Semantic Image Inpainting with Deep Generative Models

Raymond A. Yeh,* Chen Chen,* Teck Yian Lim,
Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Do
University of Illinois at Urbana-Champaign
{yeh17, cchen156, tlim11, aschwing, jhasegaw, minhdo}@illinois.edu

CVPR 2017

Slides compiled by Lars Gjesteby

July 5, 2018

Motivation

- Estimate missing image data
- Overcome limitations of existing methods that use information only from a single image

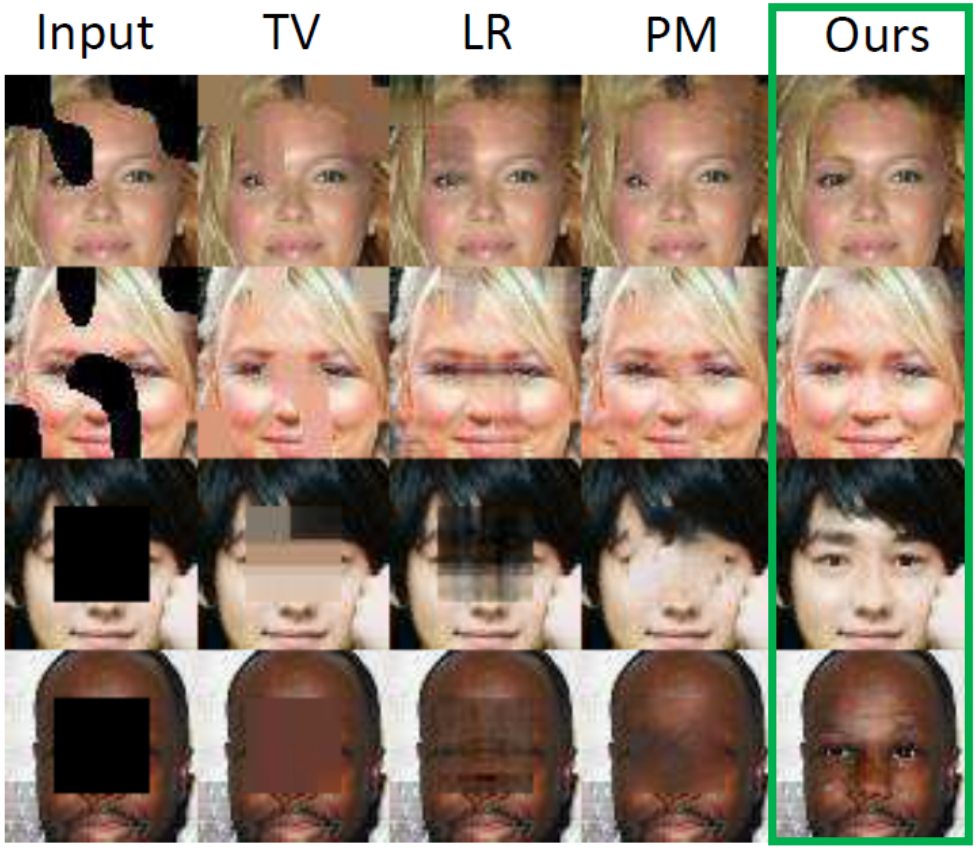
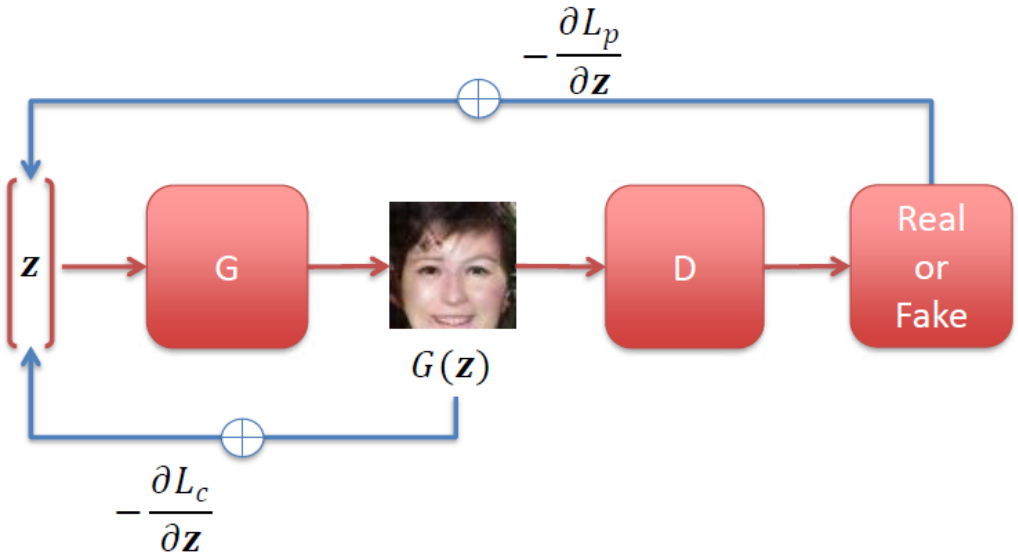


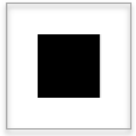
Figure 1. Semantic inpainting results by TV, LR, PM and our method. Holes are marked by black color.

TV – Total variation
LR – Low rank
PM – PatchMatch

Approach

- Semantic inpainting with a generative adversarial network model (DCGAN from Radford *et al.*)
- Train generator on uncorrupted data



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

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

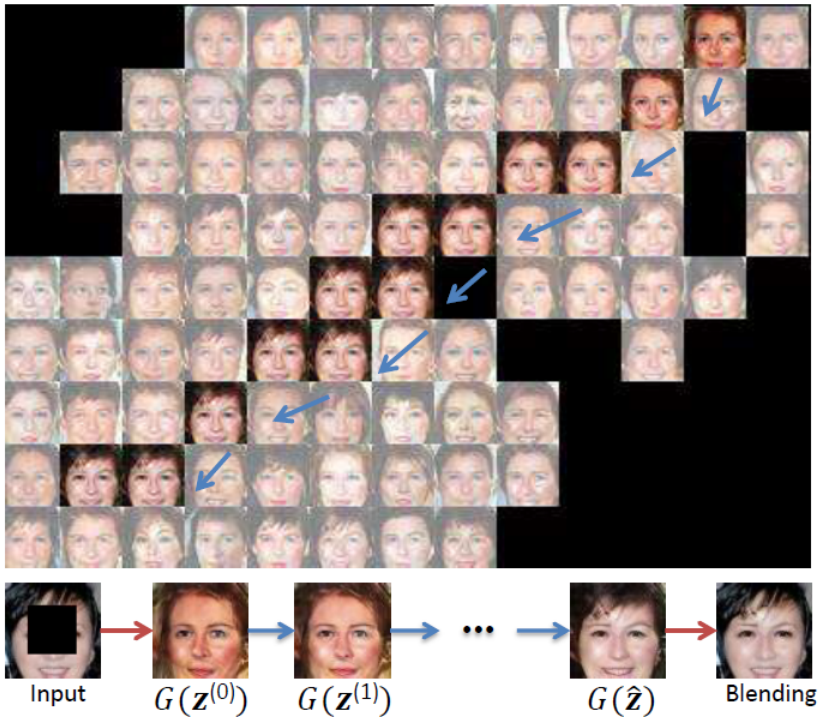
$p_{\mathbf{z}}$ – corrupt data distribution
$p_{\mathbf{data}}$ – true data distribution
z – learned encoding manifold
y – corrupted image
M – binary mask

Approach

- Training G to be efficient – an image not from p_{data} should not lie on **z**
- Iteratively update **z** to find closest mapping to corrupt image while constrained to manifold ($\hat{\mathbf{z}}$)

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

L_c – context loss: constrains generated image to input **y** and mask **M**
L_p – prior loss: penalizes unrealistic images



Loss Functions

- Context loss: a weighted ℓ1-norm difference between the recovered image and the uncorrupted portion

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

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

N(i) – pixel neighbors in a local window

- Prior loss: encourages recovered image to be similar to the samples from training set

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

λ – balancing parameter

Effect of Prior Loss

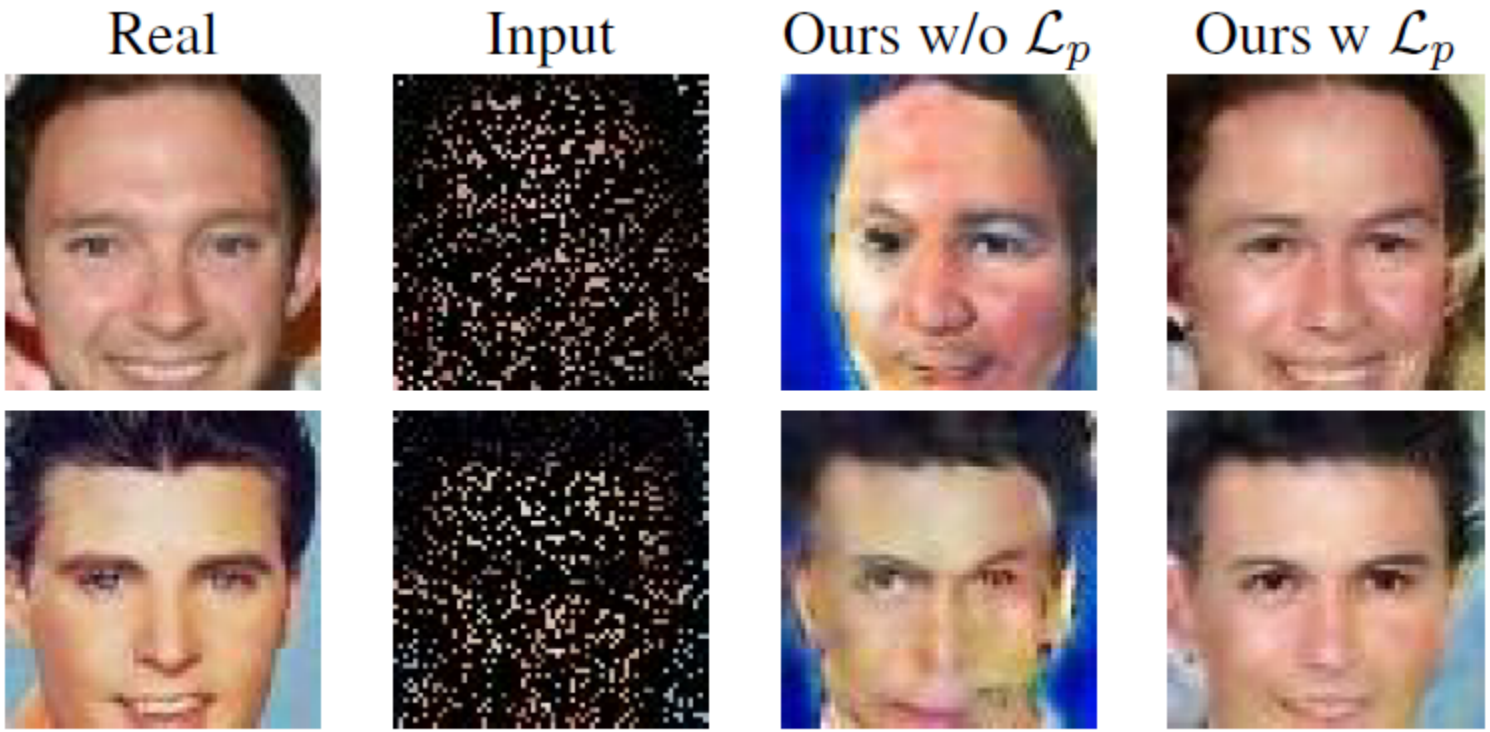


Figure 4. Inpainting with and without the prior loss.

Poisson Blending

- Predicted pixels may not exactly preserve the same intensities of the surrounding pixels, although the content is correct and well aligned
- Keep gradients of G($\hat{\mathbf{z}}$) to preserve image details while shifting the color to match the color in **y**

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\nabla \mathbf{x} - \nabla G(\hat{\mathbf{z}})\|_2^2,$$

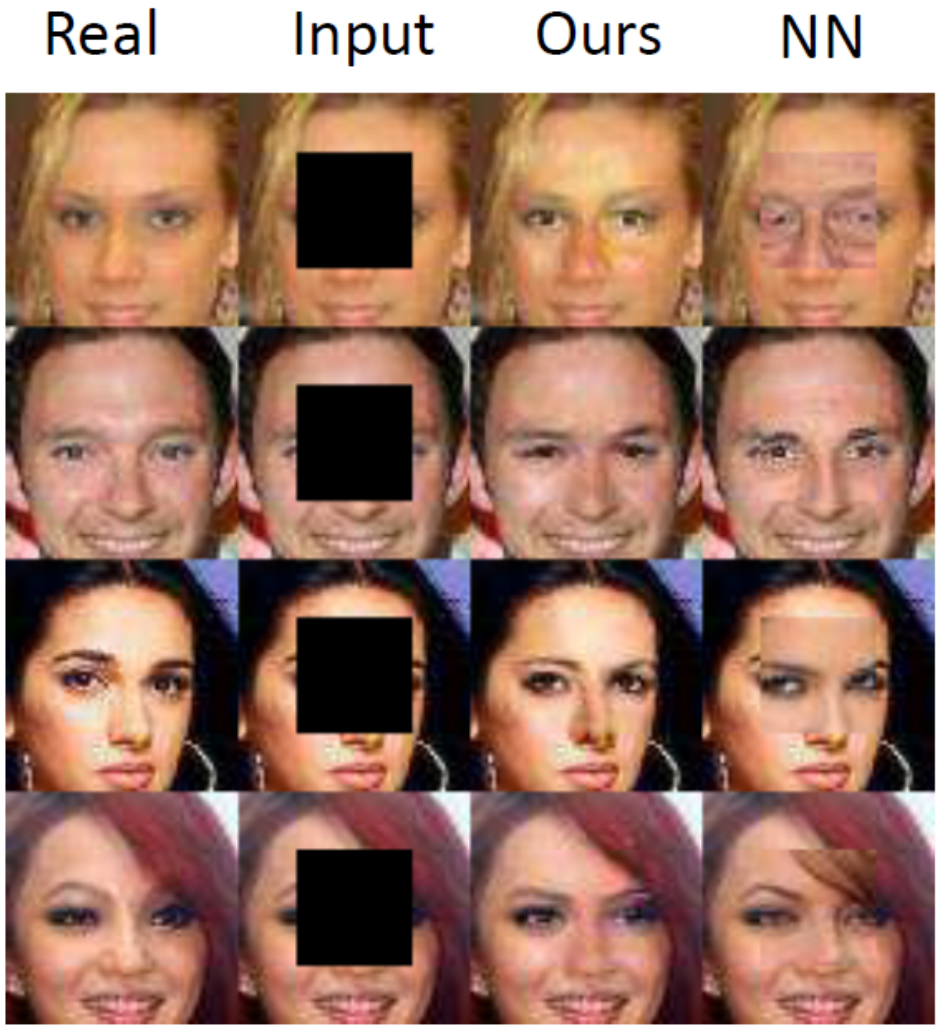
$$\text{s.t. } \mathbf{x}_i = \mathbf{y}_i \text{ for } \mathbf{M}_i = 1,$$

Figure 5. Inpainting with and without blending.

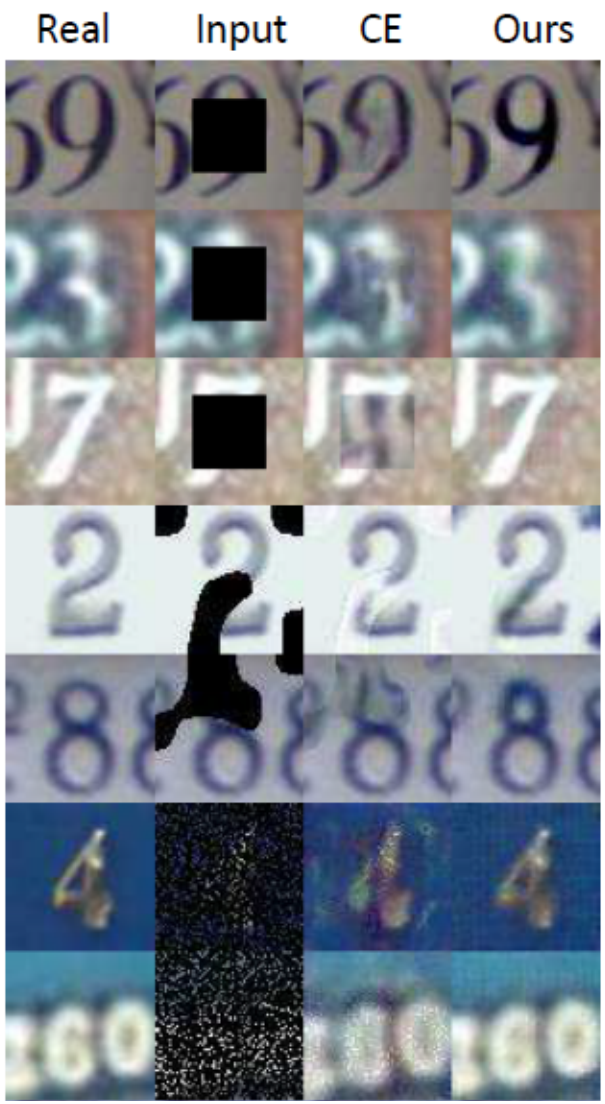
Validation

- Datasets:
  - CelebFaces Attributes (CelebA)
  - Street View House Numbers (SVHN)
  - Stanford Cars
- Four masks:
  1. Central block
  2. Random pattern masks, with approximately 25% missing
  3. 80% missing complete random masks
  4. Half missing masks (randomly horizontal or vertical)

Compare with Nearest Neighbor



Compare with Context Encoder



Quantitative Results

Table 1. The PSNR values (dB) on the test sets. Left/right results are by CE[30]/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

Paper Conclusions

- Proposed model provides better visual image quality over state-of-the-art methods
- Works well for relatively simple structures like faces, but is too small to represent complex scenes in the world
- Prediction performance strongly relies on the generative model and the training procedure

Figure 13. Some failure examples.