Patches and Attention for Image Editing

Séminaire IMAGE - GREYC

Nicolas Cherel[1], Yann Gousseau[1], Alasdair Newson[1], Andrés Almansa[2]

February 2nd 2023

[1]Télécom Paris, Institut Polytechnique de Paris

[2]MAP5, CNRS & Université de Paris Cité

Table of contents

A Patch-based Algorithm for Single Image Generation

Single Image Generation

"Generate diverse image samples, visually similar to a reference image but nonetheless different."

SinGAN's results [1]

[1] Shaham, Dekel, and Michaeli, "Singan: Learning a Generative Model from a Single Natural Image", 2019.

Challenges

Visual fidelity

- similar structure
- similar details

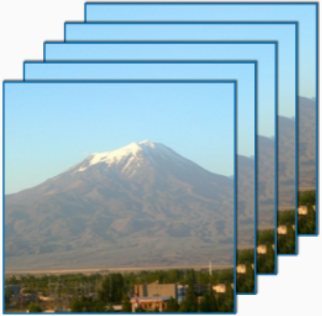
Challenges

Visual fidelity

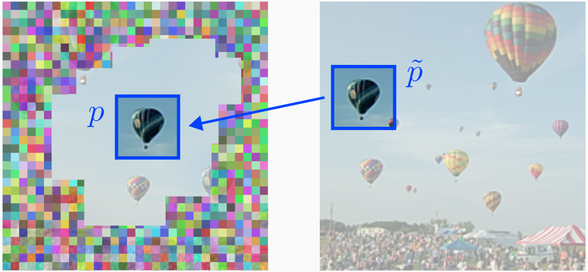
- similar structure
- similar details

Diversity

- varied samples



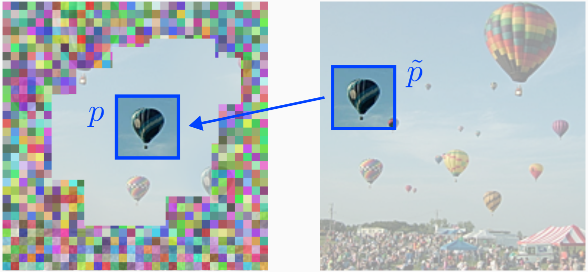
Patch-based algorithm



generated u reference $\tilde{u}$

[2] Kwatra et al., "Texture Optimization for Example-Based Synthesis", 2005.

Patch-based algorithm



generated u reference $\tilde{u}$

Minimize energy of Kwatra *et al.* [2]:

$$E(u) = \sum_{p \in u} \min_{\tilde{p} \in \tilde{u}} \|p - \tilde{p}\|_2^2$$

with patch $p, \tilde{p} \in \mathbb{R}^{11 \times 11 \times 3}$

[2] Kwatra et al., "Texture Optimization for Example-Based Synthesis", 2005.

Energy minimization

Nearest Neighbor (NN) mapping

$\phi : u \to \tilde{u}$

$$E(u, \phi) = \sum_{p \in u} \|p - \phi(p)\|_2^2$$

Alternate minimizations on u, ϕ

[3] Barnes et al., “PatchMatch”, 2009.

Energy minimization

Nearest Neighbor (NN) mapping

$\phi : u \to \tilde{u}$

$$E(u, \phi) = \sum_{p \in u} \|p - \phi(p)\|_2^2$$

Alternate minimizations on u, ϕ

optimization over ϕ - NN Search

$$\min_{\phi} \sum_{p \in u} \|p - \phi(p)\|_2^2 \tag{1}$$

Fast approximation with PatchMatch [3]

[3] Barnes et al., "PatchMatch", 2009.

Energy minimization

Nearest Neighbor (NN) mapping

$\phi : u \to \tilde{u}$

$$E(u, \phi) = \sum_{p \in u} \|p - \phi(p)\|_2^2$$

Alternate minimizations on u, ϕ

optimization over ϕ - NN Search

$$\min_{\phi} \sum_{p \in u} \|p - \phi(p)\|_2^2 \tag{1}$$

Fast approximation with PatchMatch [3]

optimization over u - Reconstruction

$$\min_{u} \sum_{p \in u} \|p - \phi(p)\|_2^2 \tag{2}$$

Least-squares problem

[3] Barnes et al., "PatchMatch", 2009.

Multiscale

Energy minimized at multiple scales

- Gaussian pyramid of factor 2^L
- coarse-to-fine synthesis

$$u_L \to u_{L-1} \to \dots \to u_0$$

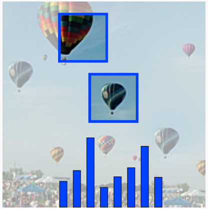
Initialization from noise

Reference

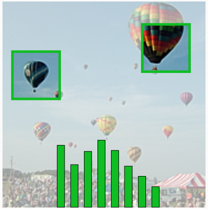
3 scales

4 scales

5 scales



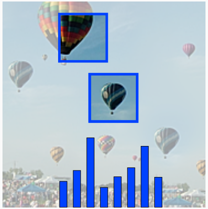
generated u



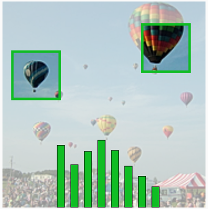
reference $\tilde{u}$

[4] Houdard et al., “Wasserstein Generative Models for Patch-Based Texture Synthesis”, 2021.

Optimal Transport



generated u



reference $\tilde{u}$

Minimize Wasserstein-2 distance between patch distributions of u and $\tilde{u}$ [4]

$$OT(u) = \max_{\beta} \sum_{p \in u} \min_{\tilde{p} \in \tilde{u}} \left(\|p - \tilde{p}\|_2^2 - \beta_{\tilde{p}} \right) + \sum_{\tilde{p} \in \tilde{u}} \beta_{\tilde{p}}$$

[4] Houdard et al., "Wasserstein Generative Models for Patch-Based Texture Synthesis", 2021.

Optimal transport energy minimization:

- computationally expensive steps
- multiscale

Strategy

1. First ℓ levels with Optimal Transport
2. Next $L - \ell$ levels with simple energy

Algorithms

PSin

$u \leftarrow \text{rand}()$
for $s = L, ..., 0$ **do**
  $u \leftarrow \text{rescale}(u, \text{scale} = s)$
  for $i = 1, ..., 10$ **do**
    $\phi \leftarrow \text{NN-Mapping}(u, \tilde{u})$
    $u \leftarrow \text{Reconstruction}(\phi, \tilde{u})$
  end for
end for

PSinOT

$u \leftarrow \text{OTSolver}(u, [L, ..., L - \ell])$
for $s = L - \ell, ..., 0$ **do**
  $u \leftarrow \text{rescale}(u, \text{scale} = s)$
  for $i = 1, ..., 10$ **do**
    $\phi \leftarrow \text{NN-Mapping}(u, \tilde{u})$
    $u \leftarrow \text{Reconstruction}(\phi, \tilde{u})$
  end for
end for

Results

Reference | SinGAN | PSin | PSinOT

Patch originality

Reference | SinGAN | PSinOT

Quantitative metrics

Fidelity: Single Image Fréchet Inception Distance (SIFID), Optimal Transport cost

Diversity: Average pixelwise standard deviation for N images generated

Algorithm	SIFID ↓	Optimal Transport ↓	Diversity ↑
SinGAN	*0.12*	1.34	0.34
PSin	0.45	*0.94*	**0.62**
PSinOT	**0.06**	**0.36**	*0.53*

Average metrics for 50 samples for images from Places50. **best**, *second best*.

Conclusion

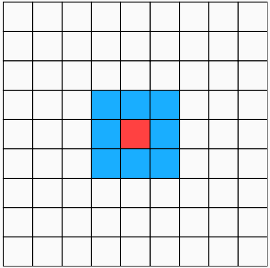
Patch-based algorithm for single image generation

+ no learning
+ good quality in seconds
+ choice between diversity and fidelity
– limited originality

Code: github.com/ncherel/psin

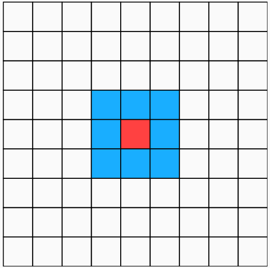
Patch-based Stochastic Attention

Local convolution

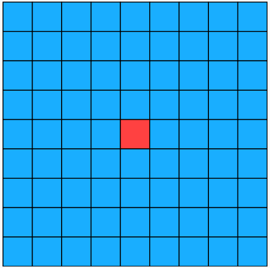


Non-local operations

Local convolution

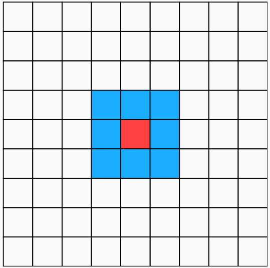


Non-local operation



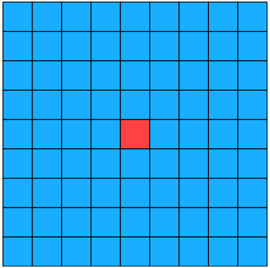
Non-local operations

Local convolution

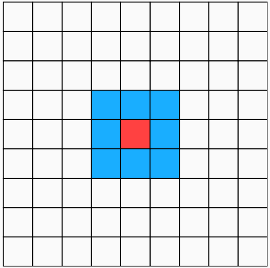


$$f(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w_{i,j} \cdot u_{x-i,y-j}$$

Non-local operation

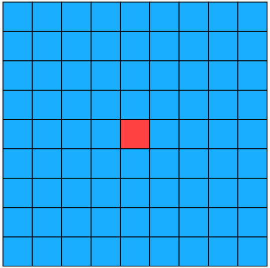


Local convolution



$$f(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w_{i,j} \cdot u_{x-i,y-j}$$

Non-local operation



$$f(x,y) = \sum_{x'} \sum_{y'} s(u_{x,y}, u_{x',y'}) \cdot u_{x',y'}$$

The Attention framework

Full Attention [5]

Queries $Q \in \mathbb{R}^{n \times d}$, keys $K \in \mathbb{R}^{n \times d}$, values $V \in \mathbb{R}^{n \times d'}$:

$$\forall i \in [1, n], \text{Attention}(q_i, K, V) = \frac{1}{C_i} \sum_{j=1}^{n} e^{\langle q_i, k_j \rangle} v_j$$

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V$$

[5] Vaswani et al., "Attention Is All You Need", 2017.

The Attention framework

Full Attention [5]
Queries $Q \in \mathbb{R}^{n \times d}$, keys $K \in \mathbb{R}^{n \times d}$, values $V \in \mathbb{R}^{n \times d'}$:

$$\forall i \in [1, n], \text{Attention}(q_i, K, V) = \frac{1}{C_i} \sum_{j=1}^{n} e^{\langle q_i, k_j \rangle} v_j$$

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V$$

Complexity for n elements (pixels, patches, ...)

- Computational complexity: $\mathcal{O}(n^2 d)$
- Memory complexity: $\mathcal{O}(n^2)$; $n = 256^2$ requires 16GB of RAM

[5] Vaswani et al., "Attention Is All You Need", 2017.

Efficient attention

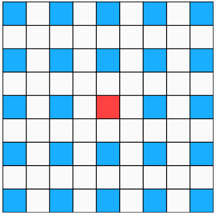
Linear approximation of softmax:

$$\text{softmax}(QK^T)V \approx \phi(Q)\psi(K)^T V$$

Linear Transformer [6], Performer [7]

Subsampling the key set K:

- strided pattern
- local neighborhood [8]



strided subsampling pattern

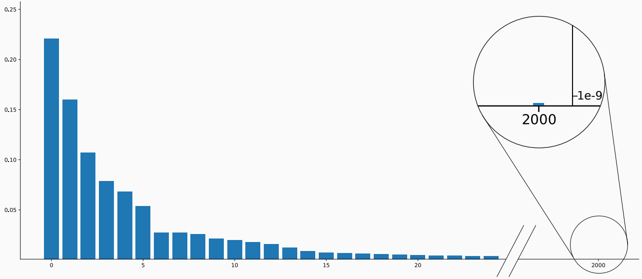
[6] Katharopoulos et al., "Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention", 2020.
[7] Choromanski et al., "Rethinking Attention with Performers", 2020.
[8] Parmar et al., "Image Transformer", 2018.

The Attention framework

Going back to the attention equation:

$$\forall i \in [1, n], \text{Attention}(q_i, K, V) = \frac{1}{C_i} \sum_{j=1}^{n} e^{\langle q_i, k_j \rangle} v_j \quad \text{where} \quad C_i = \sum_{j=1}^{n} e^{\langle q_i, k_j \rangle}$$

Finite and small amount of non-negligible weight terms



Decreasing weights in attention after normalization

Sparse attention

Sparse attention using the nearest neighbor

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V \approx AV$$

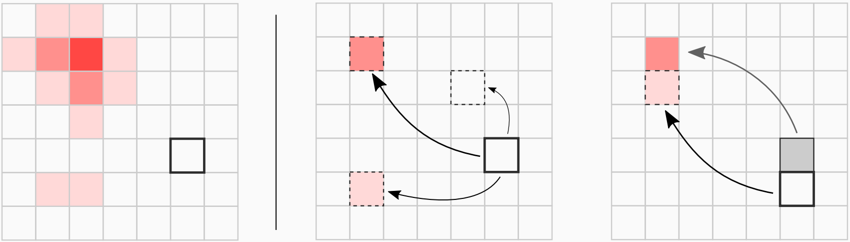
$$\text{where} \quad A_{i,j} = \begin{cases} 1 & \text{if} \quad \psi(i) = j \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \psi(i) = \underset{j \in \{1,\dots,n\}}{\arg\max} \langle q_i, k_j \rangle$$

Efficient algorithms for nearest neighbor search: KD-Trees, LSH [9], PatchMatch

[9] Kitaev, Kaiser, and Levskaya, "Reformer", 2020.

Patch-based Stochastic Attention Layer

Approximate ψ using parallel PatchMatch [10]



(a) (b) (c)

$\mathcal{O}(n)$ memory complexity and $\mathcal{O}(nd \log n)$ computational complexity vs $\mathcal{O}(n^2)$ and $\mathcal{O}(n^2 d)$ for Full Attention.

[10] Barnes et al., "PatchMatch", 2009.

Differentiability

PatchMatch is not differentiable with respect to all variables as a pseudo-argmax.

$$\text{Attention}(Q, K, V) = AV \quad \text{where} \quad A_{i,j} = \begin{cases} 1 & \text{if} \quad \psi(i) = j \\ 0 & \text{otherwise} \end{cases}$$

A depends on Q, K but not its entries. 2 solutions:

- K Nearest Neighbors (KNN)
- Neighbors aggregation

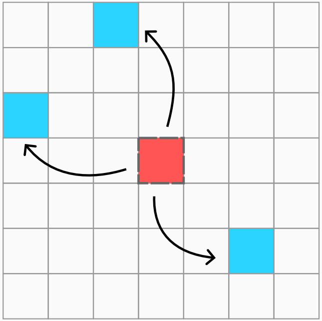
Differentiability with KNN

We consider the set of nearest neighbors of element $\psi(i)$ to construct the matrix of similarities S:

$$S_{i,j} = \begin{cases} \langle Q_i, K_j \rangle & \text{if } j \in \psi(i) \\ 0 & \text{otherwise.} \end{cases}$$

The matrix A is then obtained by normalization of the rows:

$$A = \text{softmax}(S)$$



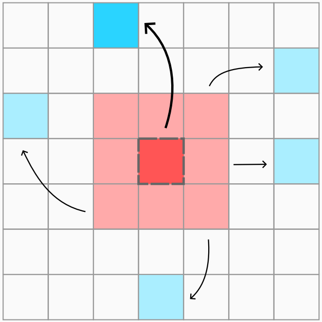
3 Nearest Neighbors

Differentiability with aggregation

We use the neighbors' neighbors. $\mathcal{N}_i$ is the set of spatial neighbors of i.

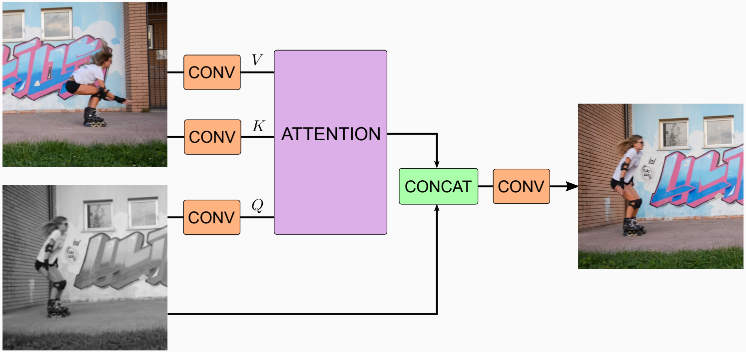
$$S_{i,j} = \begin{cases} \langle Q_{i'}, K_{j'} \rangle & \text{if } \begin{cases} i' \in \mathcal{N}_i \text{ and } j' \in \psi(i') \\ \text{and } i' - i = j' - j \end{cases} \\ 0 & \text{otherwise,} \end{cases}$$

The matrix S is then normalized along the rows.



Neighbors aggregation

Guided image colorization

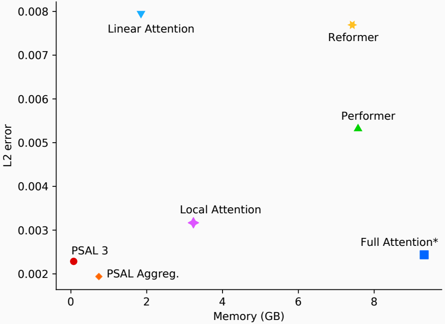


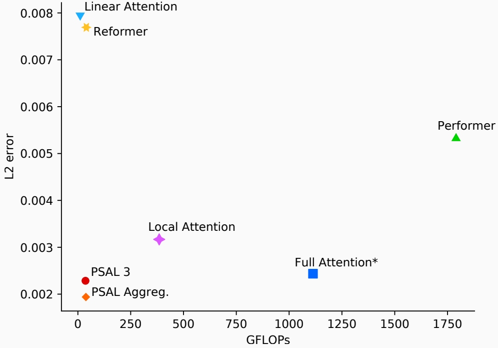
PSAL differentiability

Experiments confirm that PSAL with 1 neighbor is not differentiable end-to-end.

Attention Method	ℓ_2 loss
PSAL 1	0.0083
PSAL 3	0.0023
PSAL Aggreg.	0.0019

Colorization results





Performance vs computational constraints (memory and GFLOPs) on the colorization task

Inpainting task

Comparison with ContextualAttention[11], using PSAL:

- similar inpainting metrics
- high resolution inpainting

Attention	ℓ_1 loss $\downarrow$	ℓ_2 loss $\downarrow$	SSIM $\uparrow$
Contextual	11.8%	3.6%	53.7
PSAL (ours)	**11.6%**	**3.6%**	**54.1**

Average inpainting metrics on Places2 validation set.

2700x3300 image inpainted using PSAL

[11] Yu et al., "Generative Image Inpainting with Contextual Attention", 2018.

Conclusion

Patch-based attention

+ very low memory
+ scales to high resolution images and videos
− cannot approximate high entropy attention

Code: `github.com/ncherel/psal`

Full text: `https://arxiv.org/abs/2202.03163`

Current work

Diffusion

Diffusion is state-of-the-art for conditional and unconditional image generation:

- text-to-image
- super-resolution
- inpainting

[10] Ho, Jain, and Abbeel, "Denoising Diffusion Probabilistic Models", 2020.
[11] Rombach et al., "High-Resolution Image Synthesis With Latent Diffusion Models", 2022.

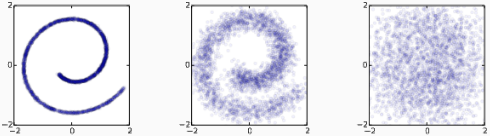
Diffusion : quick introduction

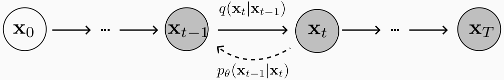
Modeling complex data distributions through:

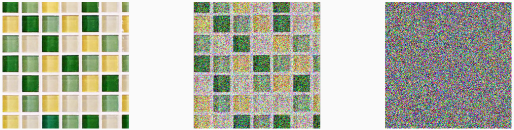
- forward process: $q(x_t \mid x_{t-1})$
- learned backward process $p_\theta(x_{t-1} \mid x_t)$

Training by denoising:

$$\mathcal{L}(\theta) = \mathbb{E}_{x,\epsilon}\left[\|x - f_\theta(x+\epsilon)\|^2\right]$$

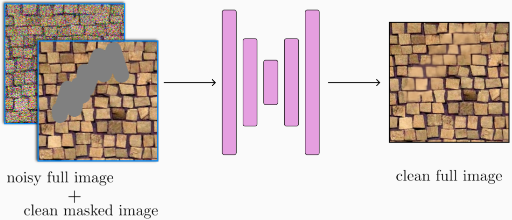






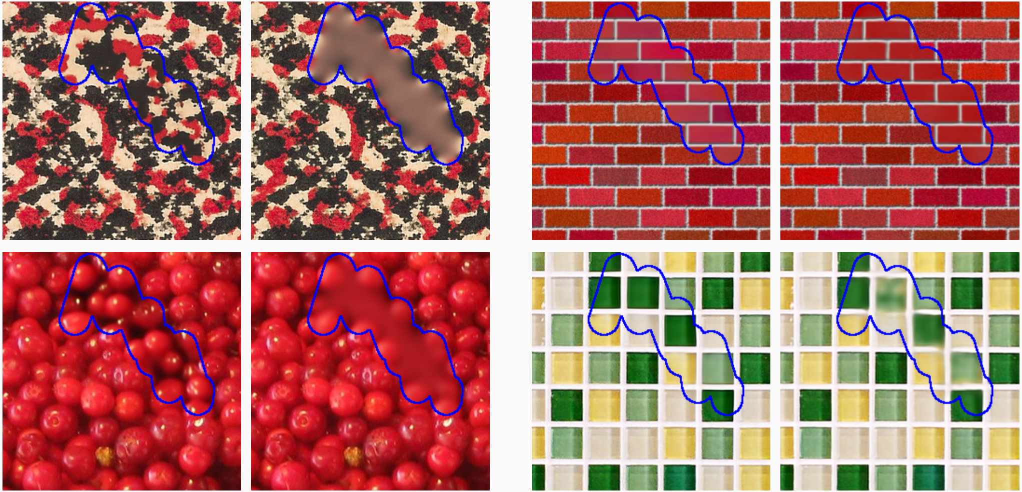
Current inpainting experiments

- Training on a single texture
- Tiny model: 160k parameters
- 20-min training



First results

Diffusion | Direct inpainting | Diffusion | Direct inpainting



Questions

References i

- Barnes, Connelly et al. “PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing”. In: *SIGGRAPH 2009*. 2009. DOI: 10.1145/1531326.1531330.
- Choromanski, Krzysztof et al. “Rethinking Attention with Performers”. In: *ArXiv* (2020).
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.
- Houdard, Antoine et al. “Wasserstein Generative Models for Patch-Based Texture Synthesis”. In: *Scale Space and Variational Methods in Computer Vision*. Vol. LNCS 12679. Cabourg, France, 2021, pp. 269–280.

References ii

- Katharopoulos, A. et al. "Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2020.
- Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya. "Reformer: The Efficient Transformer". In: *ICLR* (2020).
- Kwatra, Vivek et al. "Texture Optimization for Example-Based Synthesis". In: *ACM SIGGRAPH 2005 Papers*. 2005, pp. 795–802.
- Mei, Yiqun et al. "Image Super-Resolution With Cross-Scale Non-Local Attention and Exhaustive Self-Exemplars Mining". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 5689–5698. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00573.

Parmar, Niki et al. "Image Transformer". In: *ICML* (2018).

Rombach, Robin et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

Shaham, Tamar Rott, Tali Dekel, and Tomer Michaeli. "Singan: Learning a Generative Model from a Single Natural Image". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4570–4580.

Song, Yang and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

Vaswani, Ashish et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008.

Yu, Jiahui et al. "Generative Image Inpainting with Contextual Attention". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018).

PSAL for super-resolution

For single-image super-resolution, Cross-Scale attention [12] can be efficiently approximated with PSAL as indicated by similar PSNR scores on the Urban 100 dataset.

Attention Method	Zoom x2	Zoom x3	Zoom x4
Cross-Scale Attention	33.383	29.123	27.288
PSAL	33.375	29.112	27.184

[12] Mei et al., "Image Super-Resolution With Cross-Scale Non-Local Attention and Exhaustive Self-Exemplars Mining", June 2020.

Diffusion, denoising and score-matching

Score-matching [13] is about learning the score of the data distribution: $\nabla \log p$. For a data point x, and a gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma I)$:

$$y = x + \epsilon$$

Tweedie's formula says that the MMSE denoiser D verifies:

$$\nabla_y \log p(y) = \frac{1}{\sigma^2}(D(y) - y)$$

Through denoising, we have access to the (smoothed) log-likelihood / score.

---

[13] Song and Ermon, "Generative Modeling by Estimating Gradients of the Data Distribution", 2019.
[13] Rombach et al., "High-Resolution Image Synthesis With Latent Diffusion Models", 2022.

Diffusion - Additional Results