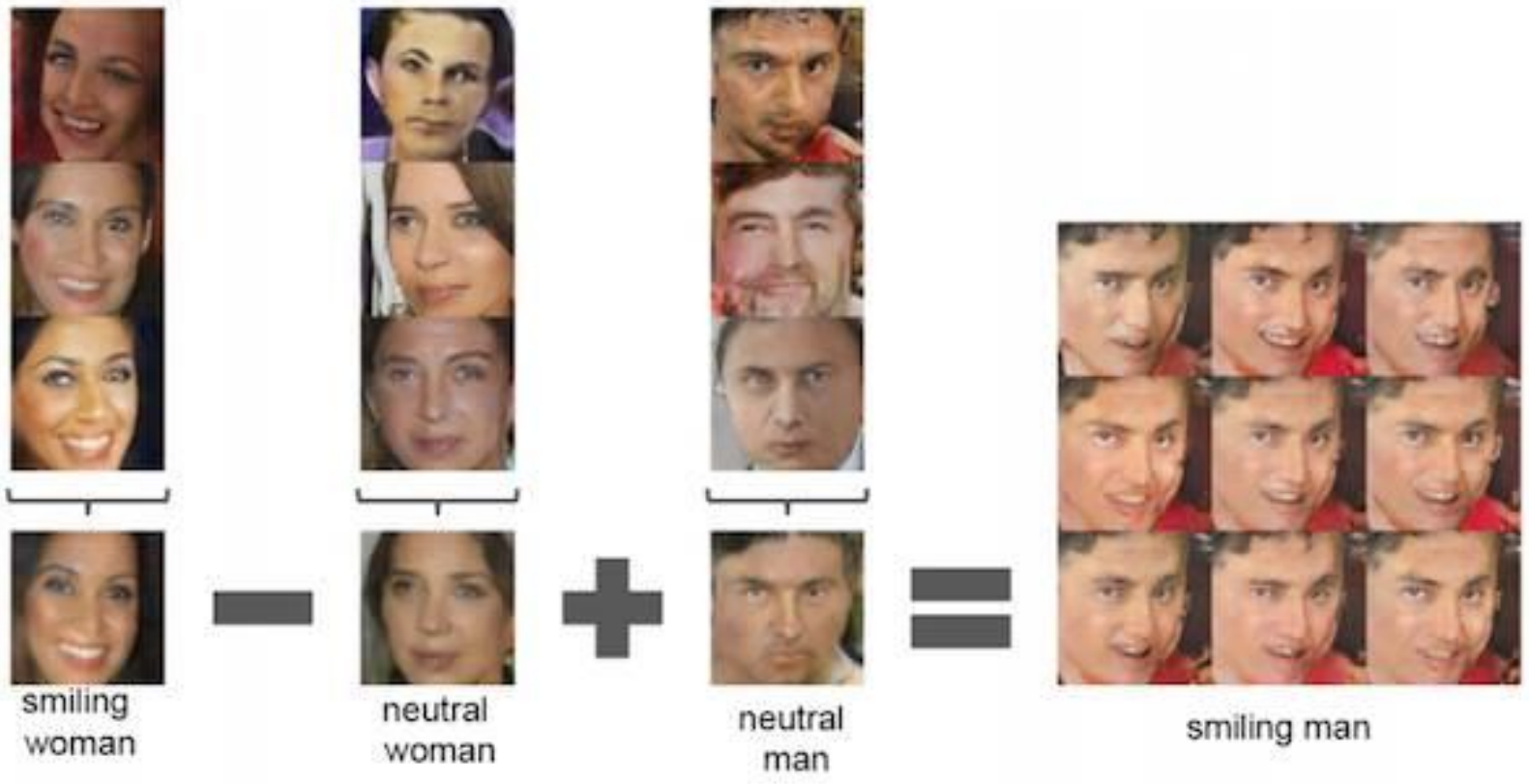
Introduction

- DCGAN-Based

https://mp.weixin.qq.com/s/JSJ2k7IeorOfI3tvpMJK7A

Introduction

- DCGAN-Based



https://mp.weixin.qq.com/s/JSJ2k7IeorOfI3tvpMJK7A

Image Inpainting

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Introduction

Instead of explicitly defining the prior, we utilize deep generative models to capture prior information

Generative Adversarial Networks:

- Generator G: deep net mapping perturbation z to artificial sample
- Discriminator D: deep net discriminating between artificial and real sample, x
- Program:

$$\min_G \max_D V(G,D) = E_{x\sim P_{data}}[\log(D(x))] + E_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Approach

Intuition of our approach:

- **Hypothesis**: Trained G is efficient-image not from p_{data} (e.g., corrupted data) **should not lie** on the learned encoding manifold; use manifold can be used as a prior

- **Objective**: Find encoding $\hat{z}$: "closest" to the corrupted image while constrained to the manifold

GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Approach

Solving for the "closest" encoding $\hat{z}$:

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) + \mathcal{L}_p(\mathbf{z})$$

Context Loss: importance weighted metric W to enforce similarity to the uncorrupted regions:

$$\mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) = \|\mathbf{W} \odot (G(\mathbf{z}) - \mathbf{y})\|_1$$

Prior Loss: prior penalizing unrealistic images based on the discriminator:

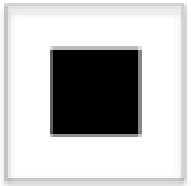
$$\mathcal{L}_p(\mathbf{z}) = \lambda \log(1 - D(G(\mathbf{z})))$$

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Approach

CONTEXTUAL LOSS PRIOR LOSS

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) + \mathcal{L}_p(\mathbf{z})$$

$$Loss = L_p(\boldsymbol{z}) + L_c(\boldsymbol{z}|$$  , 

$\boldsymbol{y}$ $\mathbf{M}$

y : corrupted image

M: binary mask(size equal to the image)

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Approach

Importance weight metric: :

$$\mathbf{W}_i = \begin{cases} \sum_{j \in N(i)} \frac{(1-\mathbf{M}_j)}{|N(i)|} & \text{if} \quad \mathbf{M}_i \neq 0 \\ 0 & \text{if} \quad \mathbf{M}_i = 0 \end{cases}$$

Wi: importance weight at pixel location i

N(i) : set of neighbors of pixel i in a local window

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Approach

Context Loss:

Importance weighted metric W to enforce similarity to the uncorrupted regions:

$$\mathcal{L}_c(\mathbf{z}|\mathbf{y},\mathbf{M}) = \|\mathbf{W} \odot (G(\mathbf{z}) - \mathbf{y})\|_1$$

y : corrupted image

M: binary mask(size equal to the image)

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Prior Loss:

- How realistic the generated image is
- Prior penalizing unrealistic images based on the discriminator

$$\mathcal{L}_p(\mathbf{z}) = \lambda \log(1 - D(G(\mathbf{z})))$$

Without prior loss, the mapping from y to z may converge to a perceptually implausible result

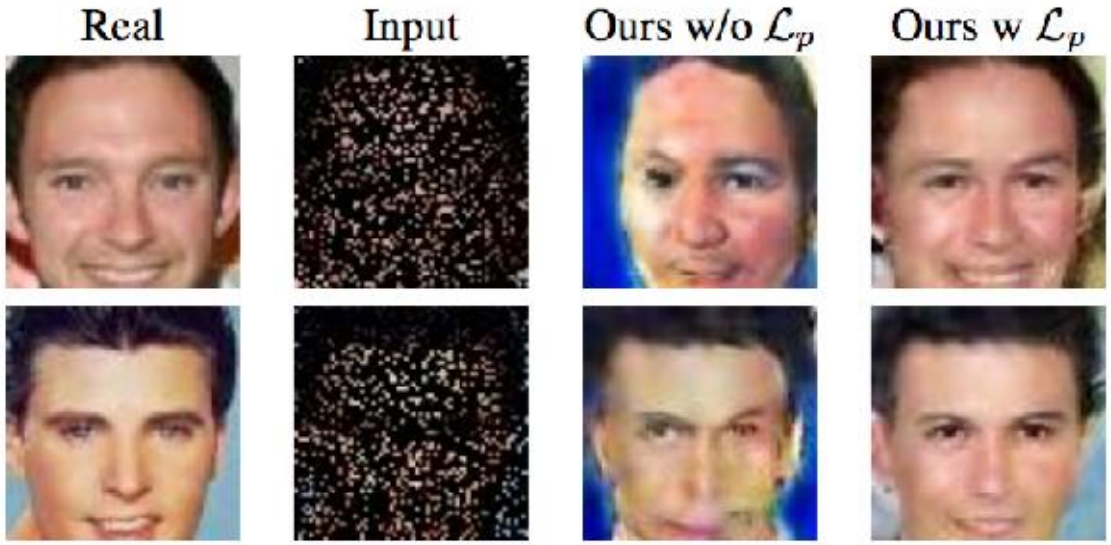
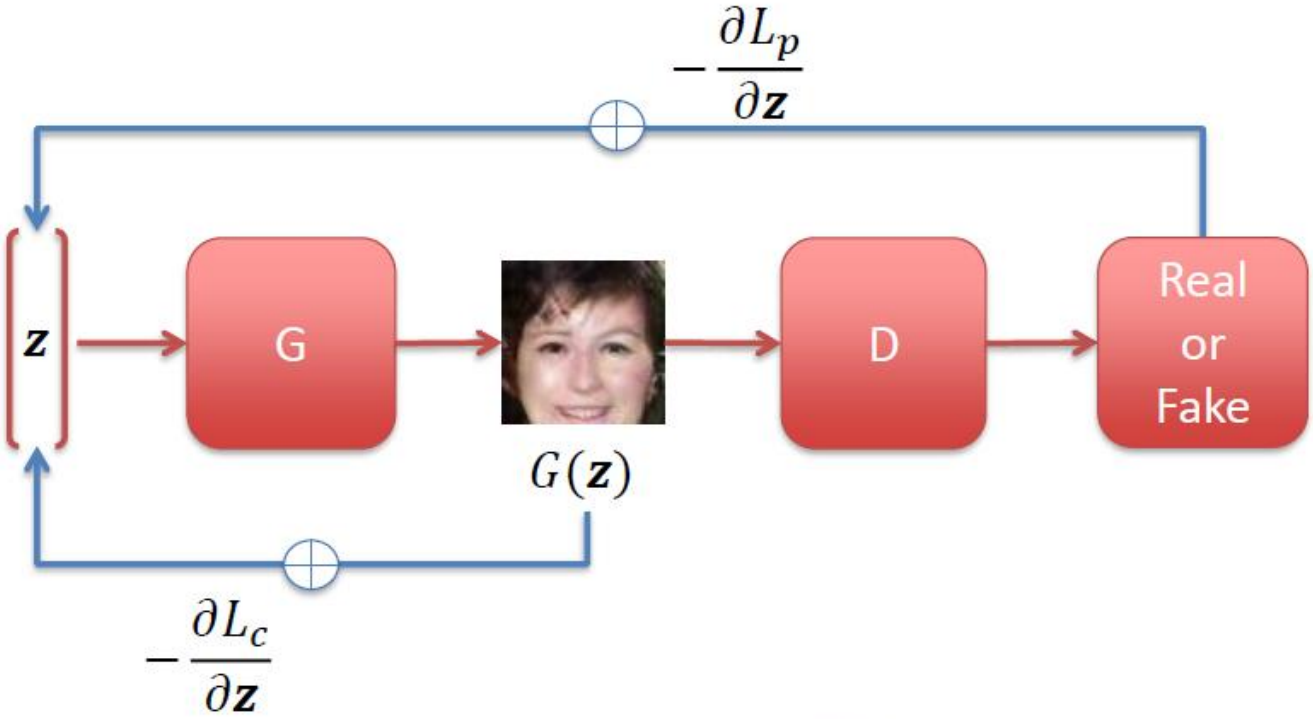


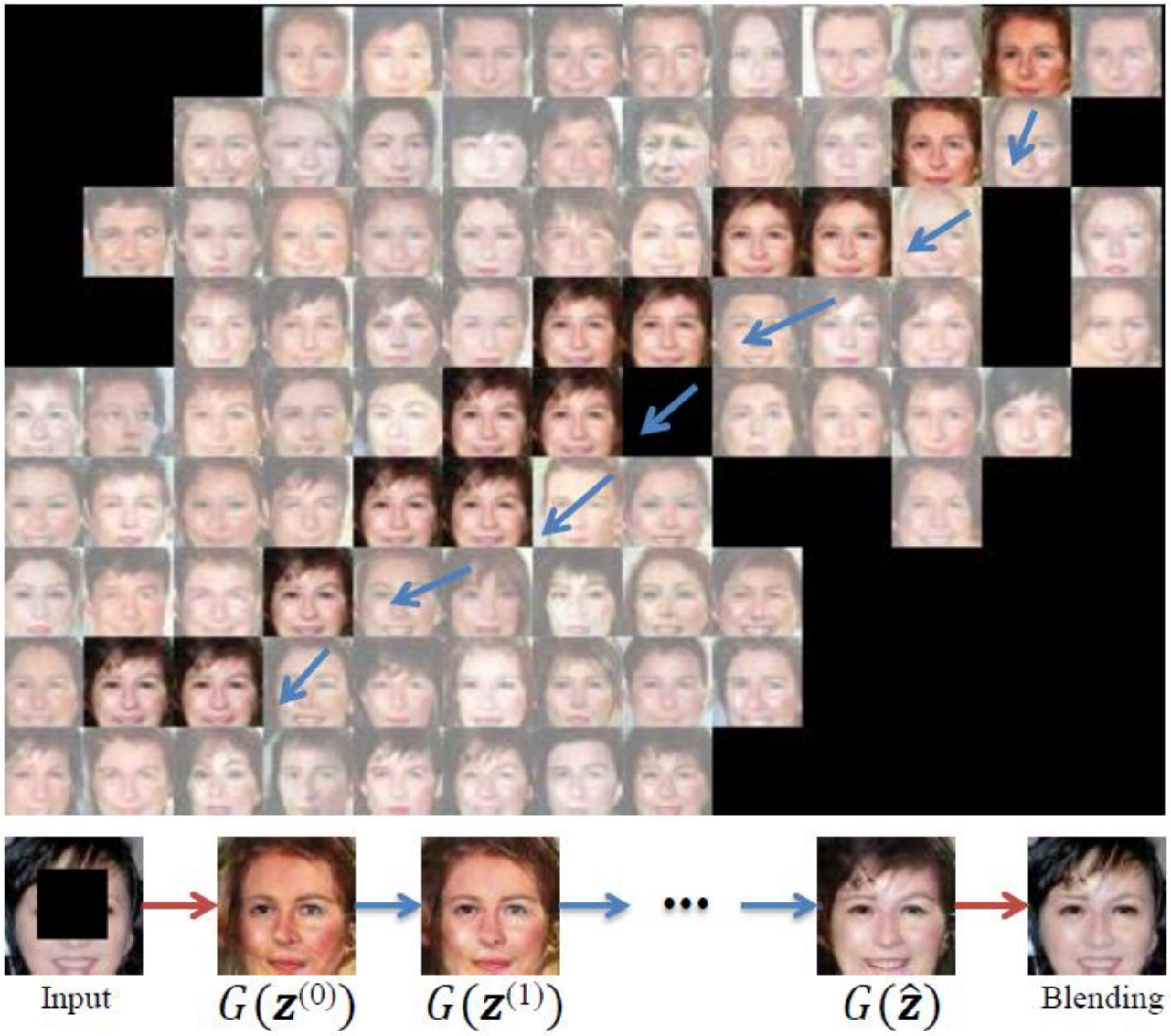
Figure 4. Inpainting with and without the prior loss.

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Illustration of the approach:



Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.



Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Approach

Poisson Blending:

Recovering prediction $\hat{x}$ via poisson blending rather than simple overlay:

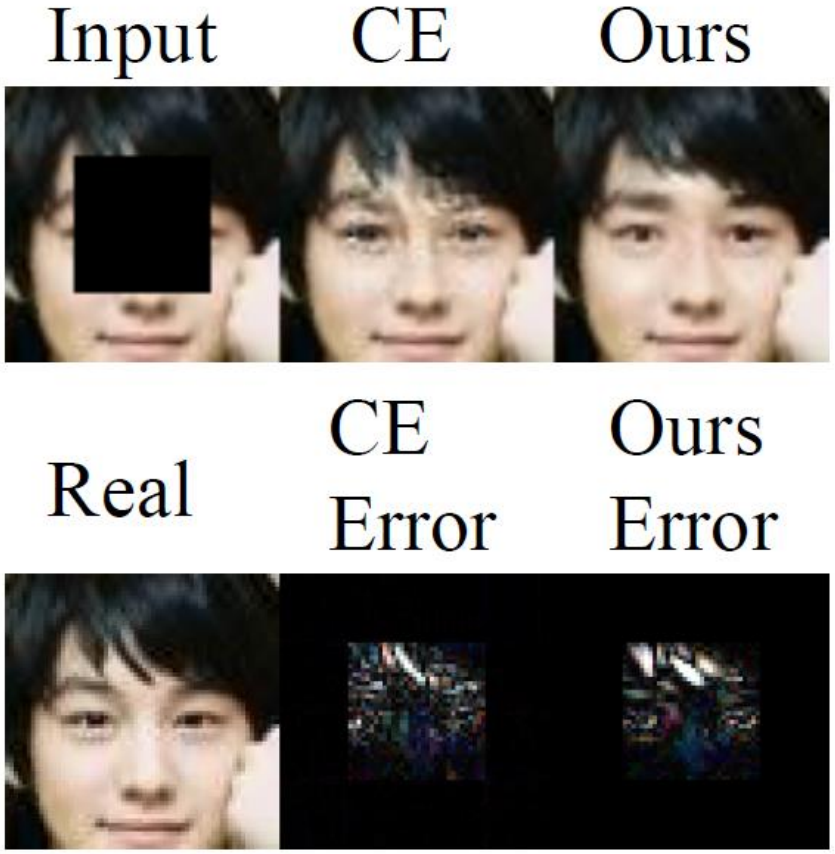
$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\nabla \mathbf{x} - \nabla G(\hat{\mathbf{z}})\|_2^2 \qquad \text{s.t. } \mathbf{x}_i = \mathbf{y}_i \text{ for } \mathbf{M}_i = 1$$

Comparison: Poisson Blending vs. Overlay:

Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Results

Quantitative Results:



Raymond A. Yeh, Chen Chen, Teck Yian Lim,Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Semantic Image Inpainting with Deep Generative Models .In CVPR, 2017.

Quantitative Results:

The PSNR values (dB) on the test sets. Left/right results are by Context Encoder (CE)/ours:

Masks/Dataset	CelebA	SVHN	Cars
Center	**21.3**/19.4	**22.3**/19.0	**14.1**/13.5
pattern	**19.2**/17.4	**22.3**/19.8	14.0/**14.1**
random	20.6/**22.8**	24.1/**33.0**	16.1/**18.9**
half	**15.5**/13.7	**19.1**/14.6	**12.6**/11.1

- In the figure above, PSNR for CE is 24.71 dB and ours is 22.98 dB
- Higher PSNR does not mean better visual quality
- The solution is not unique, many hallucinations are reasonable

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Datasets:

Images: 64*64

- CelebA dataset

  202,599 face images (2,000 for test)

- Car dataset

  16,185 car images (8,041 for test)

- SVHN dataset

  99,289 number images (26,032 for test)

GitHub https://github.com/moodoki/semantic_image_inpainting

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Results - CelebA:

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Results – The Stanford Car:

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

Results – SVHN:

http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

For paper, results, code and more:

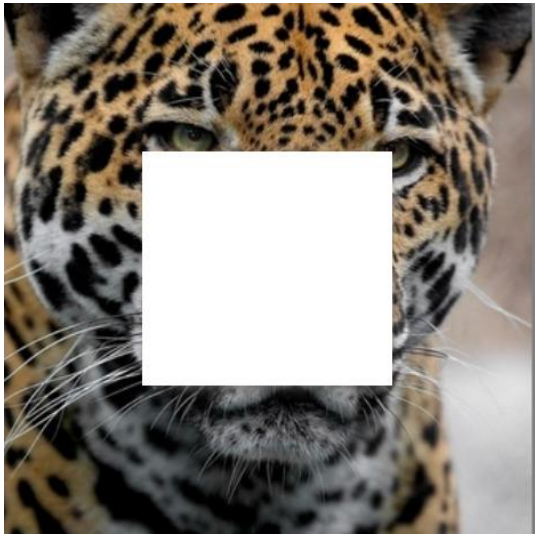
http://www.isle.illinois.edu/~yeh17/projects/semantic_inpaint/poster.pdf

- Globally and Locally Completion
- Semantic Image Inpainting
- ➢ **High-Resolution Image Inpainting**
- Content Encoders

High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis

Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li

In CVPR 2017



GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Bad things happen

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Introduction

Imaging you would like to edit a photo after breaking up, or restore an old picture from damages, we designed a MULTI-SCALE DEEP LEARNING algorithm to help you! Our approach

- Proposed a joint optimization framework that can hallucinates missing image regions by modeling a global content constraint and local texture constraint with convolutional neural networks.
- Further introduced a multi-scale neural patch synthesis algorithm for high-resolution image inpainting based on the joint optimization framework.

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Algorithm

High-Resolution Hole Filling with Multi-Scale Neural Patch Synthesis

Input: Image x,
the content network f,
the texture network t,
the number of scales N

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

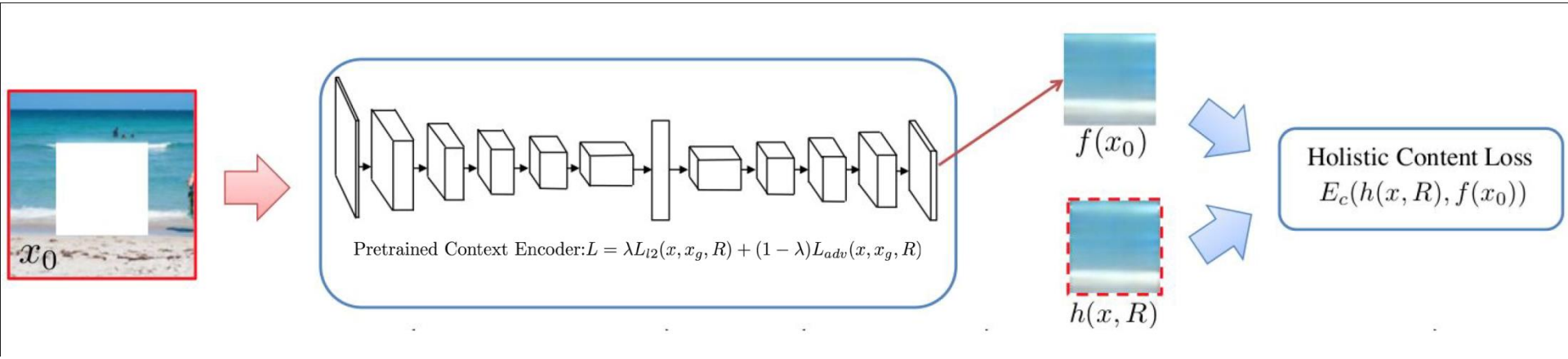
Algorithm

High-Resolution Hole Filling with Multi-Scale Neural Patch Synthesis

1: Downsize x to 128×128.
2: Compute the initial content reference x^1. by giving x as input to f.
3: **for** $s \in [1, 2, \ldots, N]$:
4. Initialize $\tilde{x} = x^s$.
4: Update $\tilde{x}$ that minimizes the joint loss: $L = L_{content} + L_{texture} + L_{tv-smoothness}$.
5: Compute x^{s+1} by up-sampling $\tilde{x}$.
7: **end for**
8: Return $\tilde{x}^N$.

https://github.com/leehomyc/CVPR_2017_Poster/

The Content Network



Context Encoder Predicts the Low-Res Content

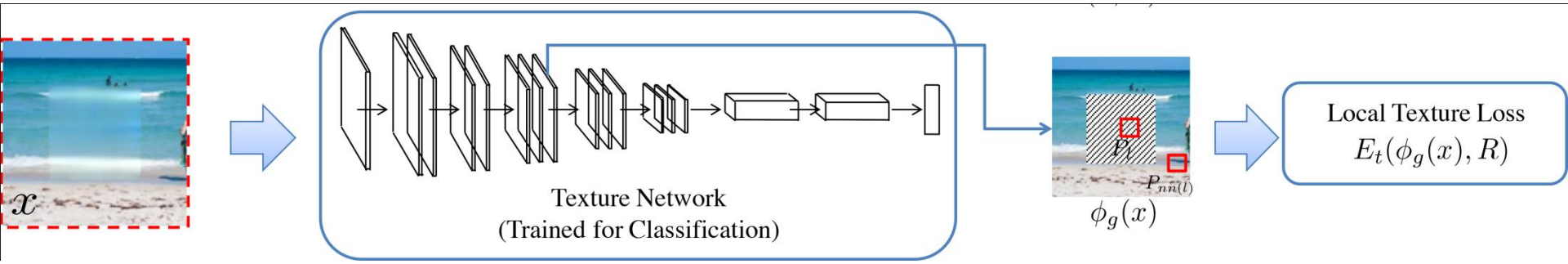
The content constraint:

$$E_c(h(x,R), h(x_i,R)) = \| h(x,R) - h(x_i,R) \|_2^2$$

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

The Texture Network



Pre-trained VGG Optimizes the High-Res Texture

The texture constraint:

$$E_t(\phi_t(x), R) = \frac{1}{|R^{\phi}|} \sum_{i \in R^{\phi}} \| h(\phi_t(x), P_i) - h(\phi_t(x), P_{nn(i)}) \|_2^2$$

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

The Joint Loss Function

At each iteration, we minimize:

$$\tilde{x}_{i+1} = \arg\min_x E_c(h(x, R), h(x_i, R)) + \alpha E_t(\phi_t(x), R^{\phi}) + \beta\gamma(x)$$

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Multi-Scale Optimization

We optimize at three scales: **128**, **256** and **512**:

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Changing the texture weight α

The weight α measures the contribution of the texture constraint relative to the content constraint.
It is a trade off between the sharpness of the texture and coherence of the structure:

(a) Input image (b) $\alpha = 1e-6$ (c) $\alpha = 1e-5$ (d) $\alpha = 4e-5$

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/

Dropping the content constraint

Without using the content term to guide the optimization, the structure of the inpainting results is completely incorrect, although they are visually sharp:

Input

Output without CN

Output with CN

https://github.com/leehomyc/CVPR_2017_Poster/

Dropping the adversarial loss

When the initial prediction is blurry (using L2 loss only), the final result becomes blurrier as well comparing with using the content network trained with both L2 and adversarial loss.

(a) CN with ℓ_2 loss (b) (a)'s final output (c) CN with ℓ_2 & Adv loss (d) (c)'s final output.

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

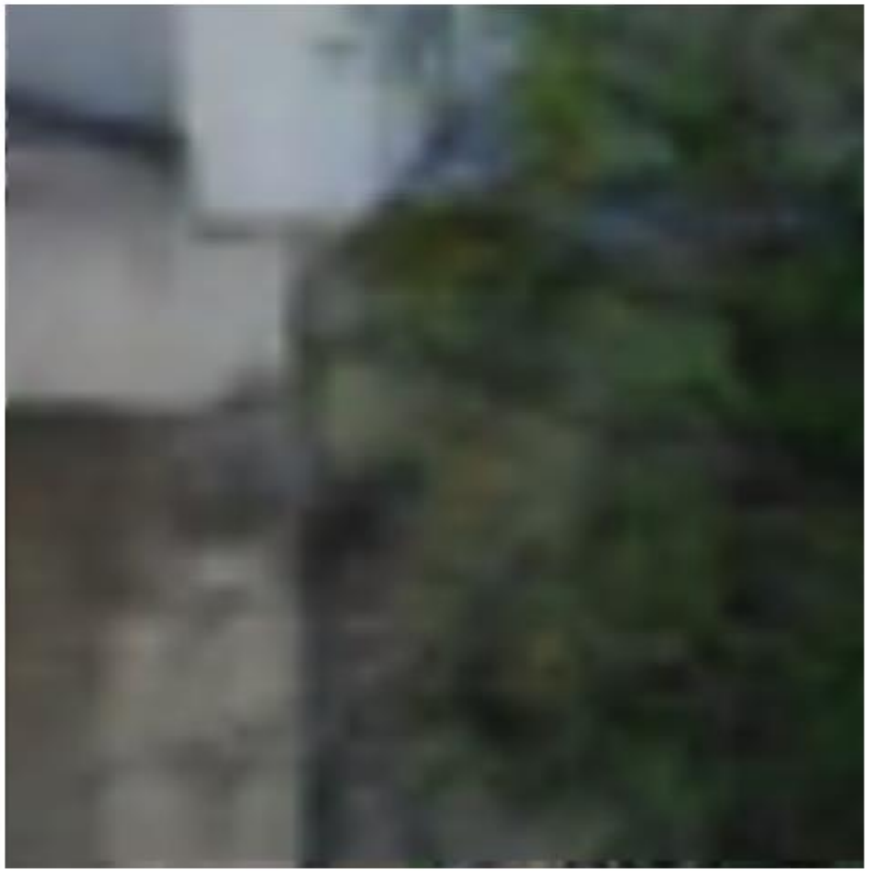
https://github.com/leehomyc/CVPR_2017_Poster/

Overall comparison with other methods

(a) Input hole (b) Context Encoder (l_2 loss) (c) Content-Aware Fill (d) Context Encoder (l_2+ Adversarial loss) (e) Our Method

GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

https://github.com/leehomyc/CVPR_2017_Poster/



(d) Zoom-in

(e) Zoom-in

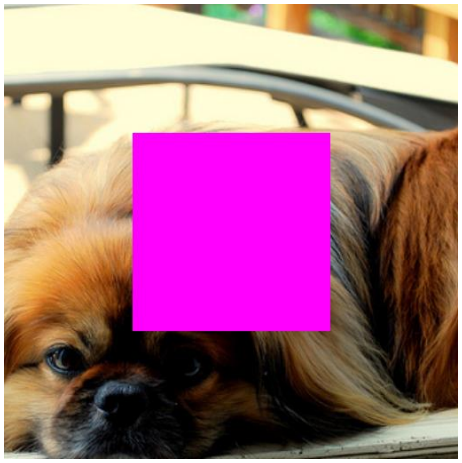
GitHub https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting

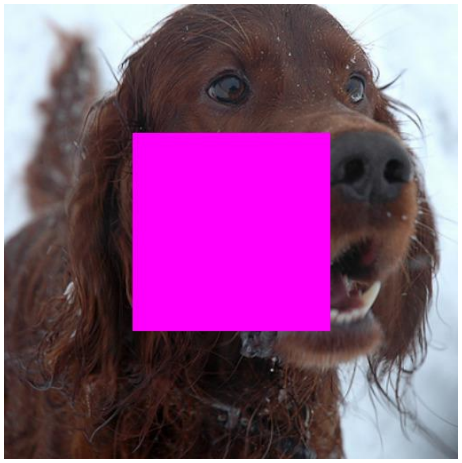
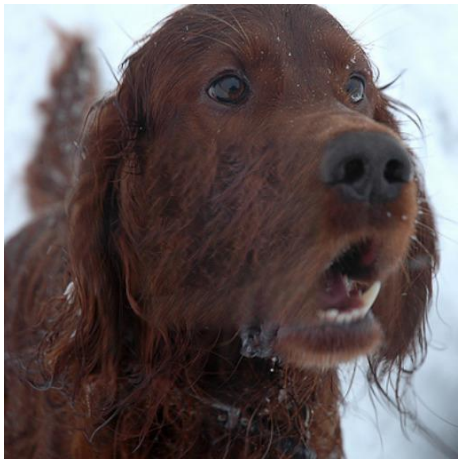
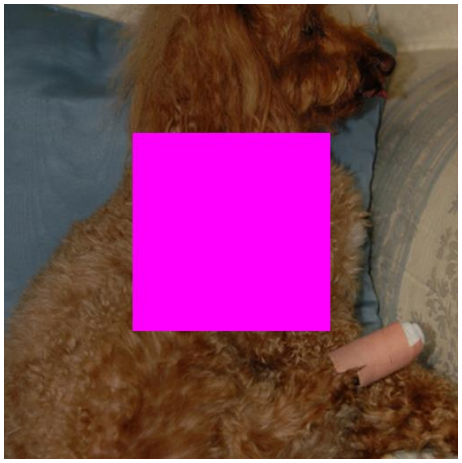
https://github.com/leehomyc/CVPR_2017_Poster/

More Results

And we have collected so many dogs and cats for you:

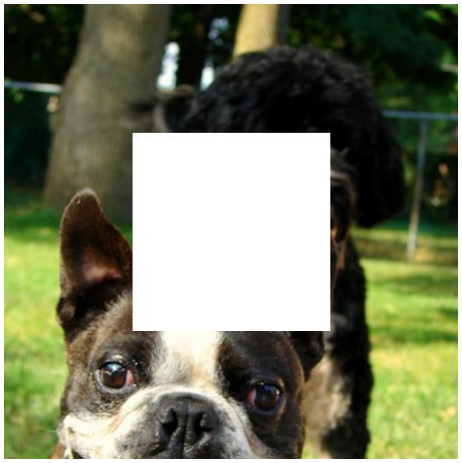
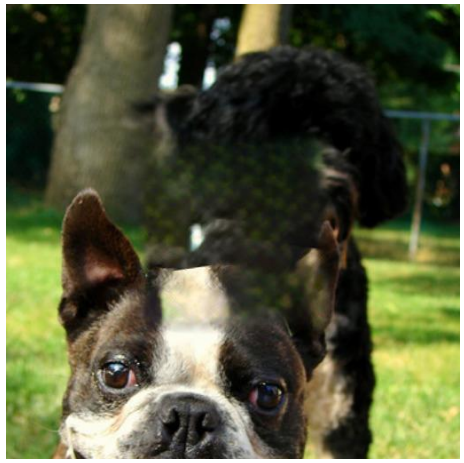




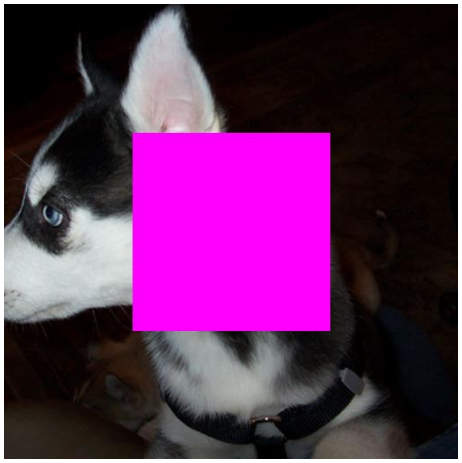
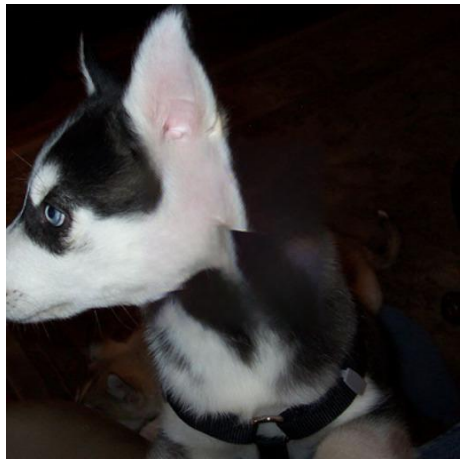





https://github.com/leehomyc/CVPR_2017_Poster/

And we have collected so many dogs and cats for you:



https://github.com/leehomyc/CVPR_2017_Poster/

And what inpaints Paris like Paris ?

https://github.com/leehomyc/CVPR_2017_Poster/

And what inpaints Paris like Paris ?

https://github.com/leehomyc/CVPR_2017_Poster/

For paper, results, code and more:

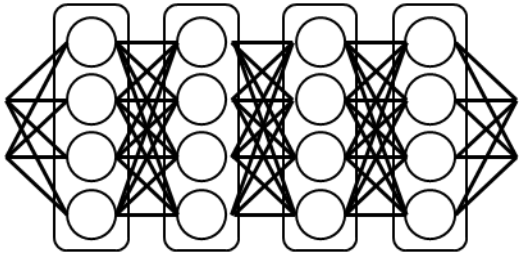
https://github.com/leehomyc/CVPR_2017_Poster/

- Globally and Locally Completion
- Semantic Image Inpainting
- High-Resolution Image Inpainting
- **Context Encoders**

Context Encoders: Feature Learning by Inpainting

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei Efros

In CVPR 2016



GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Context Encoders:

Feature Learning by Inpainting

GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

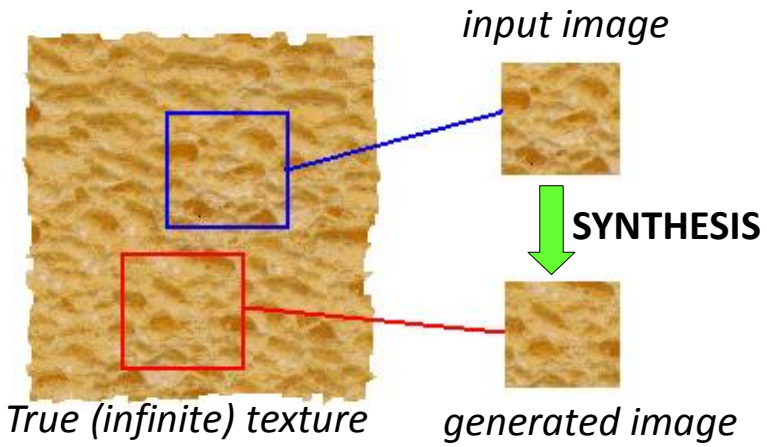
Image Inpainting

- Inpaint large missing regions in image: not possible for local texture synthesis methods

  [Efros & Leung 1999]

- Nearest neighbor based scene completion

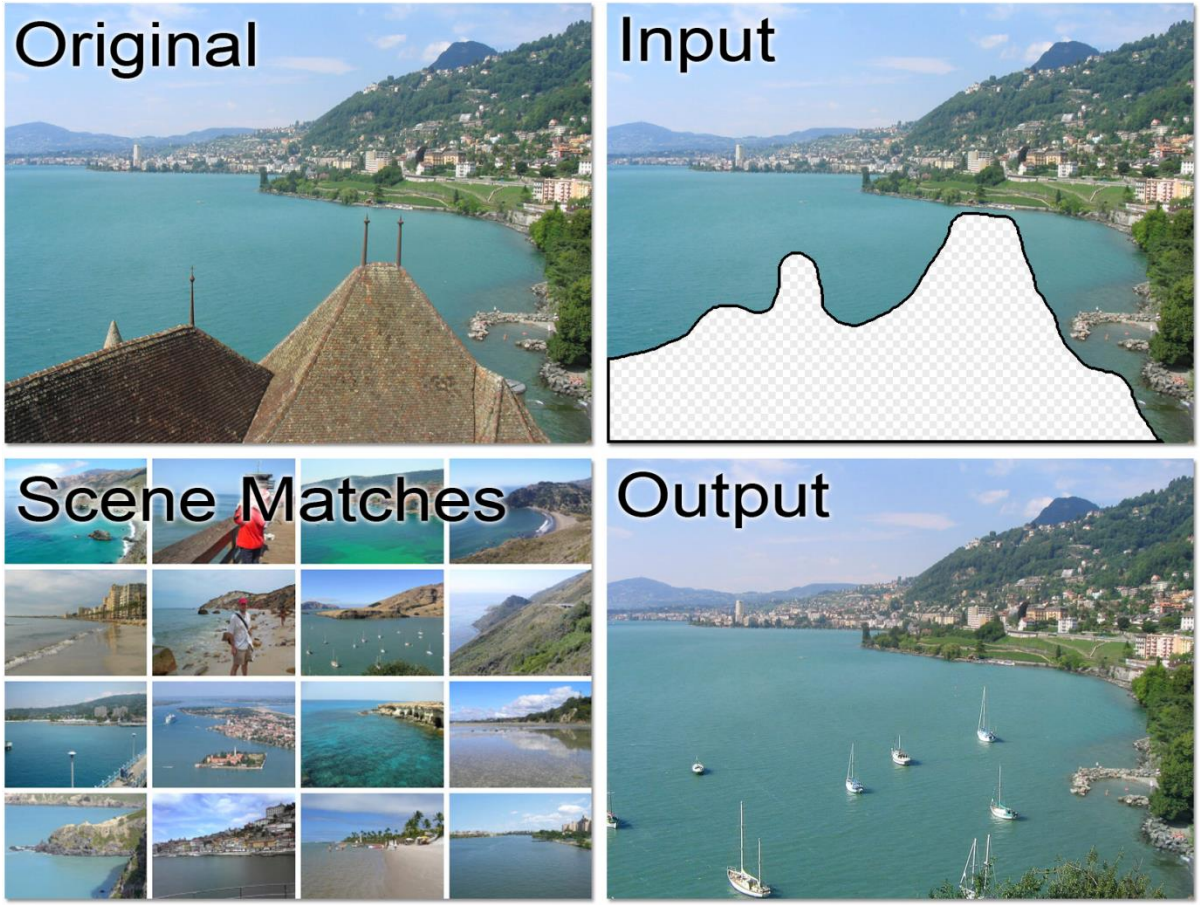
  [Hays & Efros 2007]



GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

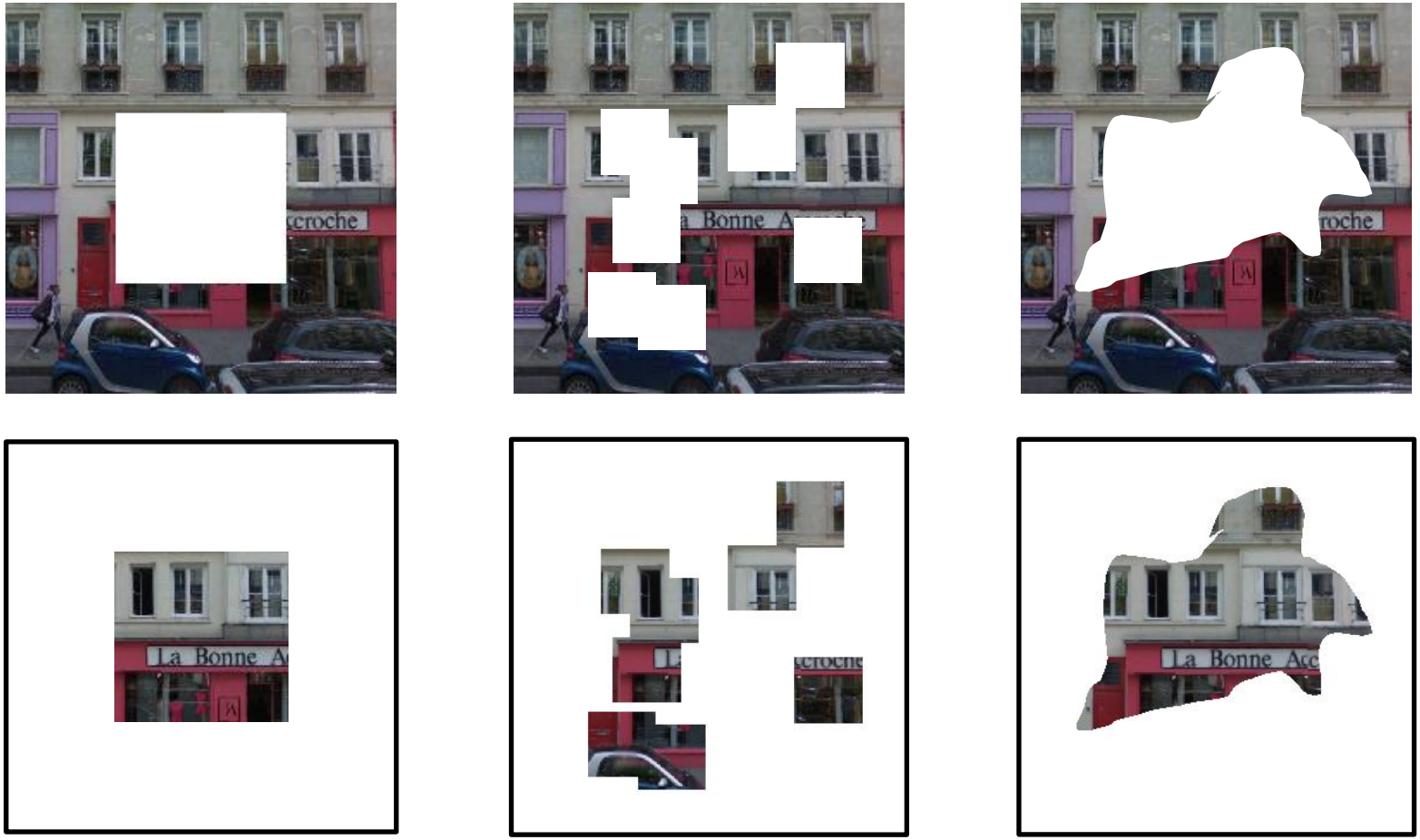
Image Inpainting



GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

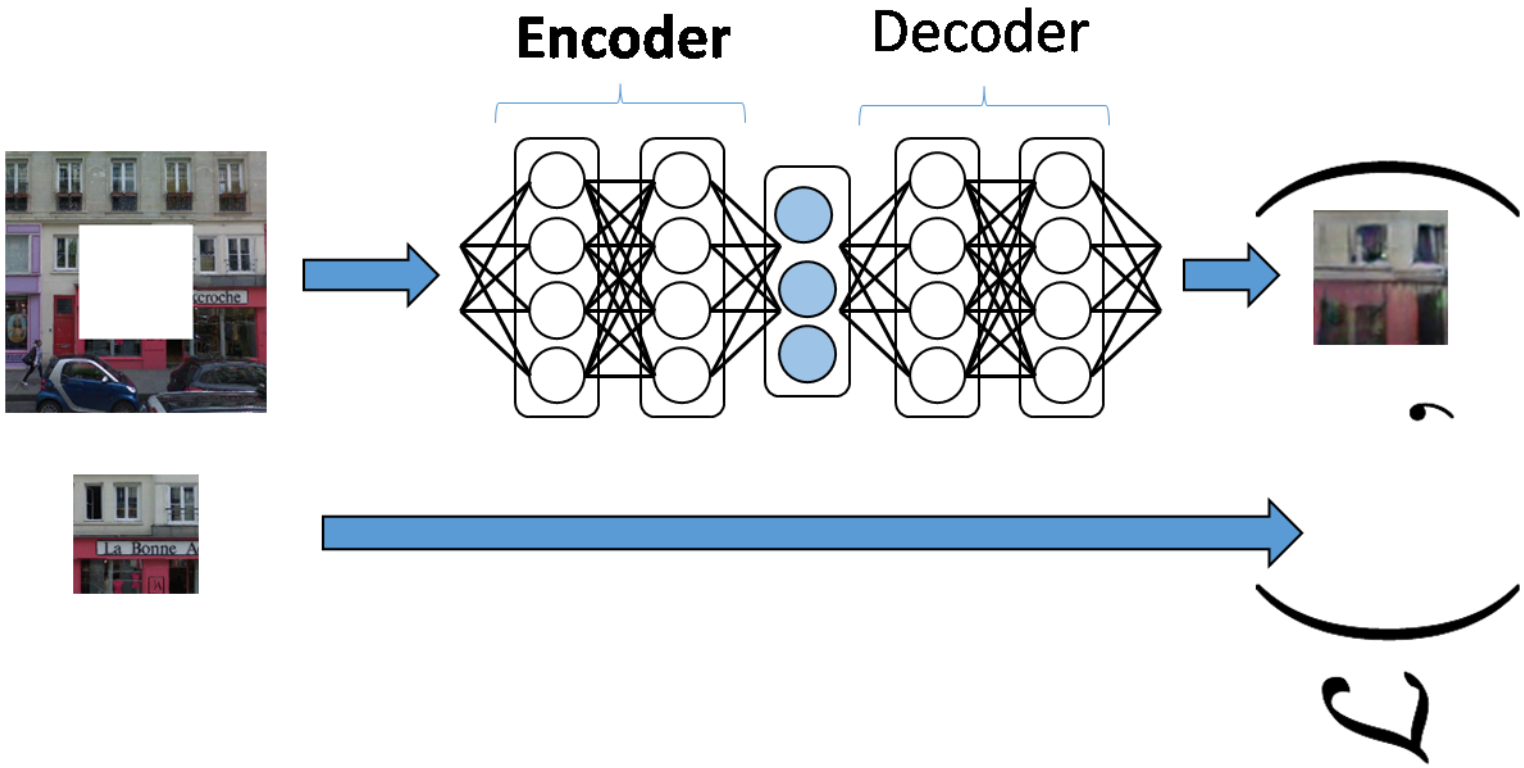
Schemes for Region Dropout



(a) Center Region (b) Random Blocks (c) Random Shapes

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Context Encoders

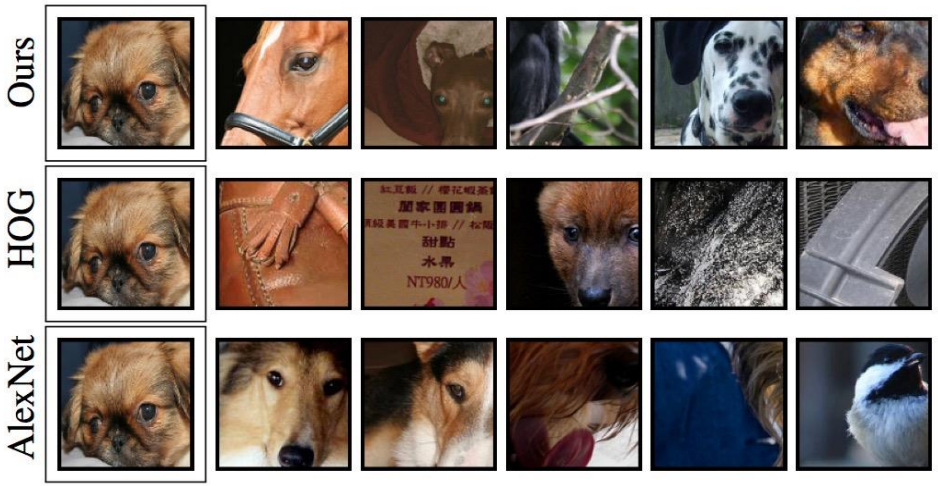


- Encoder can be substituted with any network architecture like AlexNet etc.
- Decoder is a set of UpConv/deconv/frac-strided-conv layers

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Nearest Neighbors with L2 Loss

- Nearest Neighbors are good in the representation space
- But generated output is blurry => *Artifact of L2 !!*



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Need Mode Picking => GAN

Generative Adversarial Networks

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

(Submitted on 10 Jun 2014)

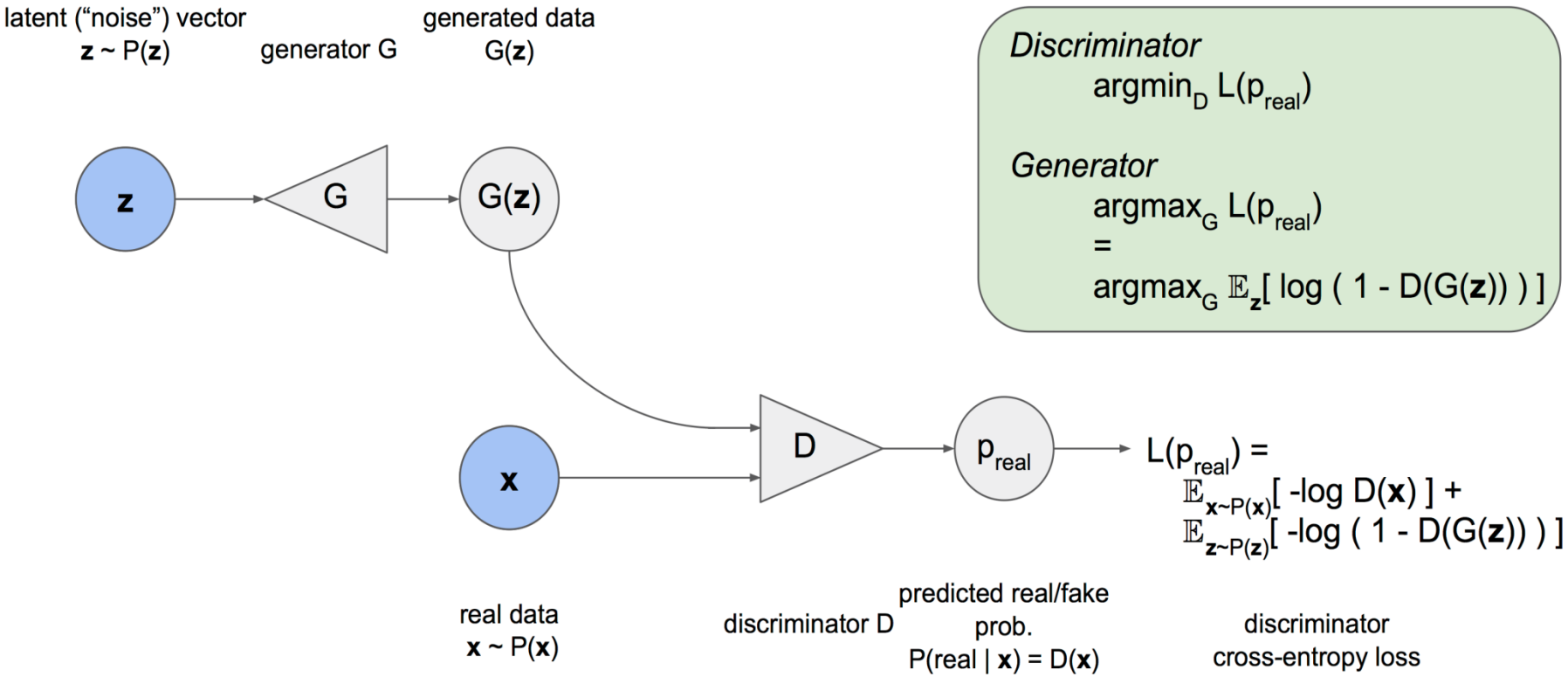
We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Subjects: **Machine Learning (stat.ML)**; Learning (cs.LG)
Cite as: **arXiv:1406.2661 [stat.ML]**
(or **arXiv:1406.2661v1 [stat.ML]** for this version)

Submission history
From: Ian Goodfellow [view email]
[v1] Tue, 10 Jun 2014 18:58:17 GMT (1257kb,D)

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Generative Adversarial Networks



[Slide adapted from Jeff Donahue]

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Loss:

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

$$\mathcal{L}_{adv}(x) = \max_D \ \mathbb{E}_{x \in \mathcal{X}} \left[\log(D(x)) + \log(1 - D(F((1 - \hat{M}) \odot x)))\right]$$

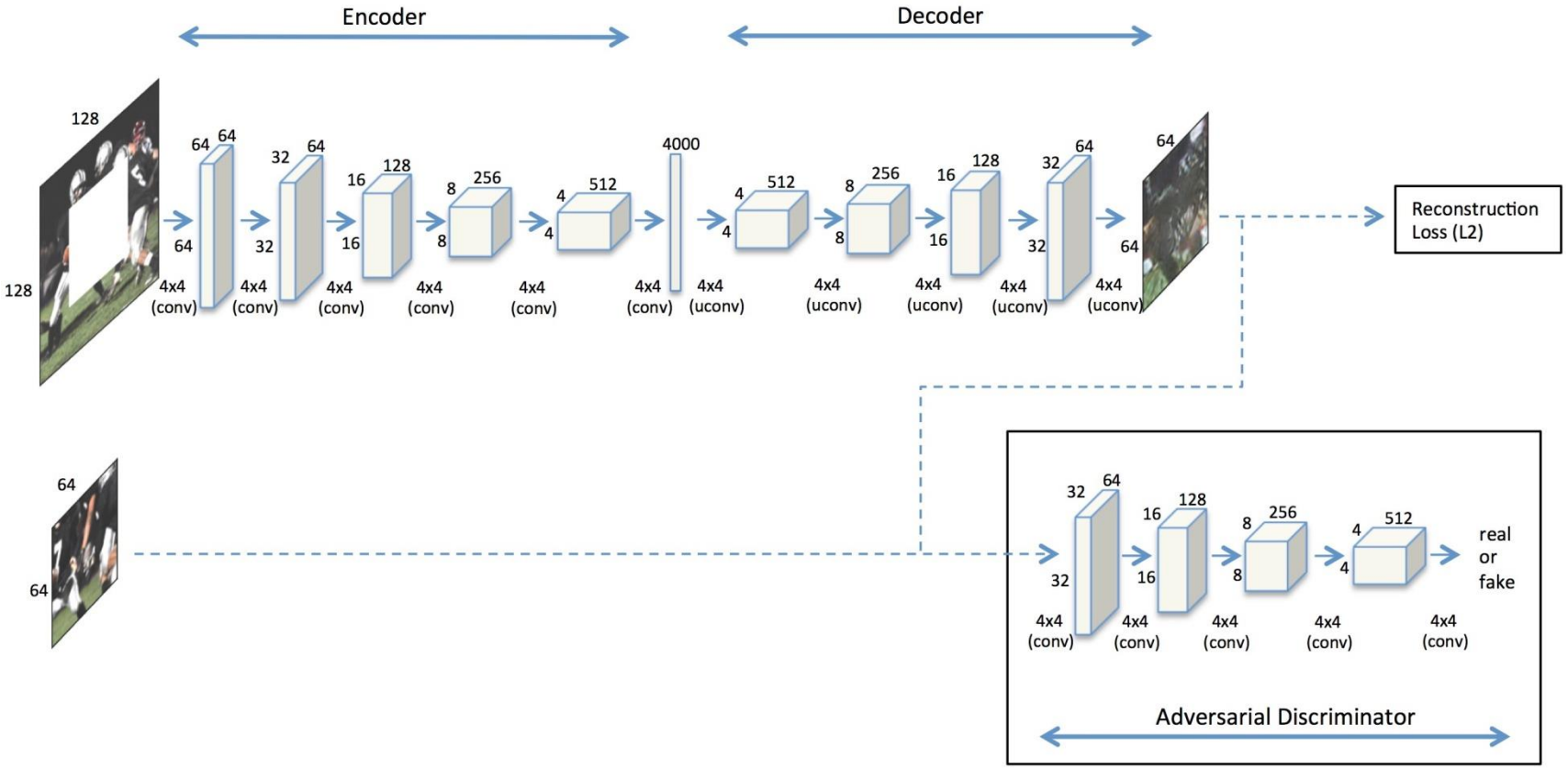
$$\mathcal{L}_{joint}(x) = \lambda_{rec}\mathcal{L}_{rec}(x) + \lambda_{adv}\mathcal{L}_{adv}(x)$$

GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Key Details

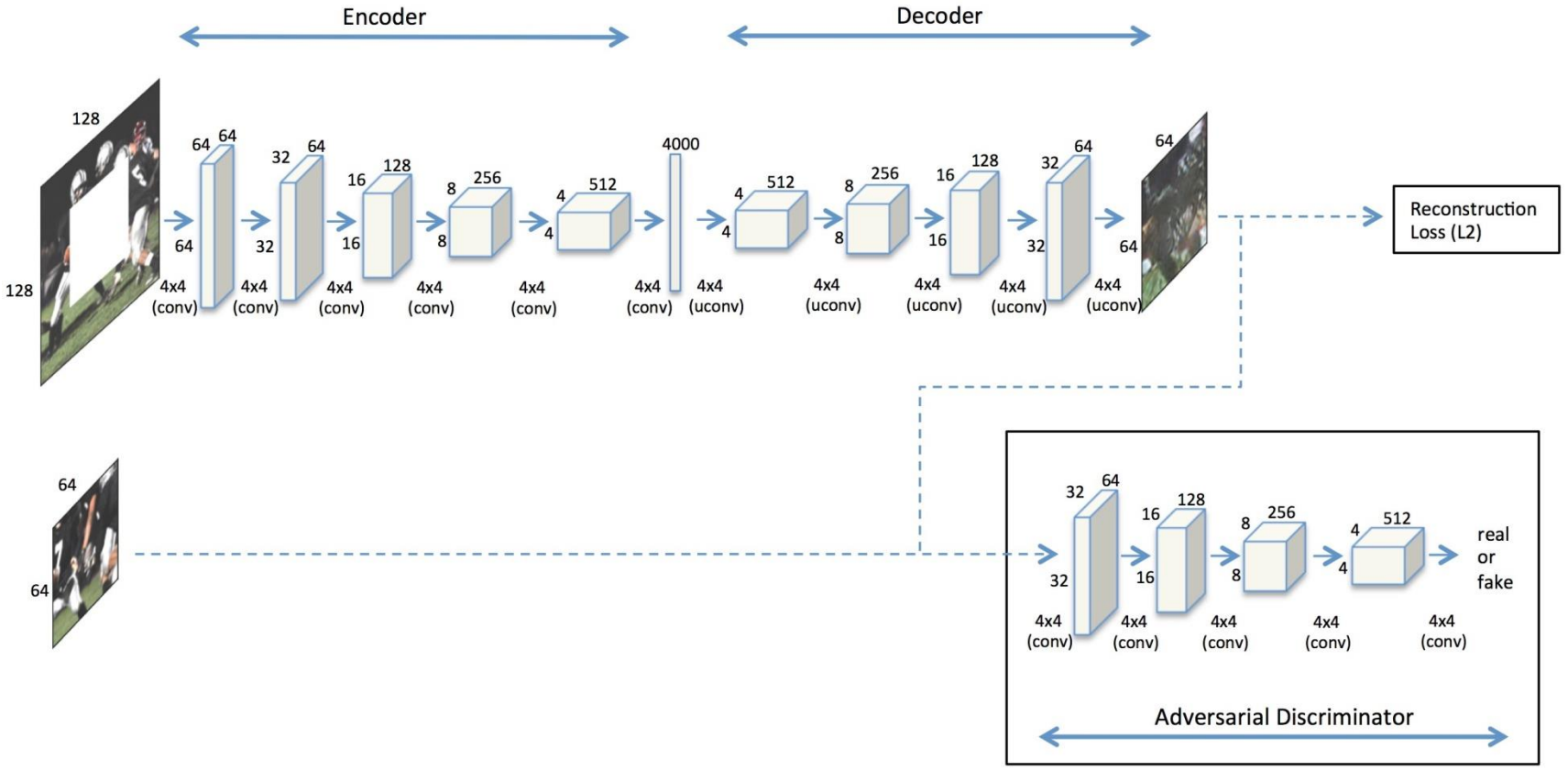
- Encoder is not conditioned on noise
- Discriminator is not conditioned on input



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

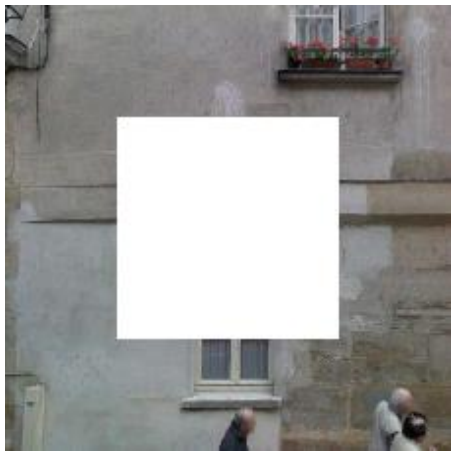
Decoupled Loss

- Reconstruction L_2 loss ensures "**correctness**"
- Adversarial Loss ensures "**realness**"



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Results: Decoupled Loss



Input Image

L2 Loss

Adversarial Loss

Joint Loss

GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

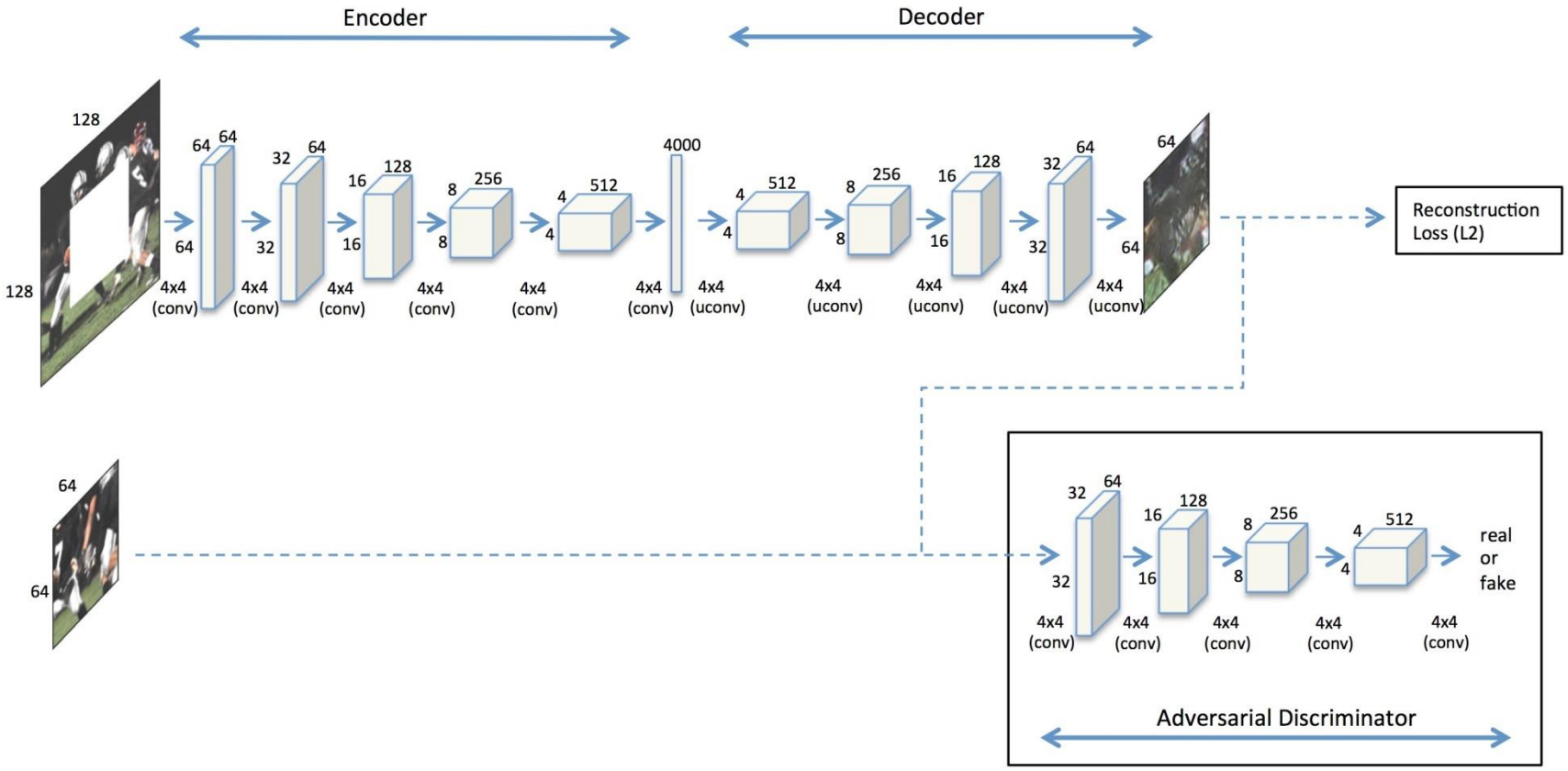
ImageNet Results

Has artifacts near the boundary !

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

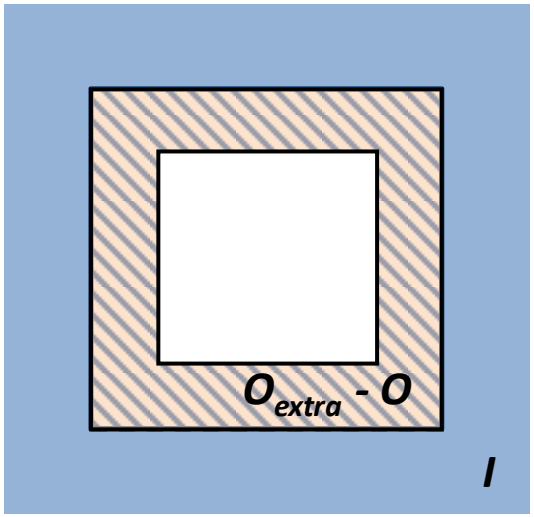
Boundary Artifacts

- Discriminator is not conditioned on input !



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Indirect Conditioning

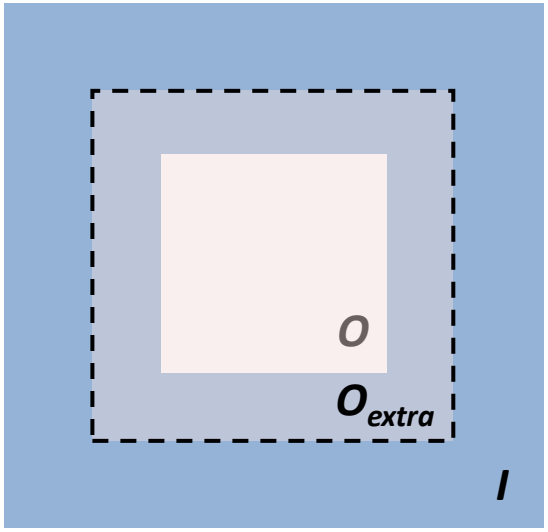


Input Area = ***I***
Target Area = ***O***
Predict Area = O_{extra}
Overlap = O_{extra} - ***O***

Penalize L2 loss heavily in overlapping area
=> so that network memorizes input in the overlap

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Indirect Conditioning



INDIRECT CONSISTENCY

- Adversary ensures O_{extra} is coherent (to be realistic)
- L2 Penalty ensures predicted $O - O_{extra}$ is same as overlapping area with input I

➔ Predicted O is coherent with I

Penalize L2 loss heavily in overlapping area
=> so that network memorizes input in the overlap

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Indirect Conditioning Results

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Indirect Conditioning Results

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

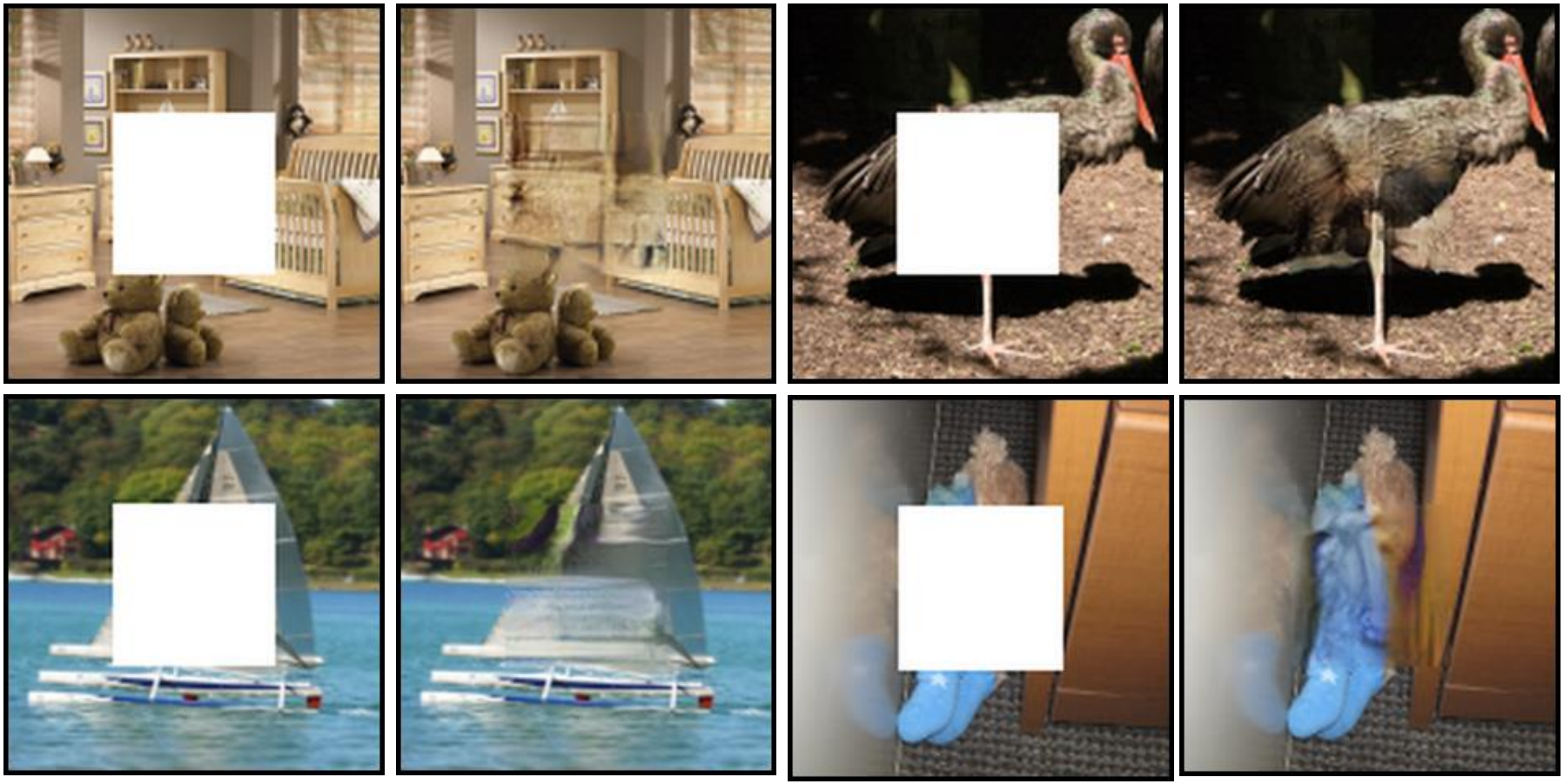
Indirect Conditioning Results

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

More Results

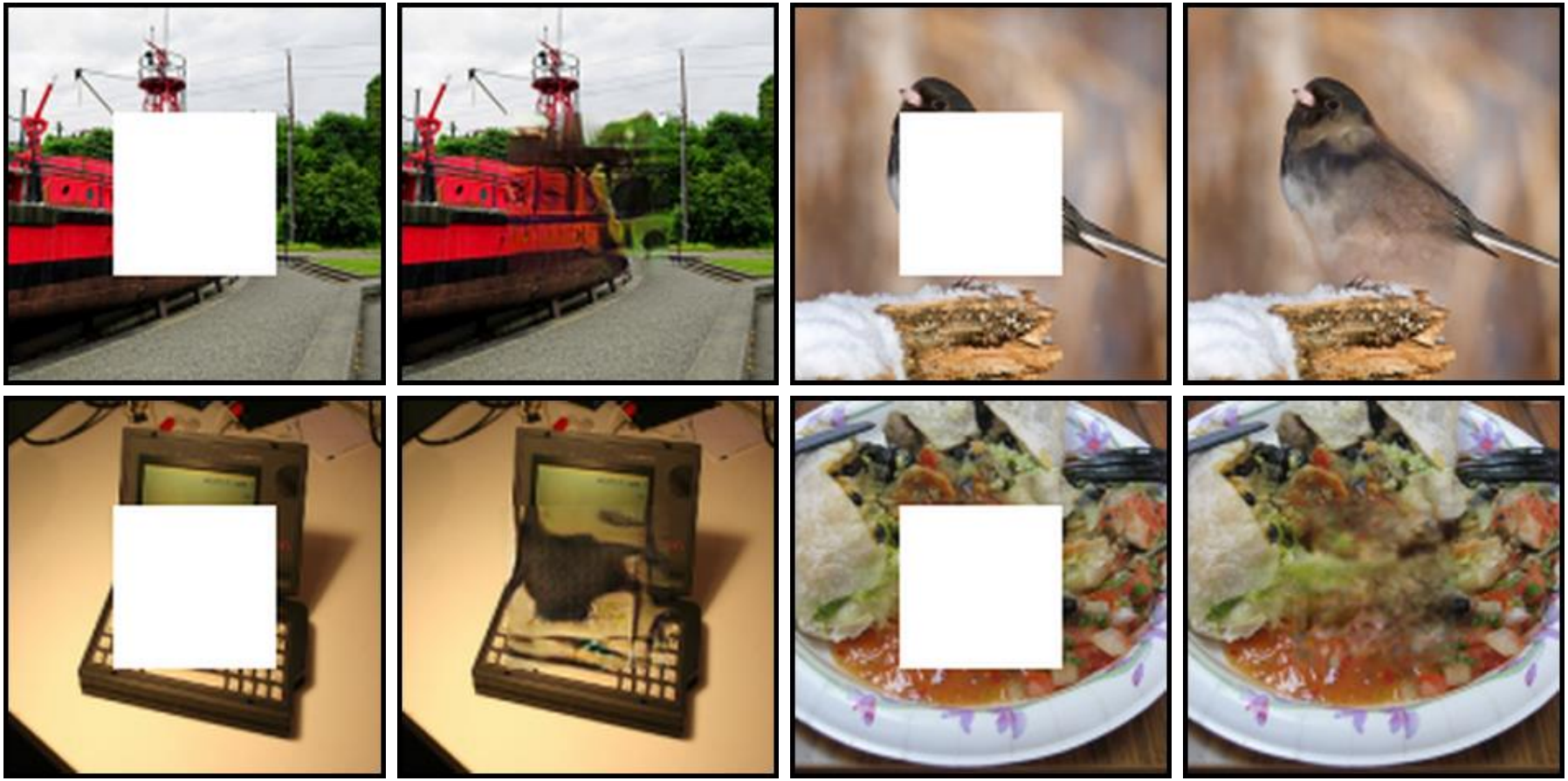
https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

More Results



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

More Results



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

More Results

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Compare

(algorithm in Adobe Photoshop)

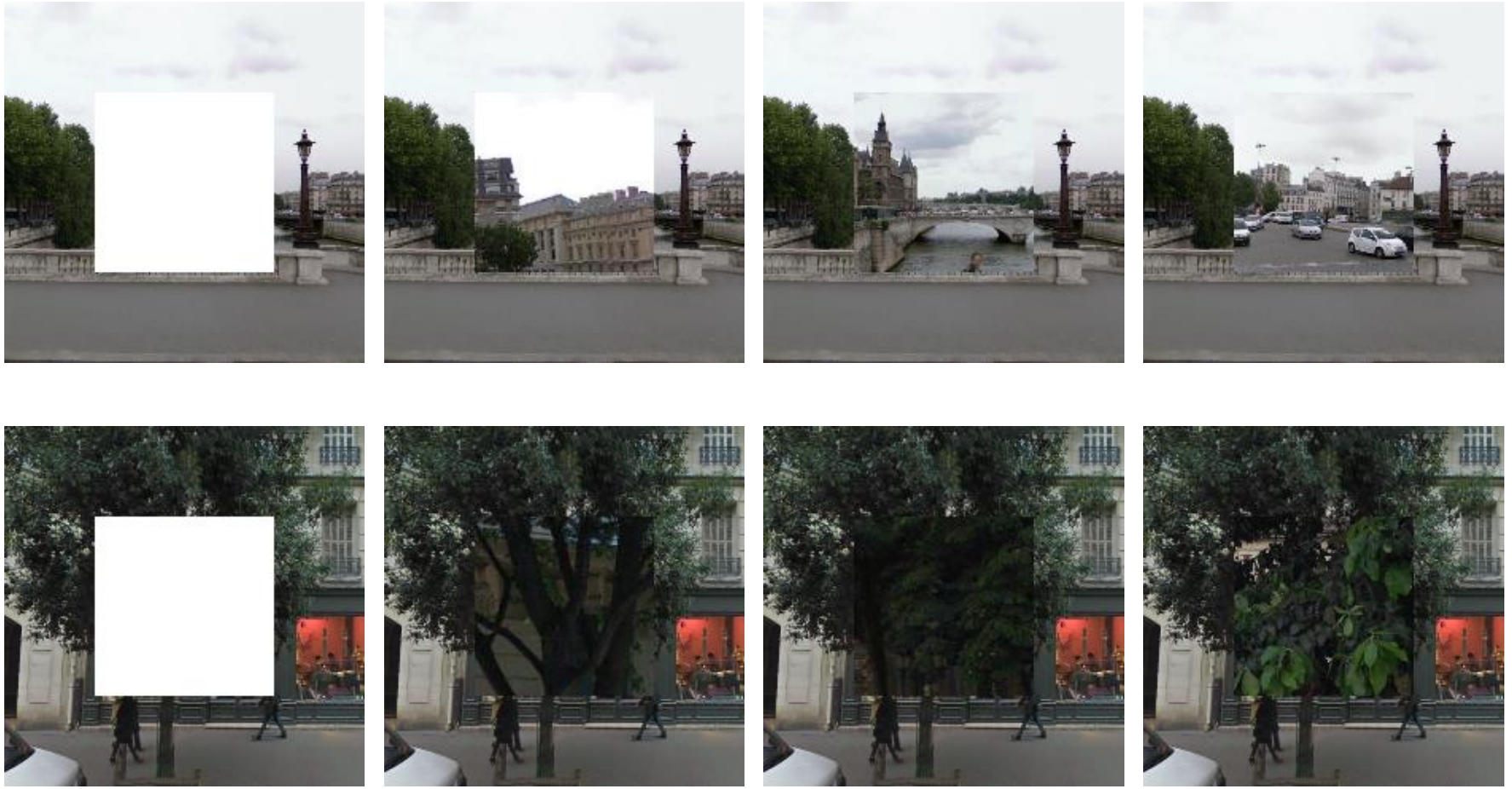
Input

Ours

Content-Aware

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

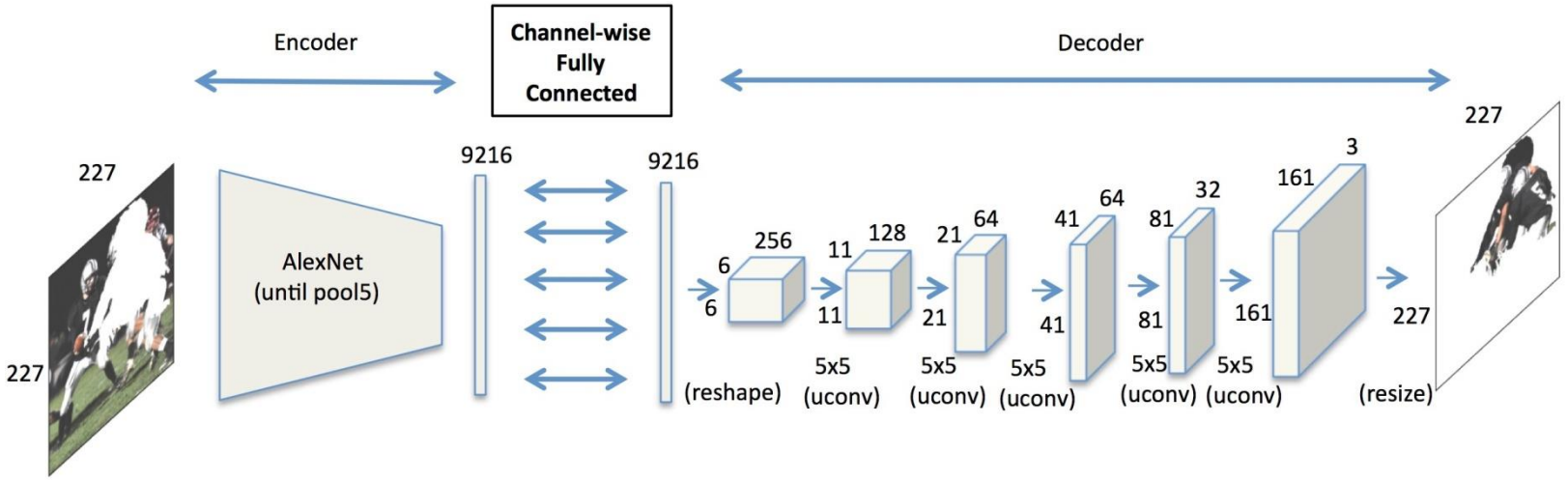
Nearest Neighbor Inpainting



https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Pre-trained Feature Evaluation

- Replace encoder with VGG or AlexNet etc.



- Channel-wise FC reduces $O(n^2)$ to $O(k^2n)$ parameters where $k << n$

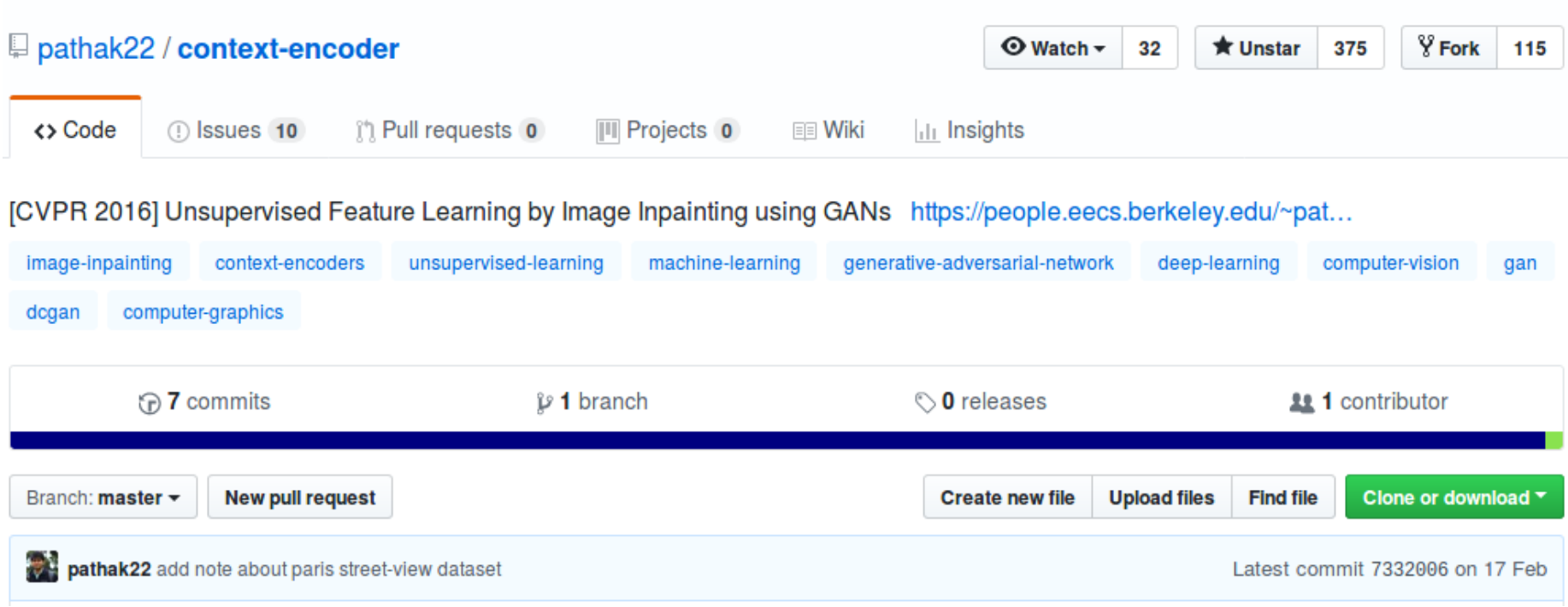
https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Pre-trained Feature Evaluation

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	$<$ 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Open Source



GitHub https://github.com/pathak22/context-encoder

https://people.eecs.berkeley.edu/~pathak/context_encoder/slides/

Thank you !