Image Inpainting Survey

Tomoki Tanimura (@tanimu)
d-hacks, Jin Nakazawa Lab, SFC, Keio University

What is Image Inpainting

- Image Inpainting is to fill missing parts of image.
- Important Point: Fill in the hole with
  - visually reaslistic
  - semantically plausible

Two main approach

- Patch Pasting
  - Search similar patches to hole-surrounding areas
  - PatchMatch, Scene Completion, etc.
- Learning based
  - Learn how to fill the hole using large scale dataset
  - GLCIC, Gated Conv, etc.

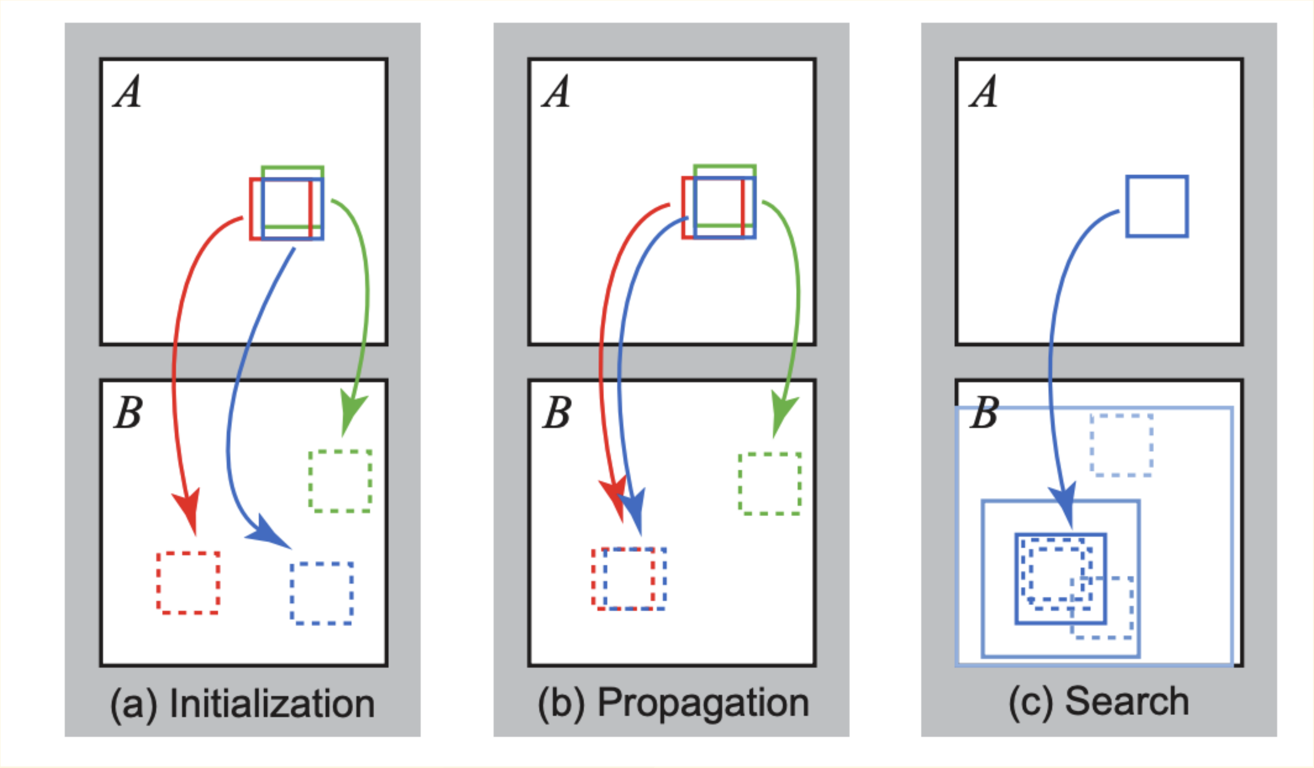
Patch Pasting

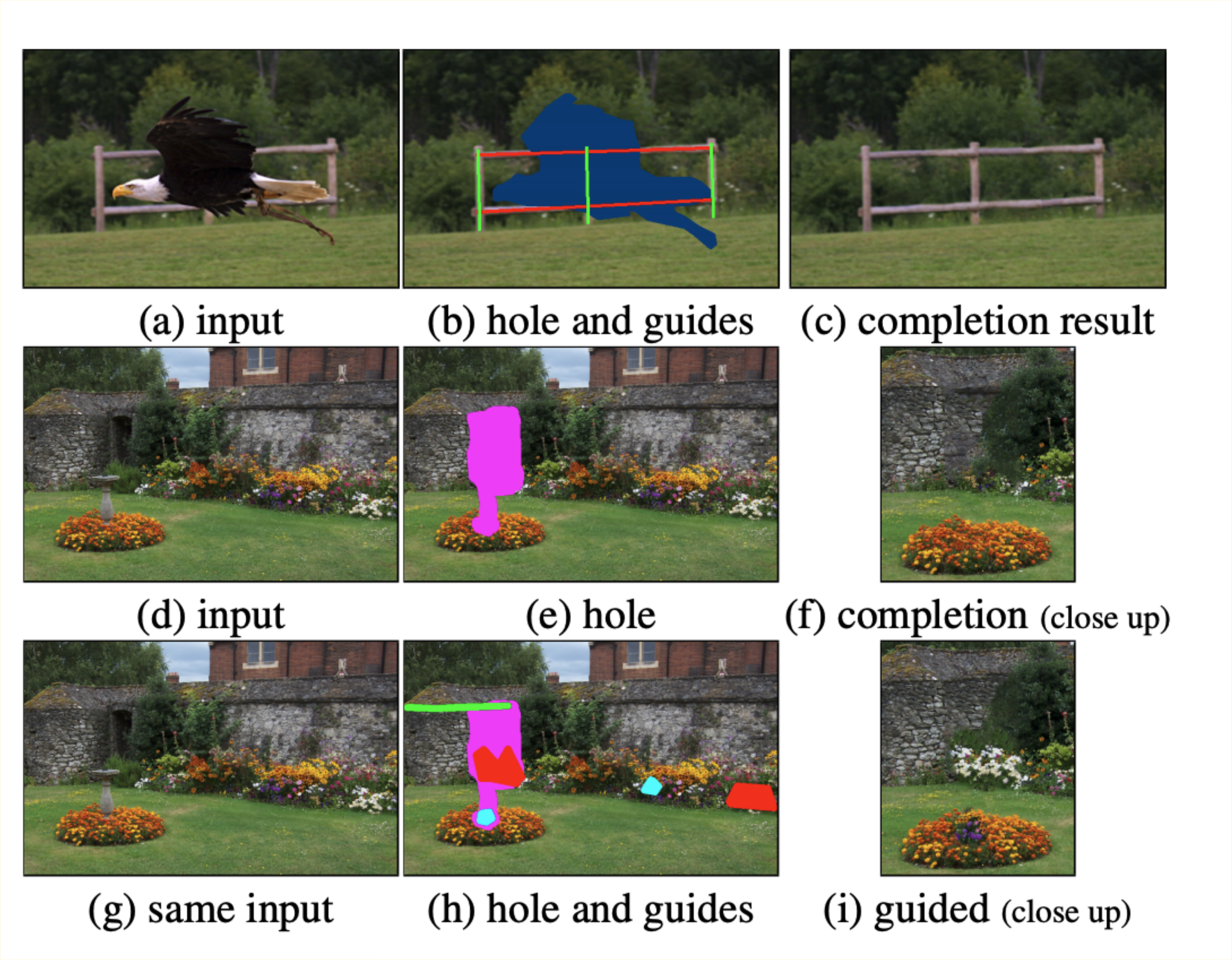
2006 - 2008

- Search Similar Patch !

PatchMatch

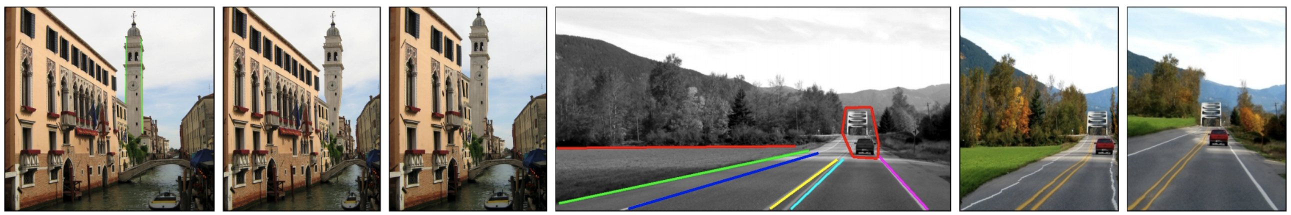
1. Search Similar patch to the hole surrounding areas in an input image
2. Paste the searched patch to the hole
3. loop (1,2)


A
B
(a) Initialization
A
B
(b) Propagation
A
B
(c) Search


(a) input
(b) hole and guides
(c) completion result
(d) input
(e) hole
(f) completion (close up)
(g) same input
(h) hole and guides
(i) guided (close up)

(a) (b) (c) (d) (e)

Figure 10: *Retargeting. From left: (a) Input image, (b) [Rubinstein et al. 2008], (c) [Wang et al. 2008], (d) Our constraints, (e) Our result.*



(a) (b) (c) (a) (b) (c)

Figure 11: *Retargeting: (a) Input image, annotated with constraints, (b) [Rubinstein et al. 2008], (c) Our output.*

Scene Completion

- Search the image similar to an input from large scale image database
- Search similar patch and paste it to the input's hole

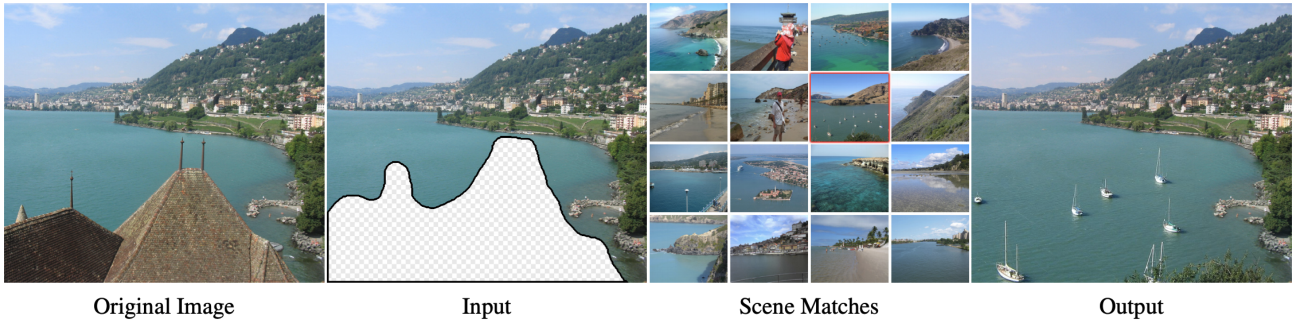


Figure 1: Given an input image with a missing region, we use matching scenes from a large collection of photographs to complete the image.

Original Input Alternative Completions

Figure 5: The algorithm presents to the user a set of alternative image completions for each input. Here we show three such alternatives.

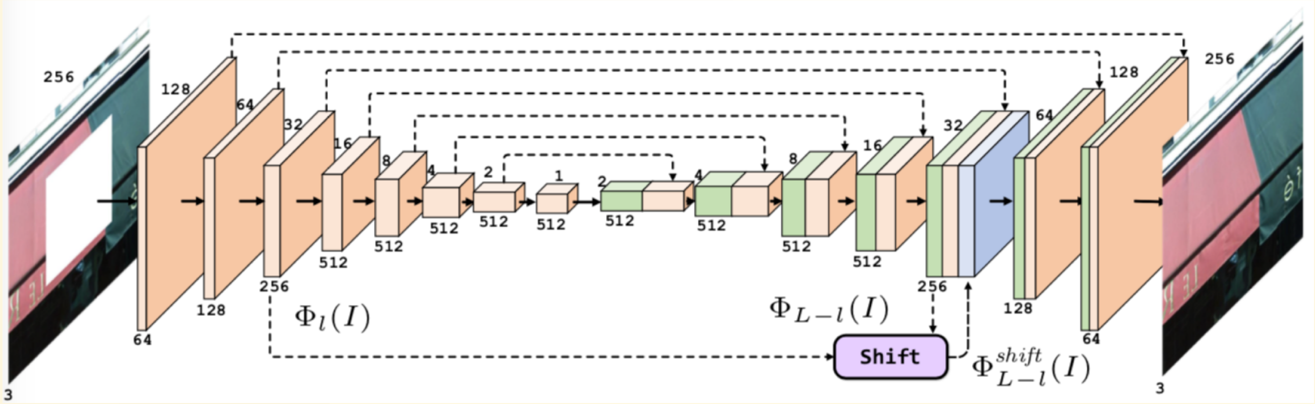
Patch Pasting

- Good at filling in the small size hole and the hole in a stationary image (specific pattern image, scene image, etc.)
- Can't fill the hole with semantically prosible contents, Not good at filling the hole in a non-stationary image (city, face, etc.)

Learning based

Learning based

- Recent popular approach
- Generate the contents of the hole using Convolutional Neural Networks (CNN)
    - Input an image with hole to a model
    - Model predict a contents in the hole



Evolution of Learning based ImIn

1. CNN Encoder-Decoder + Reconstruction Loss + Adversarial Loss
   -> e.g. ContextEncoder, GLCIC
2. 1 + Additional Loss or Input
   -> e.g. GICA, EdgeConnect, StructureImIn
3. (1 + 2 +) Feature Gating
   -> e.g. PartialConv, GatedConv

1. CNN Encoder-Decoder

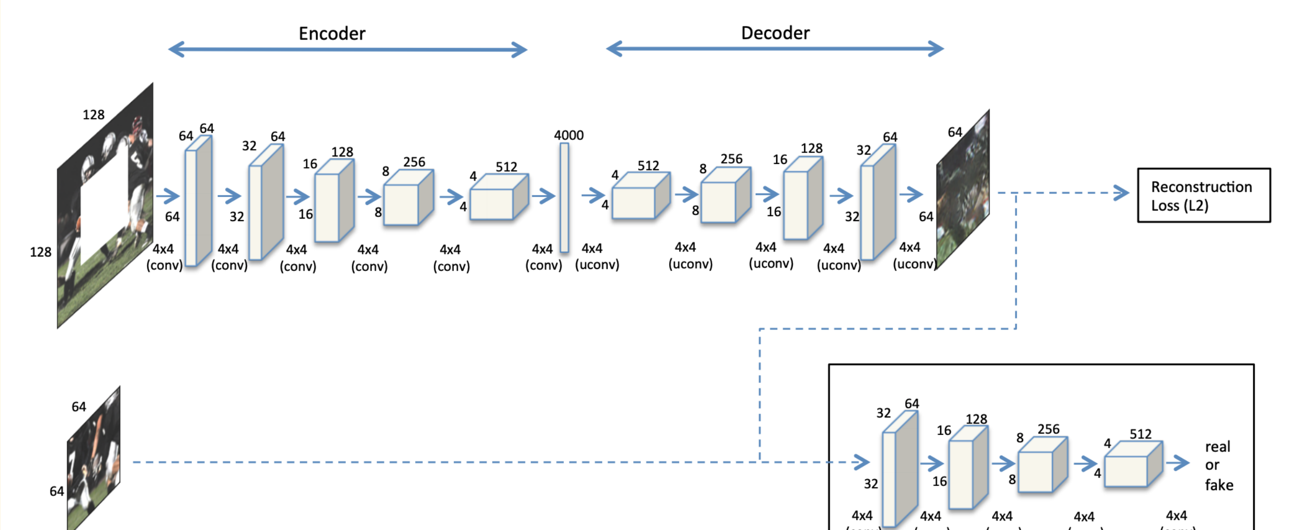
2016 - 2017

- Initial Deep Learning based approach
- Use CNN and Adversarial Loss

Context Encoder

Context Encoders: Learning Features by Inpainting

1. Input the image with hole to CNN
2. CNN encode the input to feature vector, then decode to the input while predicting the hole
3. Update model parameter to minimize the difference between the decoded image and the original input



Generate the semantically prosible contents in the hole

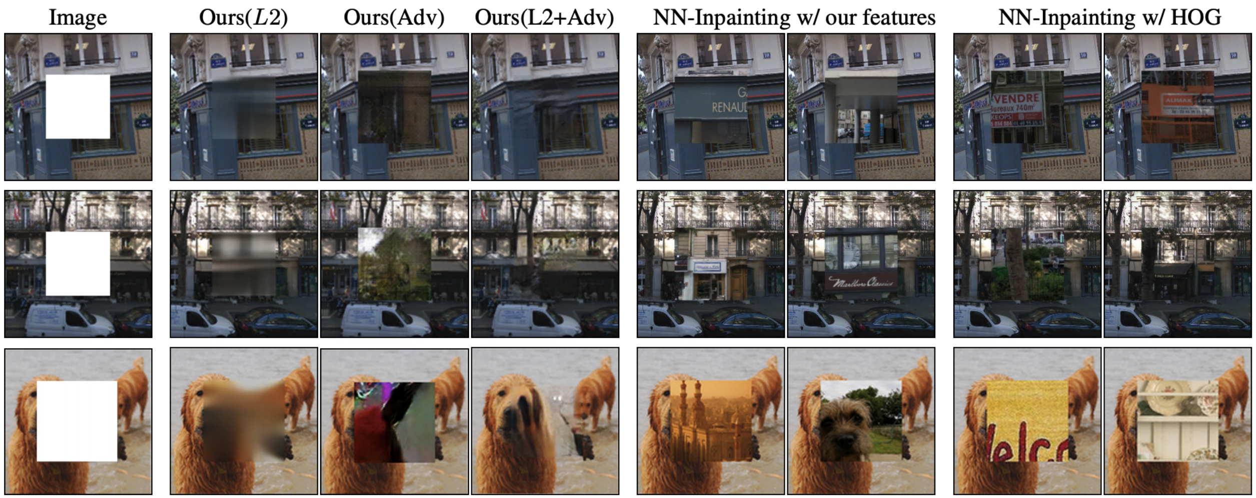


Figure 6: Semantic Inpainting using different methods on *held-out* images. Context Encoder with just L2 are well aligned, but not sharp. Using adversarial loss, results are sharp but not coherent. Joint loss alleviate the weaknesses of each of them. The last two columns are the results if we plug-in the best nearest neighbor (NN) patch in the masked region.

- Use ImIn as Representation Learning
- Use the encoder as feature extractor and test some vision tasks

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%

Table 2: Quantitative comparison for classification, detection and semantic segmentation. Classification and Fast-RCNN Detection results are on the PASCAL VOC 2007 test set. Semantic segmentation results are on the PASCAL VOC 2012 validation set from the FCN evaluation described in Section 5.2.3, using the additional training data from [18], and removing overlapping images from the validation set [28].

GLCIC

Globally and Locally Image Completion

Difference from Context Encoder

- Add dilated convolution
- Evaluate generated image with globally and locally discriminator

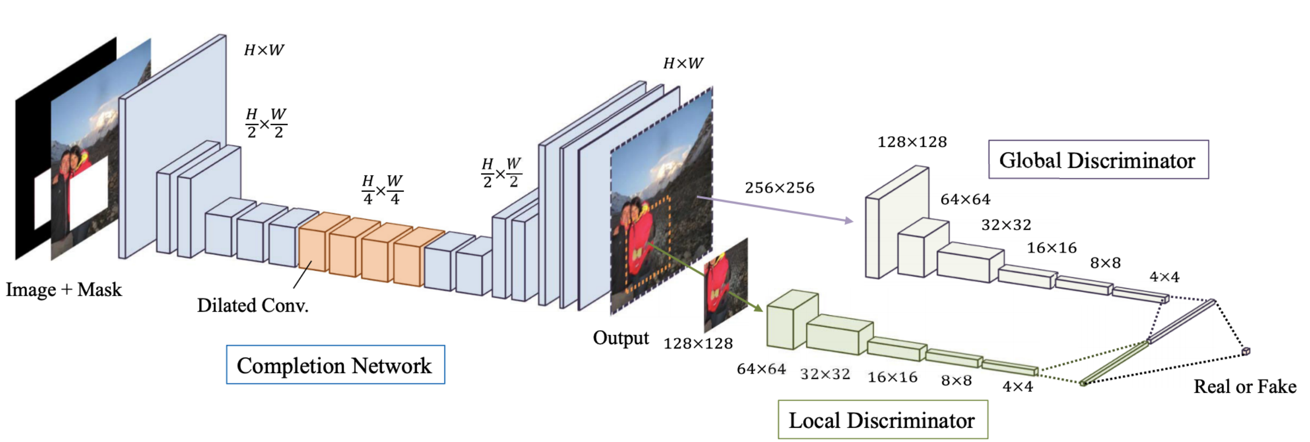


Fig. 2. Overview of our architecture for learning image completion. It consists of a completion network and two auxiliary context discriminator networks that are used only for training the completion network and are not used during the testing. The global discriminator network takes the entire image as input, while the local discriminator network takes only a small region around the completed area as input. Both discriminator networks are trained to determine if an image is real or completed by the completion network, while the completion network is trained to fool both discriminator networks.

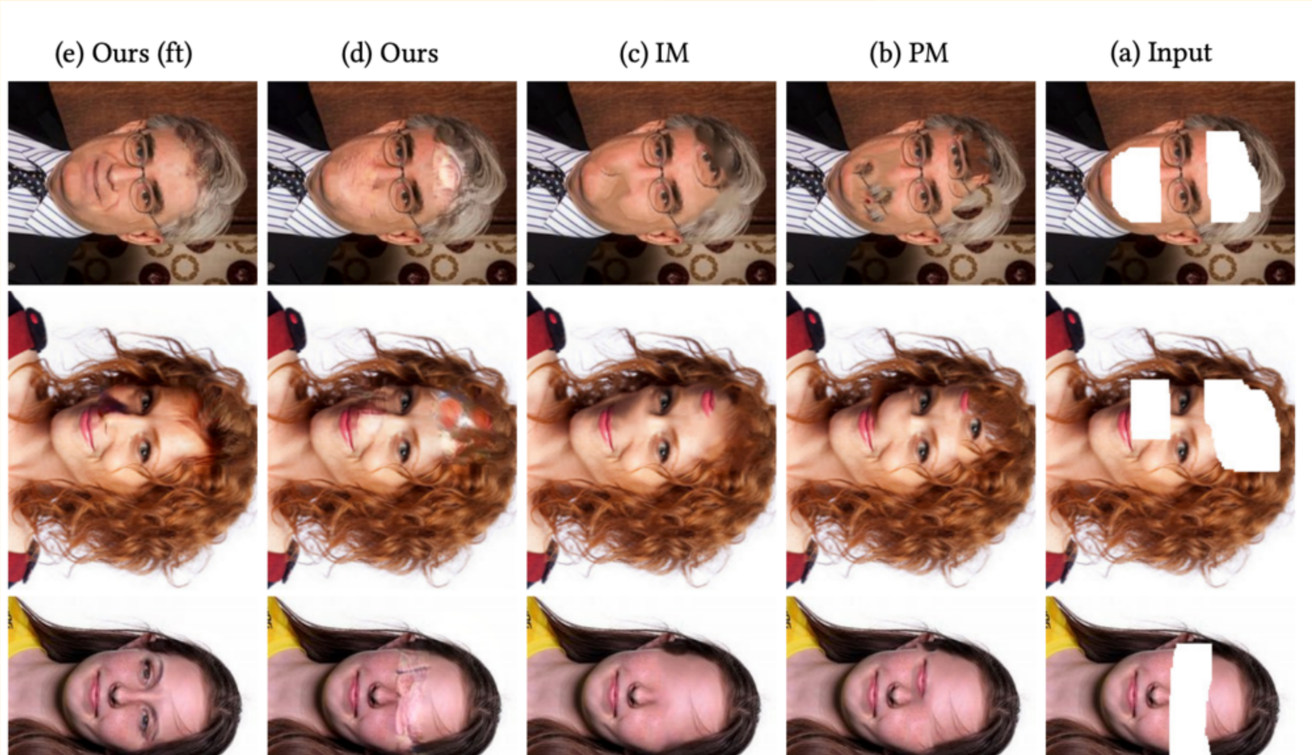
- Fill the hole with semantically prosible and visually realistic contents
    - Effect of Globaly Discriminator
- Deal with high resolution image and non-stationary image

Fig. 1. Image completion results by our approach. The masked area is shown in white. Our approach can generate novel fragments that are not present elsewhere in the image, such as needed for completing faces; this is not possible with patch-based methods. Photographs courtesy of Michael D Beckwith (CC0), Mon Mer (Public Domain), davidgsteadman (Public Domain), and Owen Lucas (Public Domain).

CNN Encoder-Decoder

Advance

- CNN and Adversarail Loss enable to learn to generate the semantically prosible contents using large scale dataset



Drawback

- Context Encoder is not perfect
- GLCIC requires heavy post process (poisson blending)

2. Additional Loss and Input

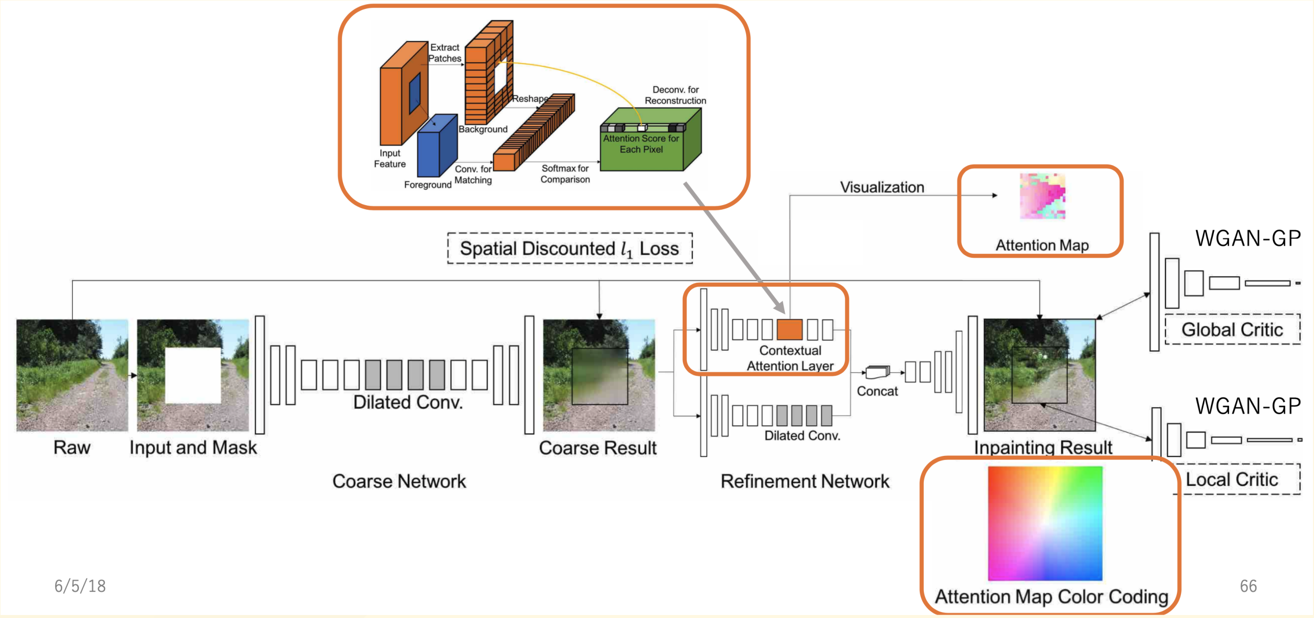
2018 - 2019

- Use additional information and module to improve ImIn performance

GICA

Generative Image Inpainting with Contextual Attention

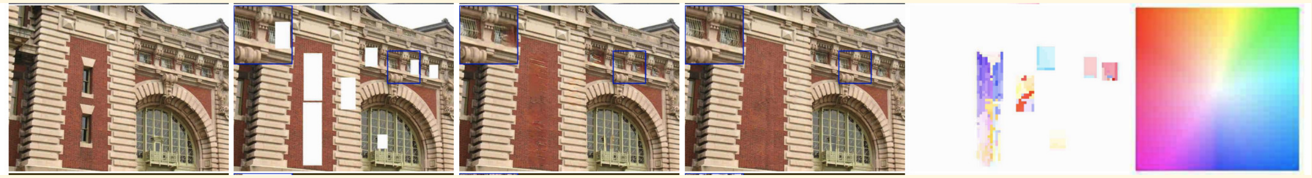
- GLCIC + Coarse-to-Fine + Contextual Attention



- Is there difference from GLCIC ?

Mask Iizuka Baseline

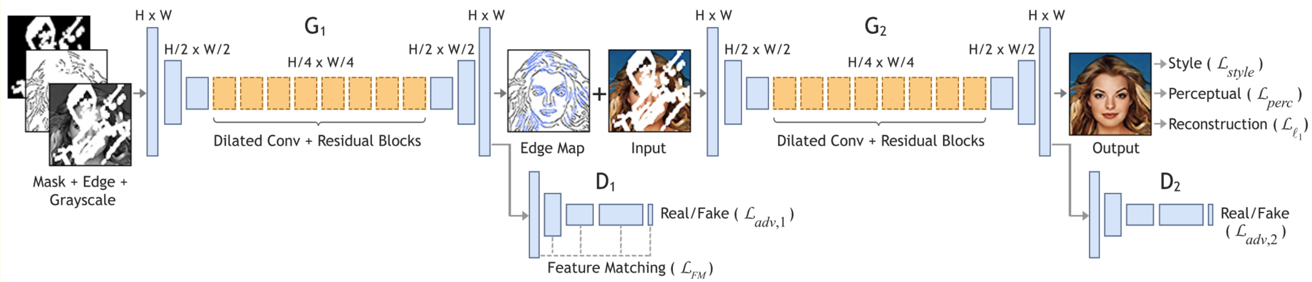
- Pay attention to image structure



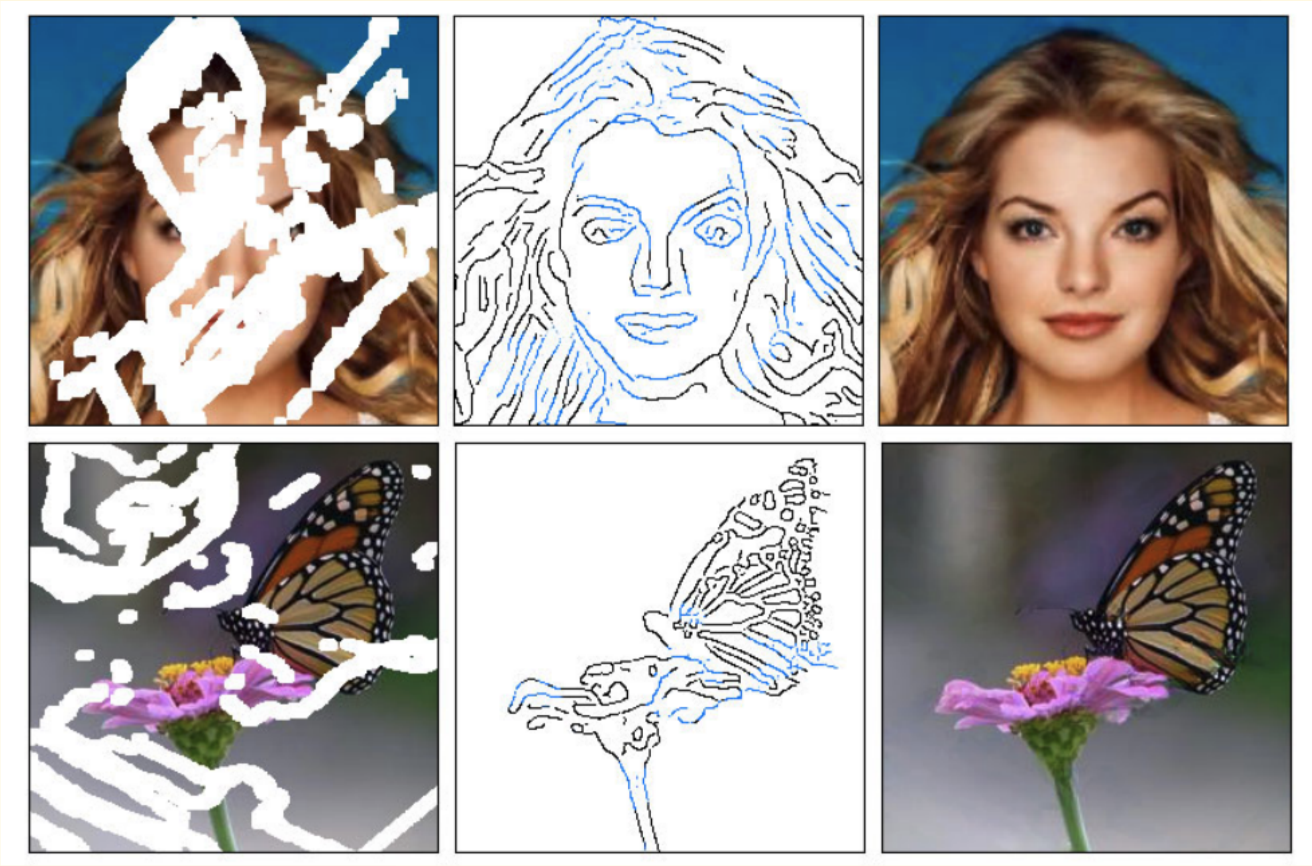
EdgeConnect

EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning

- Use edge information to support image inpainting model
  - Edge image with hole -> **Model** -> Completed edge image
  - Completed Edge image, Color image with hole -> **Model** -> Completed color image



- EdgeConnect is inspired by **"lines first, color next"**



3. Feature Gating

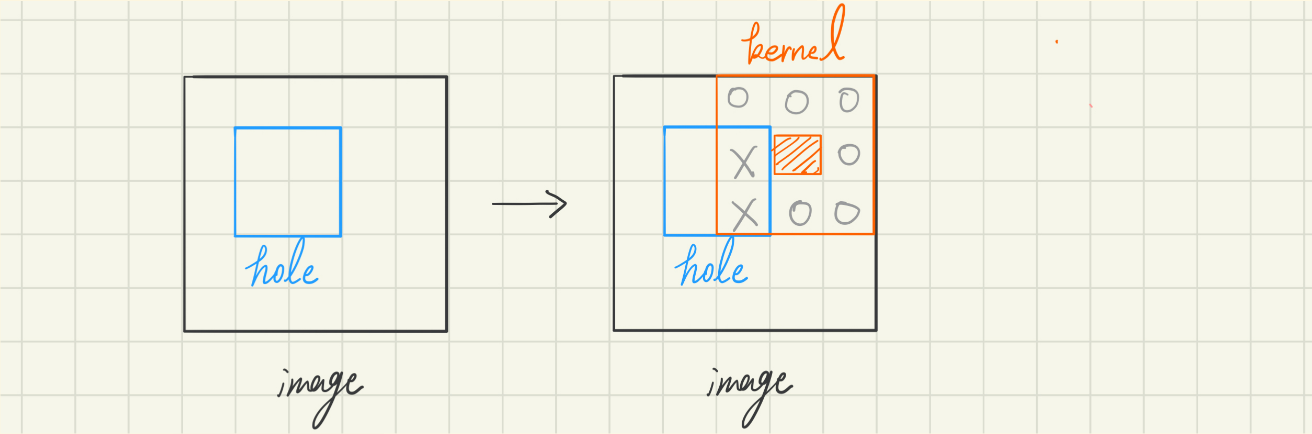
2018 - 2019

- Filter invalid pixel information to improve performance
- Get better performance with simple and reasonable extension

Partial Convolution

Image inpainting for irregular holes using partial convolutions

- Use pixel information other than holes
- Introduce **Mask** to filter pixel information



$$x' = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M}) \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(\mathbf{M})} + b, & \text{if } \mathrm{sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

- More effective to irregular holes because of not using hole area pixel information
- Achieve good performance with a simple U-Net architecture

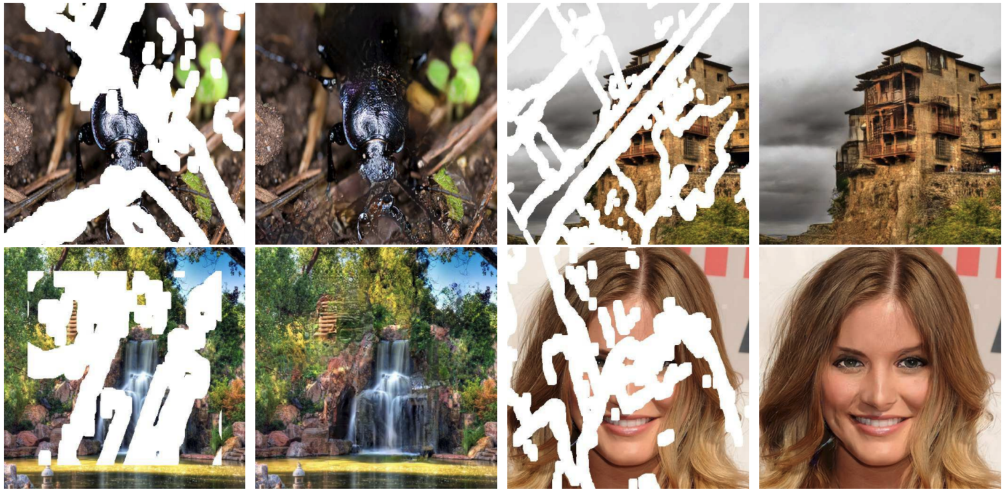


Fig. 1. Masked images and corresponding inpainted results using our partial-convolution based network.

Gated Convolution

Free-form image inpainting with gated convolution

- Introduce soft mask feature gating with **Learnable Mask**
- Enable for user to sketch the guide to input

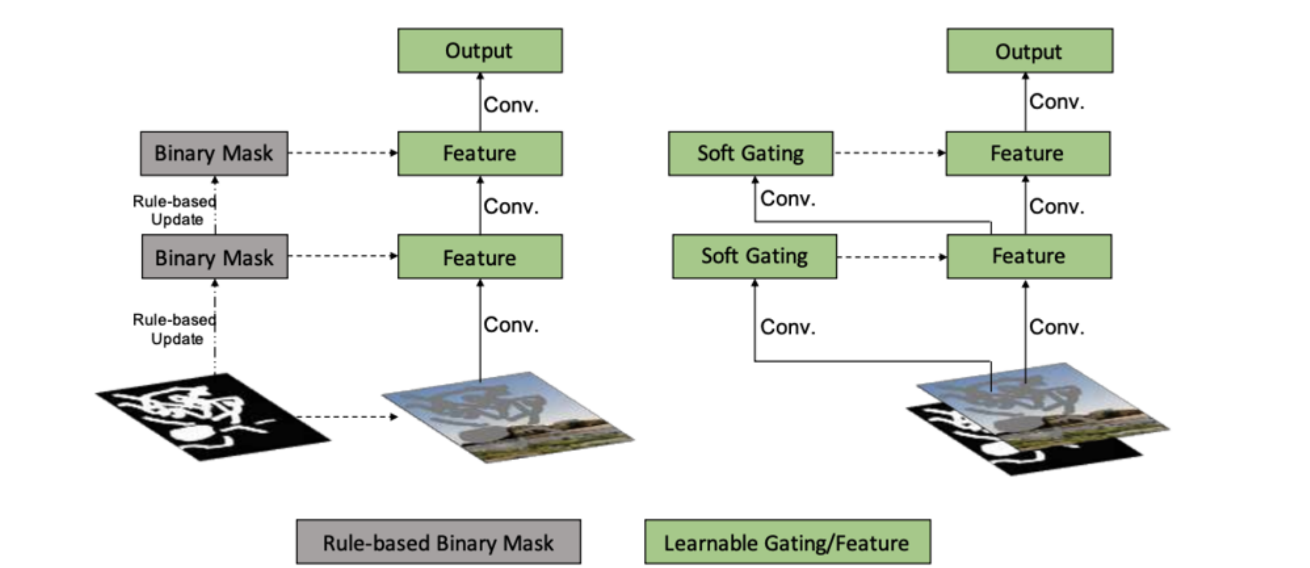


Figure 2: Illustration of partial convolution (left) and gated convolution (right).

- Use Convolution Kernel and SoftMask Kernel

$$Gating_{y,x} = \sum\sum W_g \cdot I$$
$$Feature_{y,x} = \sum\sum W_f \cdot I$$
$$O_{y,x} = \phi(Feature_{y,x}) \odot \sigma(Gating_{y,x})$$

- Use the same architecture as GICA
- Introduce Spectral-Normalized Markovian Discriminator

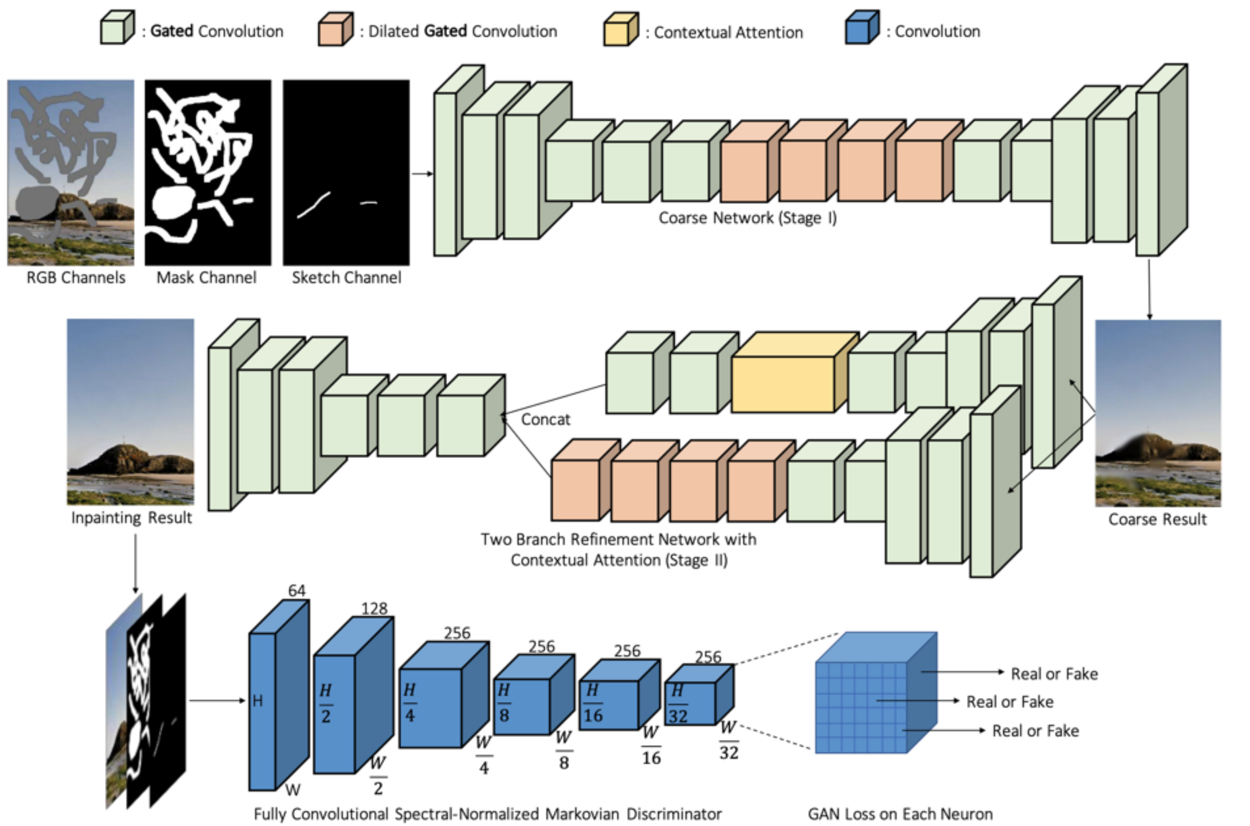


Figure 3: Overview of our framework with gated convolution and SN-PatchGAN for free-form image inpainting.

Summary

How ImIn has evolved

- Patch Pasting
  - **Traditional Heuristic Approach** (reasonable, but worse performance)
  - Search patches similar to hole surroundings and paste
- CNN Encoder-Decoder with Adversarial Loss
  - **Appear Deep Learning! High Perfomance!**
  - CNN + GAN (discriminator)
- Additinal Loss or Input
  - **Attension or Some Extension**
  - Attension / Edge Information

- Feature Gating
    - **Improve More Fundemental Problem**
    - Use only valid pixel information
- (None)
    - (Theoretical Analysis and Improvement)

References

- PatchMatch: a randomized correspondence algorithm for structural image editing
- Scene Completion Using Millions of Photographs
- Context Encoders: Feature Learning by Inpainting
- Globally and locally consistent image completion
- Generative Image Inpainting with Contextual Attention
- EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning
- Image Inpainting for Irregular Holes using Partial Convolution
- Free-form Image Inpainting with Gated Convolution