RePaint: Inpainting using Denoising Diffusion Probabilistic Models

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, Luc Van Gool

Group 6
Zhaoning Wang, Jeffrey Chan, Qingyuan Li, Kevin Samms
Outline

● **Introduction**
  ○ Problem statement
  ○ Motivation

● **Method**
  ○ Overall Method
  ○ Resampling + Jumping

● **Evaluation**
  ○ User study & Objective analysis
  ○ Qualitative Analysis

● **Limitations**

● **Demo**
Background/Motivation
Image Inpainting

- Photo-realistic
- Semantically consistent
- One-to-Many Generation
Existing approaches train with a given mask distribution.
Existing approaches train with a given mask distribution

Mask Distribution  Image Dataset

MODEL

poor generalization to extreme cases
RePaint

No Training

MODEL

with generalization to extreme cases
Preliminary: DDPM

Forward process (rewritten using independence property of noise added at each step)

\[
q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \quad \rightarrow \quad q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t} x_0, (1 - \bar{\alpha}_t) I)
\]

Reverse process

\[
p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))
\]

Neural network predicts
Known and Unknown X

\[ x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}} \]

Known is obtained from Forward Process

\[ x_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \]

Unknown is obtained from the denoise process

\[ x_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \]
Method
With a mask like this, how can we create the following diffusion process with a **pre-trained** Diffusion model?

\[ X_0 \sim q \quad \xrightarrow{\text{noise}} \quad \xrightarrow{\text{denoise}} \quad X_T \sim q \]

\[ X_0 \sim p \quad \xrightarrow{\text{noise}} \quad \xrightarrow{\text{denoise}} \quad X_T \sim p \]
$X_{T-1} \sim q$ \hspace{2cm} $X_T \sim q$

$X_{T-1} \sim p$ \hspace{2cm} $X_T \sim p$
$X_{T-1} \sim p$

$X_T \sim q$

$X_T \sim p$

masked $X_T \sim p$

denoise

$X_{T-1} \sim q$
With a mask like this, how can we create the following diffusion process?
Resampling + Jumping

This approach somewhat works but introduce semantically disharmony
The main issue is when sampling the original noise (red), it has no information about the generated part (non-red).
Resampling

- Diffusion model can harmonize image in a single steps but the process is inconsistent

- Solution: Repeat the noising and de-noising steps during inference
Resampling with more steps (r = 10)

dis-harmonized $X_{T-1}$

harmonized $X_{T-1}$

$X_T$

$\{ \ldots \}$

r = 10
Jumping

- Reason:
  - Resampling at every step would make the image blur

- Only do resampling every \( j \) time stamps, starting from \( T-j-1 \).
  - For example, when \( j = 10 \) and \( T = 250 \), only do resampling when \( t = 230, 220, 210 \ldots \), and the length of resampling would be 10
Resampling with jump length ($j = 10$)
Resampling + Jumping visualization

$r = 10, j = 10, T = 250$
Metrics
Setup

Datasets

- CelebA-HQ: High Quality dataset of celebrity faces, 30,000 images, 1024 x 1024
- ImageNet: Image database organized by nouns

Method relies on pretrained guided diffusion model

- Same training hyper-parameters:
  - ImageNet model (pre-trained) 256 x 256, 250 time steps
  - CelebA-HQ model (trained) 256 x 256, 250 time steps
    - 3 batches @ 4 x V100 GPUs each
    - Trained for 250,000 iterations; 5 days
User Study

- User compares two inpainting solutions for realism.
- Answers submitted only if user is >=75% self-consistent, questions all asked twice.

100 test images:
- Wide, Narrow, Super-Resolve, Every Second Line, Half Image, and Expand
- 1000 votes (5 x 2 x 100) per method-to-method comparison in each dataset and mask, and show the 95% confidence interval next to the mean votes.

<table>
<thead>
<tr>
<th>Mask</th>
<th>LPIPS ( \downarrow )</th>
<th>Votes [%]</th>
<th>LPIPS ( \downarrow )</th>
<th>Votes [%]</th>
<th>LPIPS ( \downarrow )</th>
<th>Votes [%]</th>
<th>LPIPS ( \downarrow )</th>
<th>Votes [%]</th>
<th>LPIPS ( \downarrow )</th>
<th>Votes [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wide</td>
<td>0.104</td>
<td>11.6 ± 2.0</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
</tr>
<tr>
<td>Narrow</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
<td>0.604</td>
<td>8.3 ± 1.7</td>
</tr>
<tr>
<td>Super-Resolve</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
<td>0.604</td>
<td>8.3 ± 1.7</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
</tr>
<tr>
<td>Every Second</td>
<td>0.104</td>
<td>11.6 ± 2.0</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
</tr>
<tr>
<td>Line</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
<td>0.604</td>
<td>8.3 ± 1.7</td>
</tr>
<tr>
<td>Half Image</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
<td>0.604</td>
<td>8.3 ± 1.7</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
</tr>
<tr>
<td>Expand</td>
<td>0.104</td>
<td>11.6 ± 2.0</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
<td>0.667</td>
<td>2.4 ± 1.0</td>
<td>0.287</td>
<td>9.0 ± 1.8</td>
</tr>
</tbody>
</table>

RePaint  Reference

Votes is ratio of votes w.r.t. RePaint. 2.4 ± 1.0 means: 2.4% went to competitor with 1.0 margin of error.
User Study

Which image looks more realistic?

Task 14/20

Reference | Candidate 1 | Candidate 2

Answers submitted only if user is >=75% self-consistent.
All questions all asked twice

Figure 8. User Study Interface. Example of the user-study interface. Based on the reference image on the Left, the user selects the image that looks more realistic.
Perceptual Metric

- Learned distance metric based on the deep feature space of AlexNet
- Lower score is better

<table>
<thead>
<tr>
<th></th>
<th>Wide</th>
<th>Narrow</th>
<th>Super-Resolve 2×</th>
<th>Altern. Lines</th>
<th>Half</th>
<th>Expand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Votes (%)</td>
<td>LPIPS↓</td>
<td>Votes (%)</td>
<td>LPIPS↓</td>
<td>Votes (%)</td>
<td>LPIPS↓</td>
</tr>
<tr>
<td>LPiPS↓</td>
<td>0.104</td>
<td>11.6 ± 2.0</td>
<td>0.047</td>
<td>12.8 ± 2.1</td>
<td>0.714</td>
<td>1.1 ± 0.6</td>
</tr>
</tbody>
</table>
User opinion: RePaint outperforms all others except for ICT on Half mask in ImageNet

LPIPS: RePaint appears highly competitive, but according to the authors, LPIPS punishes RePaint on Expand and Half masks due to Repaint’s flexibility to generate a semantically different image than ground-truth.
ImageNet - Half Mask
ImageNet - Expand Mask
Qualitative
Qualitative: ImageNet

Input

DSI [33]

ICT [42]

LaMa [40]

RePaint (ours)
Qualitative: ImageNet
Qualitative: CelebA-HQ
Qualitative: CelebA-HQ
Limitations
Limitations

- RePaint’s per-image DDPM optimization process is significantly slower than GAN and autoregressive-based methods.
- Not good for real-time, but newer work is on increasing efficiency.
- Alternative is to use FID score for more fair quantitative analysis to baselines but computing FID for inpainting requires 1,000 images and the current DDPM is too slow, runtime is infeasible for most research institutes.
- RePaint can produce realistic images completions that are very different from the Ground Truth image.
- RePaint’s choices will reflect bias of underlying dataset

Notable use case:
- Face anonymization - replace faces with hallucinated faces
Demos

We show demos on two trained models (ImageNet and Celeb-A)

ImageNet demo

Image

Celeb-A demo

Image
Demo with ImageNet model
Demo with Celeb-A model
END