SYNTHETIC TRAINING OF DEEP IMAGE RESTORATION NETWORKS : PRINCIPLES AND APPLICATIONS

Imaging in Paris, Institut Henri Poincaré, Paris, France

Raphaël Achddou

October 7th,2025

A3SI, LIGM, ESIEE Paris

https://rachddou.github.io
raphael.achddou@esiee.fr

IMAGE DENOISING: FROM PRIORS TO DEEP LEARNING

Real images are corrupted by noise. In the simplest setting:

$$y = x + n,\ n \sim \mathrm{N}(0, \sigma^2), \text{with:} \tag{1}$$

- x: the true (unknown signal)
- y: the noisy observation

Goal of image denoising: find the best noiseless estimate $\hat{x}(y)$. *Which characterization?*

(a) Noiseless image x (b) Noisy image y (c) Denoised estimate $\hat{x}(y)$

Image denoising and the Mean Square Error

Quality criterion for denoising

The most used criterion for denoising is the *Mean Square Error (MSE)*:

$$\text{MSE} = \mathbb{E}_{x,y}\left(||\hat{x}(y) - x||^2\right) \tag{2}$$

Goal: find the denoiser $\hat{x}(y)$ that minimizes the MSE.
Optimal denoiser: the Minimum Mean Square Error (MMSE)

$$\hat{x}_{MMSE}(y) = \text{argmin}_{\hat{x}}\mathbb{E}_{x,y}\left(||\hat{x}(y) - x||^2\right) = \mathbb{E}(x|y) \tag{3}$$

$$= \int_{\mathbf{x}} x.p(x|y)dx \tag{4}$$

$$\tag{5}$$

An elegant formulation, but not easily tractable.

Bayesian formulation and MAP estimate

Another popular approach is to find an estimate that maximizes the posterior distribution $p(x|y)$ (MAP):

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \text{const} * \exp\left\{\frac{-||x-y||^2}{2\sigma^2}\right\} p(x) \tag{6}$$

Which gives the MAP estimate:

MAP estimate

$$\hat{x}_{MAP} = \text{argmax}_x p(x|y) = \text{argmax}_x \exp\left\{\frac{-||x-y||^2}{2\sigma^2}\right\} p(x) \tag{7}$$

$$= \text{argmin}_x \underbrace{\frac{||x-y||^2}{2\sigma^2}}_{\text{data fidelity}} - \underbrace{\log(p(x))}_{\text{prior}} \tag{8}$$

$\rightarrow$ widely used formulation for optimization-based methods.[Rudin et al., 1992, **?**, Yu and Sapiro, 2011]

What are those typical priors?

Image Prior or Regularizers

Main categories of image priors:

- Smoothness
- Sparsity
- Self-similarity

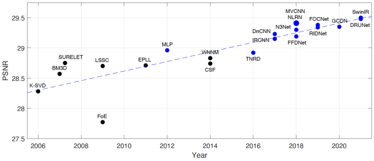
Table 3.1
Evolution of priors for images.

Years	Core concept	Formulae for $\rho(\cdot)$
$\sim$ 1970	Energy regularization	$\|\mathbf{x}\|_2^2$
1975–1985	Spatial smoothness	$\|\mathbf{Lx}\|_2^2$ or $\|\mathbf{D}_v\mathbf{x}\|_2^2 + \|\mathbf{D}_h\mathbf{x}\|_2^2$
1980–1985	Optimally learned transform	$\|\mathbf{Tx}\|_2^2 = \mathbf{x}^T\mathbf{R}^{-1}\mathbf{x}$ (via PCA)
1980–1990	Weighted smoothness	$\|\mathbf{Lx}\|_{\mathbf{W}}^2$
1990–2000	Robust statistics	$\mathbf{1}^T\mu\{\mathbf{Lx}\}$ *e.g.*, Hubber-Markov
1992–2005	TV	$\int_{v\in\Omega} \|\nabla\mathbf{x}(v)\| dv = \mathbf{1}^T\sqrt{\|\mathbf{D}_v\mathbf{x}\|^2 + \|\mathbf{D}_h\mathbf{x}\|^2}$
1987–2005	Other PDE-based options	$\int_{v\in\Omega} g\left[\nabla\mathbf{x}(v), \nabla^2\mathbf{x}(v)\right] dv$
2005–2009	Field-of-experts	$\sum_k \lambda_k \mathbf{1}^T\mu_k\{\mathbf{L}_k\mathbf{x}\}$
1993–2005	Wavelet sparsity	$\|\mathbf{Wx}\|_1$
2000–2010	Self-similarity	$\sum_k \sum_{j\in\Omega(k)} d\{\mathbf{R}_k\mathbf{x}, \mathbf{R}_j\mathbf{x}\}$
2002–2012	Sparsity methods	$\|\alpha\|_0$ $s.t.$ $\mathbf{x} = \mathbf{D}\alpha$
2010–2017	Low-rank assumption	$\sum_k \|\mathbf{X}_{\Omega(k)}\|_*$

Figure: List of mathematical priors given in [Elad et al., 2023].

The revolution of the deep learning era

Figure: Evolution of the performances in PSNR of denoising methods on BSD68



Requirements of deep learning

- a neural network $\hat{x}_{NN} = f_\theta(y)$
- a large dataset $p_{\text{data}} \rightarrow \mathrm{U}\left(\{x_i, y_i\}_{[1,\ldots,N]}\right)$
- an optimization framework as a variant of the SGD:

$$\hat{\theta} = \text{argmin}_{\theta \in \Theta} \mathbb{E}_{x,y \sim p_{\text{data}}} ||x - f_\theta(y)||_2^2$$

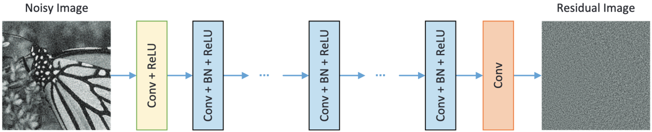


Figure: A denoising network (DnCNN) [Zhang et al., 2017].

Paradigm shift: No explicit formulation of the prior for deep denoising networks. f_θ : a mapping function.

$$\rho(x) \sim \mathrm{F}(\text{the dataset: } D \times \text{architecture: } \theta \times \text{loss: } \mathrm{L}) \tag{9}$$

Limitations of deep denoising networks

- hard to interpret
- unexplainable hallucinations
- limited generalization capacities (overfitting)
- limited performances when data is scarce

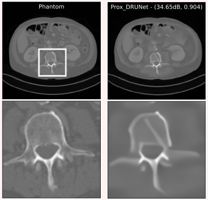


Figure: Hallucination effect in denoising [Goujon et al., 2024].

Proposed alternative: Train denoising networks on images of which we control all the properties → ***Synthetic Learning***

SYNTHETIC LEARNING: PRINCIPLES

Definition

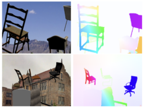
Synthetic learning refers to the process of training machine learning models on artificially generated data, rather than real-world data.

Benefits: data abundance, perfect ground truth, controlled data.

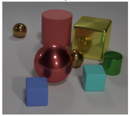
Challenges: domain gap, realism of the synthetic data.

LOW-LEVEL VISION

Optical flow estimation

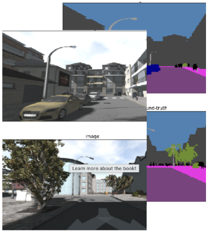


Monocular Depth estimation



HIGH-LEVEL VISION

Semantic Segmentation



Data Augmentation by Diffusion

For image restoration, what are the desired properties of these generators?

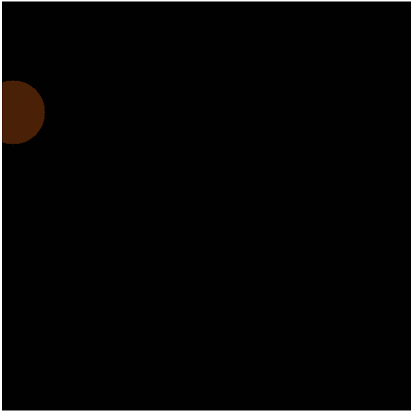
Classification of synthetic image generators

Table: Classification of Synthetic Image Generators

Properties / Category	No Semantic/ human bias	No Heavy engineering	Property disentanglement	Practicality	Diversity	Accuracy/ realism
3D rendering engines	✗	✗	✓	✗	$\sim$	$\sim$
Deep Generative Models	✗	✗	✗	$\sim$	$\sim$	✓
Parametric Procedural Models	✓	✓	✓	✓	$\sim$	???

Parametric Procedural Models: Stochastic image generators, based on random processes.
Examples: Gaussian Textures, Fractals, Reaction Diffusion, Perlin Noise, Occlusion model.
$\rightarrow$ **The Dead Leaves Image Model** [Matheron, 1968].

A random superimposition of shapes of random size, color, and positions.



Mathematical formulation

the dead leaves model

a random process $(x_i, t_i, X_i)_{i \in \mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma \delta_{x_i, t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Mathematical formulation

the dead leaves model

a random process $(x_i, t_i, X_i)_{i \in \mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma \delta_{x_i, t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Useful definitions:

the dead leaves model

a random process $(x_i, t_i, X_i)_{i \in \mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma \delta_{x_i, t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Useful definitions:

- **A leaf:** the set of positions $x_i + X_i$

the dead leaves model

a random process $(x_i, t_i, X_i)_{i \in \mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma\delta_{x_i,t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Useful definitions:

- **A leaf:** the set of positions $x_i + X_i$
- **The visible part:** the positions of a leaf which are not covered by previous leaves: $V_i = (x_i + X_i) \setminus \bigcup_{t_j \in (t_i, 0)} (x_j + X_j)$

the dead leaves model

a random process $(x_i, t_i, X_i)_{i \in \mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma\delta_{x_i,t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Useful definitions:

- **A leaf:** the set of positions $x_i + X_i$
- **The visible part:** the positions of a leaf which are not covered by previous leaves: $V_i = (x_i + X_i) \backslash \bigcup_{t_j \in (t_i, 0)} (x_j + X_j)$
- **dead leaves tesselation**: $T = \bigcup_i V_i$

Mathematical formulation

the dead leaves model

a random process $(x_i, t_i, X_i)_{i\in\mathbb{N}}$

- $x_i, t_i \sim \mathrm{P} = \Sigma\delta_{x_i,t_i}$, a Poisson point process on $\mathbb{R}^2 \times (-\infty, 0]$
- X_i random sets of $\mathbb{R}^2$; usually the set of disks of radius $r_i \sim p(r)$

Useful definitions:

- **A leaf:** the set of positions $x_i + X_i$
- **The visible part:** the positions of a leaf which are not covered by previous leaves: $V_i = (x_i + X_i) \setminus \bigcup_{t_j \in (t_i, 0)} (x_j + X_j)$
- **dead leaves tesselation**: $T = \bigcup_i V_i$
- **the dead leaves image:** the result of coloring the visible parts with $c_i \sim q(c)$

Examples I

- $r \sim const$
- $c \sim \mathrm{U}([0,1]^3)$

Properties

artificial colors, constant shape size
→ not very natural.

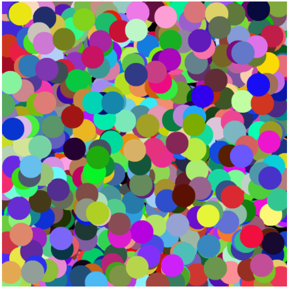


Figure: Process sample

Examples II

- $r \sim const$

Properties

colors sampled from a real image's histogram
$\rightarrow$ same color distribution
over-simplistic geometry.

- $c \sim$ color_histo(I), I: natural image

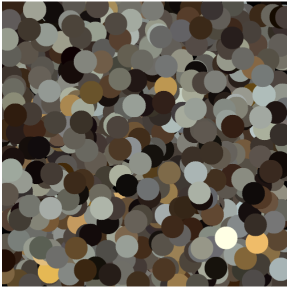


Figure: Process sample

Examples III

- $r \sim p(r) = C.r^{-\alpha}$, usually $\alpha = 3$

Properties

natural colors + and power law distribution of the radius.
Special case:
$\alpha = 3 \rightarrow$ scale invariance property.

- $c \sim$ color_histo(I), I: natural image

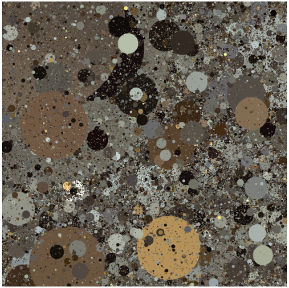


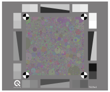
Figure: Process sample

Dead Leaves in Computer Vision

The Dead Leaves Model Chronology

1965
Invention of the Dead Leaves Image model by Georges Matheron for porous media

2008-2014
Dead leaves image model used for camera evaluation protocol as an SFR target [Cao et al., 2009, Artmann, 2015]



1995-2002
Statistical modeling of natural images scaling property

- [Alvarez et al., 1999]
- [Lee et al., 2001]

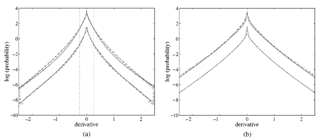
2021-today
Dead leaves images used for training deep neural networks.

- [Achddou et al., 2021, Achddou et al., 2023a, Achddou et al., 2023b]
- [Baradad et al., 2021, Baradad et al., 2022]
- [Madhusudana et al., 2021]

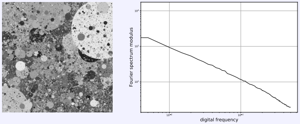
Advantages of the Dead Leaves Model

Advantages

- few parameters /good compromise between complexity and fidelity
- ”natural” statistical properties.
- direct control over contrast, colors.
- direct control over invariance properties:
  - scale invariance
  - rotation invariance
  - shift invariance
  - contrast invariance



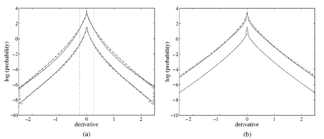
(a) Distribution of the gradient



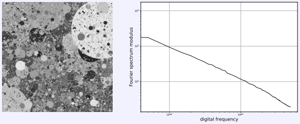
(b) Average 1-D power spectrum

Advantages

- few parameters /good compromise between complexity and fidelity
- ”natural” statistical properties.
- direct control over contrast, colors.
- direct control over invariance properties:
  - scale invariance
  - rotation invariance
  - shift invariance
  - contrast invariance



(a) Distribution of the gradient

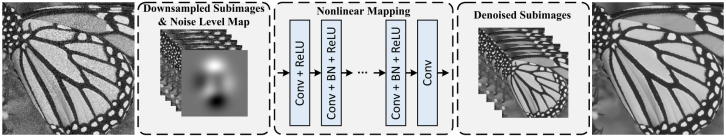


(b) Average 1-D power spectrum

Properties	No Semantic/ human bias	No Heavy engineering	Property Factorization	Practicality	Diversity	Accuracy/ realism
Dead Leaves Image Model	✓	✓	✓	✓	✓	∼

FFDNet: a lightweight image denoising CNN[Zhang et al., 2018].

Architecture.



Experimental Protocol, presented in [Achddou et al., 2021] *

- Generate 10k DL images of size $(500, 500, 3)$ / specific configuration as GT,
- Train FFDNet for color image denoising for each dataset
- Test the models on natural image benchmarks *(CBSD68, Kodak24, McMaster).*

*Synthetic Images as a Regularity Prior for Image Restoration Neural Networks, SSVM 2021

Database: 5000 images de from the Waterloo Exploration Database.

PSNR: 31.54 dB

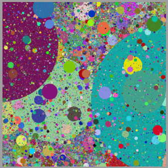
Database: a non gaussian random process of dead leaves images with a fixed radius.

PSNR: 30.1 dB (-1.4)

Training with dead leaves images with random radii

Database:

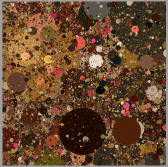
- DL images with scaling properties (power law of radii with $\alpha = 3$,$r_{\min} = 1$, $r_{\max} = 2000$)
- Colors uniformly drawn in the RGB cube.



PSNR: 29.6 dB (-1.8)

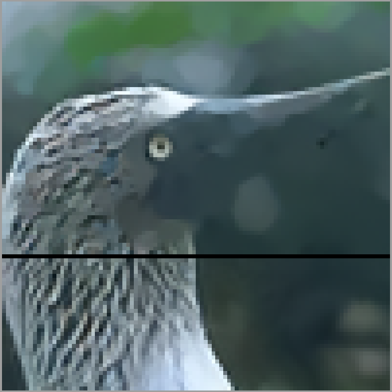
Training with DL images with natural colors

- DL images with scaling properties (power law of radii with $\alpha = 3$, $r_{\text{min}} = 1$, $r_{\text{max}} = 2000$)
- Colors drawn from natural images histograms



PSNR: 30.61 dB (-0.9)

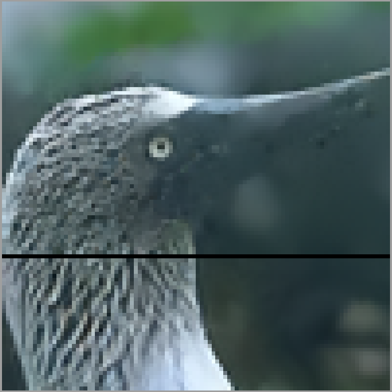
Training with DL images $r_{min} = 16$



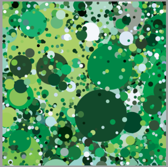
- DL images with scaling properties (power law of radii with $\alpha = 3, r_{\min} = 16$, $r_{\max} = 2000$)
- Colors drawn from natural images histograms

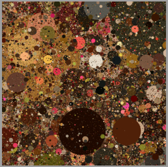
PSNR: 30.55 dB (-1)

Training with a mix of DL images with $r_{min} \in \{1, 16\}$



Database: A mix of DL images with $\alpha = 3, r_{min} \in \{1, 16\}, r_{max} = 2000.$





PSNR: 30.94 dB (-0.6)

Take-away Messages

Take-away Messages

→ Better insights on the crucial properties for training image restoration CNNs:

- **Non-gaussianity** of the image model (occlusions/Clear edges)
- **Scale invariance** property
- **Color distribution** close to natural images
- **Diversity** of the training database

→ Image restoration performances close to training with natural images.
→ Synthetic training is a good alternative that guarantees no semantic bias

BUT: some unanswered questions remain:

- *Can this model be used for other tasks than denoising?*
- *Can we avoid using natural images for the color distribution?*

Extensions to harder imaging problems

→ Exploration in an extended work [Achddou et al., 2023a] *

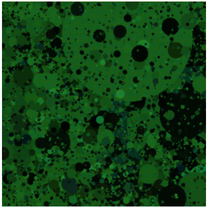
- Single-Image Super-Resolution
- Smartphone RAW Image Denoising
- **Low-Light RAW Image Enhancement**

*Fully synthetic training for image restoration tasks, CVIU 2023

→ Exploration in an extended work [Achddou et al., 2023a] *

- Single-Image Super-Resolution
- Smartphone RAW Image Denoising
- **Low-Light RAW Image Enhancement**

Database: RAW Dead Leaves Images



Distortion Model: [Wei et al., 2021]

$$Y = \frac{X}{\gamma} + N_s\left(\frac{X}{\gamma}\right) + N_r + N_b + N_q,$$

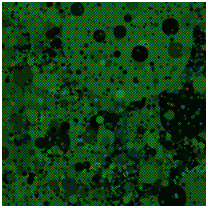
A new model based on Diffusion Models [Lu et al., 2025]

*Fully synthetic training for image restoration tasks, CVIU 2023

→ Exploration in an extended work [Achddou et al., 2023a] *

- Single-Image Super-Resolution
- Smartphone RAW Image Denoising
- **Low-Light RAW Image Enhancement**

Database: RAW Dead Leaves Images

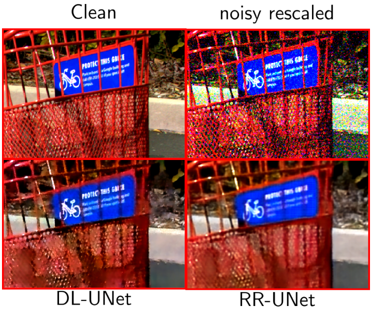


Distortion Model: [Wei et al., 2021]

$$Y = \frac{X}{\gamma} + N_s\left(\frac{X}{\gamma}\right) + N_r + N_b + N_q,$$

A new model based on Diffusion Models [Lu et al., 2025]

Low-Light Enhancement Results: Results of training a U-Net model [Chen et al., 2018]



*Fully synthetic training for image restoration tasks, CVIU 2023

Data agnostic model: a parametric color model

Objective: Develop a sampling algorithm for plausible natural color histograms with a parametric model [Achddou et al., 2023b]*

*Learning Raw Image Denoising Using a Parametric Color Image Model, ICIP 2023

Data agnostic model: a parametric color model

Objective: Develop a sampling algorithm for plausible natural color histograms with a parametric model [Achddou et al., 2023b]*

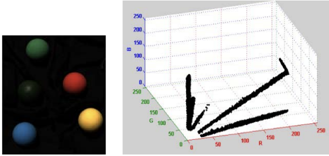


Figure: Lambertian objects distributed along a straight line in the RAW-RGB cube

Consequent idea: Factorize chrominance and luminance for a single object

$$p(x,y,z) = \underbrace{p(z|x,y)}_{\text{luminance}} \underbrace{p(x,y)}_{\text{chrominance}}$$

*Learning Raw Image Denoising Using a Parametric Color Image Model, ICIP 2023

Objective: Develop a sampling algorithm for plausible natural color histograms with a parametric model [Achddou et al., 2023b]*

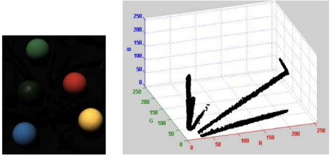
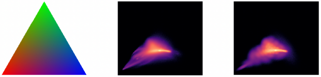


Figure: Lambertian objects distributed along a straight line in the RAW-RGB cube

Consequent idea: Factorize chrominance and luminance for a single object

$$p(x,y,z) = \underbrace{p(z|x,y)}_{\text{luminance}} \; \underbrace{p(x,y)}_{\text{chrominance}}$$

Chrominance model: Mixture of Gaussians fitted on a large dataset of RAW images, in the chromaticity plane (x, y)



(a) Triangle of the possible directions

(b) Log-histogram of $\mathcal{P}$ in the 2D plan

(c) Log-density of the GMM fitted on $\mathcal{P}$

Luminance model: A Gamma distribution conditioned on the chrominance

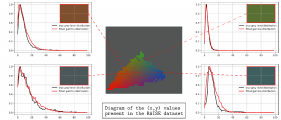


Fig. 3: Distribution of the average grey level knowing the position (x, y) in the 2D color representation.

*Learning Raw Image Denoising Using a Parametric Color Image Model, ICIP 2023

Limitations

Disadvantages

- oversimplistic geometry
- texture only arise from very small leaves / no repetitve textures
- completely flat areas
- no depth modeling except for occlusions

$\Longrightarrow$ **sub-optimal performances on Deep Learning tasks**

Figure: Comparison of denoising results: natural vs Dead leaves training of DRUNet

THE VIBRANT LEAVES MODEL

Dead Leaves vs Vibrant Leaves [Achddou et al., 2025]*

Differences between the two models

Properties	Natural Colors	Scaling properties	Depth modelling	Complex Geometry	Repetitive Textures
Dead Leaves model	✓	✓	∼	✗	✗
VibrantLeaves model	✓	✓	✓	✓	✓

More precisely, we propose the following additions:

- A random **free-form** generator
- A **texture model** for repetitive and random textures
- A **depth-of-field** simulator

*VibrantLeaves: A principled parametric image generator for training deep restoration models

Dead Leaves vs Vibrant Leaves [Achddou et al., 2025]*

Differences between the two models

Properties	Natural Colors	Scaling properties	Depth modelling	Complex Geometry	Repetitive Textures
Dead Leaves model	✓	✓	∼	✗	✗
VibrantLeaves model	✓	✓	✓	✓	✓

More precisely, we propose the following additions:

- A random **free-form** generator
- A **texture model** for repetitive and random textures
- A **depth-of-field** simulator

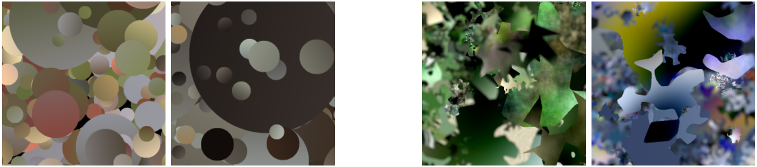


Figure: Dead Leaves vs Vibrant Leaves

*VibrantLeaves: A principled parametric image generator for training deep restoration models

Geometry: a random free-form generator

VL geometry

A random shape generator based on α-*shapes*.
α-shapes: a generalization of the convex hull, controlled by a parameter α. Special-case: $\alpha = 1$ leads to the convex hull of the points.

- *Base model*: the α-shapes of a set of points uniformly sampled in a disk
- *Curved model*: a smoothening/thresholding of the alpha-shape

VL geometry

A random shape generator based on α-*shapes*.
α-**shapes**: a generalization of the convex hull, controlled by a parameter α. **Special-case:** $\alpha = 1$ leads to the convex hull of the points.

- *Base model*: the α-shapes of a set of points uniformly sampled in a disk
- *Curved model*: a smoothening/thresholding of the alpha-shape

$x_i \sim U(D_1)$ $\alpha = 0.2$ $\alpha = 0.4$ $\alpha = 0.6$

(a) Base: α-shapes of a set of points

(b) Smoothening operation

VL geometry

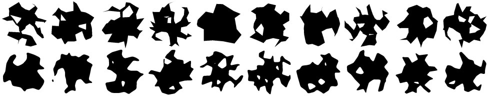
A random shape generator based on α-*shapes*.
α-**shapes**: a generalization of the convex hull, controlled by a parameter α. **Special-case:** $\alpha = 1$ leads to the convex hull of the points.

- *Base model*: the α-shapes of a set of points uniformly sampled in a disk
- *Curved model*: a smoothening/thresholding of the alpha-shape

$x_i \sim U(D_1)$ $\alpha = 0.2$ $\alpha = 0.4$ $\alpha = 0.6$

(a) Base: α-shapes of a set of points

(b) Smoothening operation



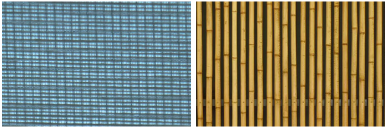
What is a texture?

No clear definition of a texture: a pattern that repeats itself with slight modifications at various scales. $\sim$ 2 types of textures

- pseudo-periodic textures
- micro-textures

DL model : only a few types of micro-textures, caused by the overlapping of small disks.

(a) Micro-textures



(b) Pseudo-periodic textures

Textures modeling

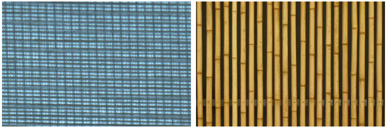
What is a texture?

No clear definition of a texture: a pattern that repeats itself with slight modifications at various scales. $\sim$ 2 types of textures

- pseudo-periodic textures
- micro-textures

DL model : only a few types of micro-textures, caused by the overlapping of small disks.

(a) Micro-textures



(b) Pseudo-periodic textures

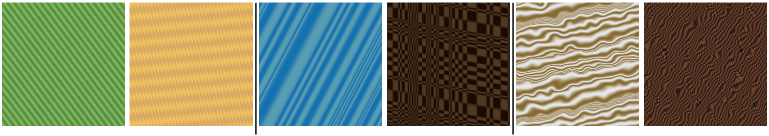
Goal: Propose an **exemplar-free texture generator**, based on principles of *randomness* and *repetitions.*

Pseudo-Periodic Pattern Generator

A model that creates pseudo-periodic interpolation maps T between two colors, with increasing complexity.

1. $T_1(x, \omega) = \sin(\omega x)$ a 1 or 2D sinusoidal map of random period,
2. $T_2(x, \omega) = \text{sigmoid}(\sin(\omega x), \alpha)$, sharper transitions with a logit funtion
3. $T_3(x) = \text{stack}(\{T_2(\omega_i)\}_{i<n})(x)$ a random oscillating field, obtained by stacking oscillations
4. $T_4(x) = \text{athmospheric\_distortion}(T_3(x))$, a displacement map obtained by filtered noise maps.

T_2 : Sinusoidal textures	T_3 : Random periods	T_4 : Warped textures



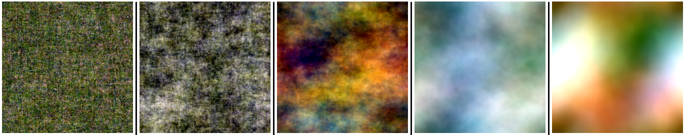
Common prior of natural images: $|\mathrm{F}(x,\nu)| \simeq \frac{C}{\nu^{\alpha}}$ with $\alpha \in [2-\epsilon, 2+\epsilon]$.

Micro-texture generator

Inspired by the phase randomization texture model of [Galerne et al., 2010]:

1. Generate a **white noise sample** with a natural color histogram:
2. Fix the power spectrum to a power function (isotropic)
3. Reconstruct the image by inverse Fourier transform

$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 1.5$	$\alpha = 2.0$	$\alpha = 2.5$



Modeling depth and the acquisition process

Depth in natural images

- **Perspective:** a non-linear mapping of 3D to 2D $\rightarrow$ vanishing points and parralel ines
- **Depth-of-field:** local, non-uniform blur based on object depth
- **Occlusions:** Objects occlude each other in the scene

Depth in the DL model

- only occlusions
- limitted model for physical depth

VL additions: A depth-of-filed simulator and a perspective model for texture maps

Depth-of-field approximation: a tri-plane division of space: blurred background / focused middle ground / blurred foreground

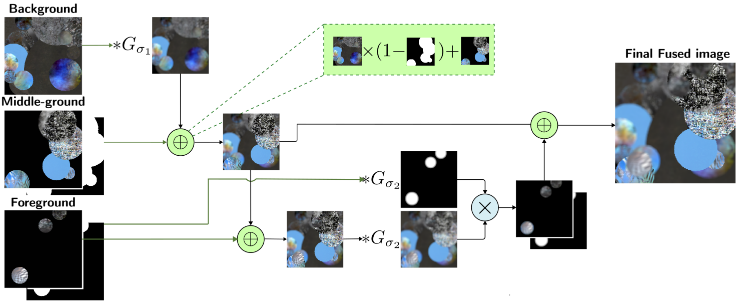


Figure: Diagram of the depth-of-field algorithm. After generating three DL stack (background, middle-ground and foreground), we fuse them by applying blur kernels $G_{\sigma_1}, G_{\sigma_2}$ respectively to the background and foreground.

Figure: Examples of samples from the VibrantLeaves model, which integrates modeling for *geometry, textures, and depth*.

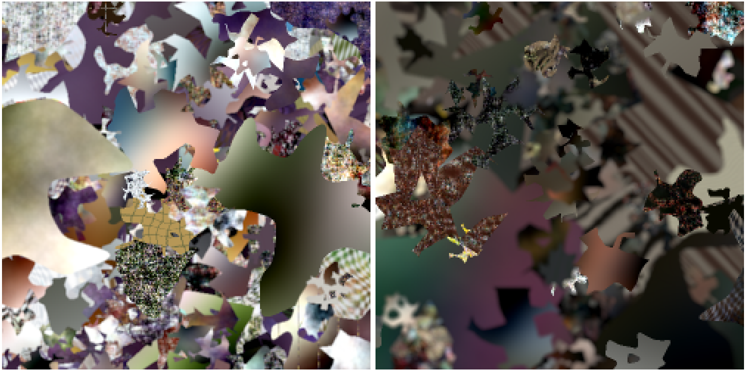


Figure: Examples of samples from the VibrantLeaves model, which integrates modeling for *geometry, textures, and depth.*

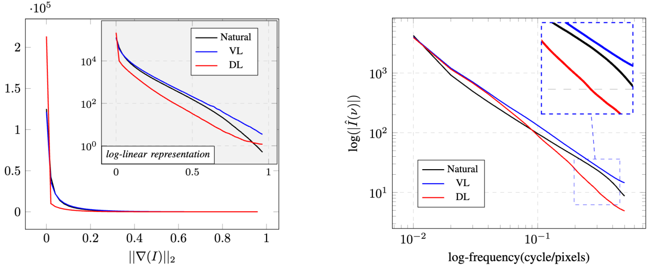


Figure: secon-order statistics comparison of DL (red), VL(blue) and Natural Images (black). *(Left).* Histograms of the image gradient $||\nabla(I)||_2$ estimated on 1000 patches of size (500×500) randomly drawn from each datasets. The grey plot represents the same quantities in a log-linear representation, to show the behavior better for higher gradient values. *(Right).* Average 1D power spectrum $(|\hat{I}(\nu)|)$ in a log-log representation for each datasets. To obtain a 1D representation, we average the 2D power spectrum radially.

Metric	*FID* ↓	*KL-Gradient*↓	α_{Spectrum} (R^2) ($_{Nat} = 1.43$)
DL[Achddou et al., 2021]	318	0.286	1.73 *(0.992)*
CleVR[Johnson et al., 2017]	217	0.517	1.67 *(0.992)*
GTA-5[Richter et al., 2016]	**186**	<u>0.015</u>	<u>1.49</u> *(0.982)*
FractalDB[Kataoka et al., 2020]	342	1.91	0.51 *(0.584)*
DL-textured[Baradad et al., 2021]	312	0.228	0.99 *(0.98)*
VL	<u>193</u>	**0.006**	**1.41** *(0.995)*

Table: Comparison of image ”naturalness” metrics for different synthetic datasets. We report the FID, as well as the KL of the gradient's distribution computed with respect to the natural images from WaterlooDB. We also report the slope α of the power spectrum, as well as the R^2 score of the linear regression. Overall, VL has better metrics than the other synthetic image datasets.

Protocol:

- generate 10K images for every configuration
- train a DRUNet denoiser on these individual datasets
- test on natural image datasets

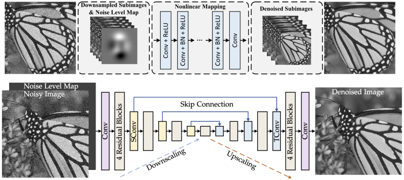
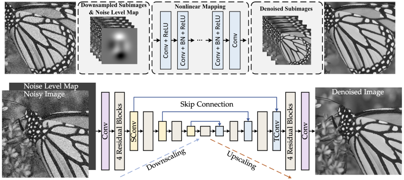


Image denoising results

Protocol:

- generate 10K images for every configuration
- train a DRUNet denoiser on these individual datasets
- test on natural image datasets



Test-set	σ	DRUNet			FFDNet		
		Nat. Images	VibrantLeaves	DeadLeaves	Nat. Images	VibrantLeaves	DeadLeaves
Kodak 24	25	**32.89**	32.16	30.95	32.13	31.72	30.91
	50	**29.86**	29.14	28.09	28.98	28.61	28.02
CBSD68	25	**31.69**	31.21	30.20	31.21	30.85	30.23
	50	**28.51**	28.06	27.18	27.96	27.68	27.19
McMaster	25	**33.14**	32.62	31.25	32.35	31.85	31.10
	50	**30.08**	29.56	28.32	29.18	28.78	28.18

Table: Image denoising results of FFDNet and DRUNet trained on either NatImages, VibrantLeaves or Dead leaves Best results are in **bold** and secon results are underlined.

Image denoising examples

Figure: Denoising visual results. We compare the same DRUNet architecture trained either on Dead Leaves, Vibrant Leaves or Nat images.

Image denoising examples

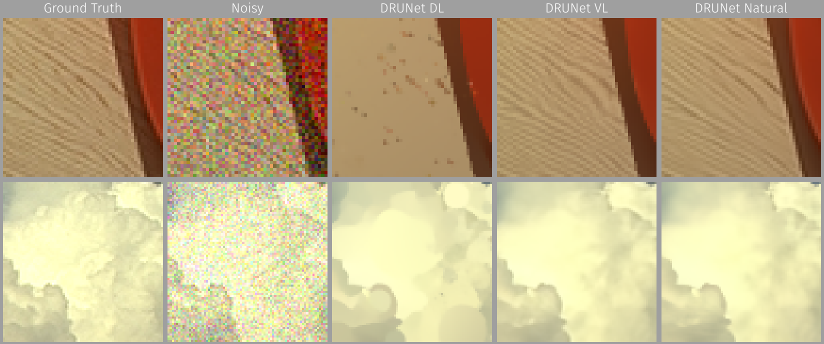


Figure: Denoising visual results. We compare the same DRUNet architecture trained either on Dead Leaves, Vibrant Leaves or Nat images.

Protocol:

- generate 10K images for every configuration
- train a lightweight SWINIR for SR at scale 2 and 4
- test on natural image datasets

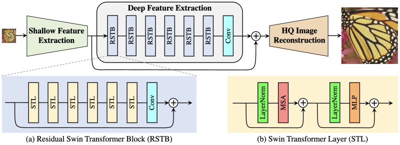


Figure: SWINIR architecture

Protocol:

- generate 10K images for every configuration
- train a lightweight SWINIR for SR at scale 2 and 4
- test on natural image datasets

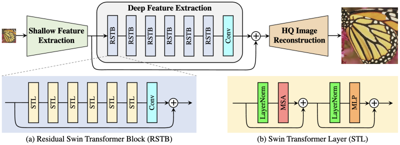


Figure: SWINIR architecture

Test-set	factor	SWIN-IR		
		Natural Images	VIbrantLeaves	DeadLeaves
Set5	2	**38.14**	37.39	35.92
	4	**32.44**	31.76	30.60
Set14	2	**33.86**	33.29	32.03
	4	**28.77**	28.49	27.76
DIV2K	2	**36.46**	35.48	34.19
	4	**30.65**	30.08	29.31

Table: Single-Image Super-Resolution results. The models are tested on several SISR benchmarks. Best results are in **bold** and secon results are underlined.

SISR examples

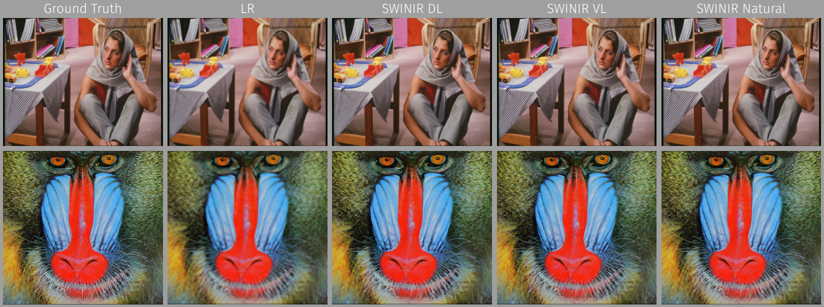


Figure: SISR visual results. We compare the same SWIN-IR architecture trained either on Dead Leaves, Vibrant Leaves or Nat images.

SISR examples

Figure: SISR visual results. We compare the same SWIN-IR architecture trained either on Dead Leaves, Vibrant Leaves or Nat images.

ADVANTAGES OF SYNTHETIC LEARNING

What are the advantages?

Questions

- Does the learned denoiser inherit invariance properties?
- Does the simplicity of the images lead to faster convergence of the training algorithm?
- Can we isolate image properties that are crucial for image restoration NNs?
- Does the network understand the prior encoded in the VL images?
- Can we use the trained denoiser as a prior in PnP?

Invariances Properties

Dead leaves images are supposed to have many invariances:

- Rotation
- scale
- Contrast
- shift

Is our learnt denoiser invariant to these? *or better than the natural baseline...*

Dead leaves images are supposed to have many invariances:

- Rotation
- scale
- Contrast
- shift

Is our learnt denoiser invariant to these? *or better than the natural baseline...*

Testing protocol

- find variances σ_1, σ_2 so that performances match
- test both models while varying the distortion.
- report the PSNR gap

Rotation Invariance

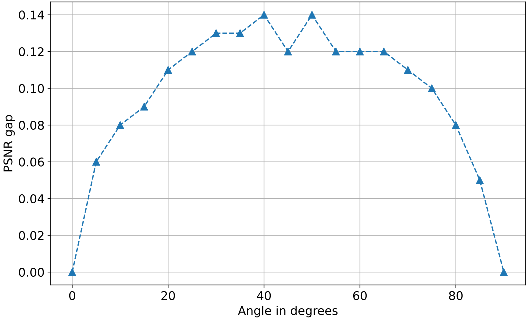


Figure: Performance gap for rotation invariance

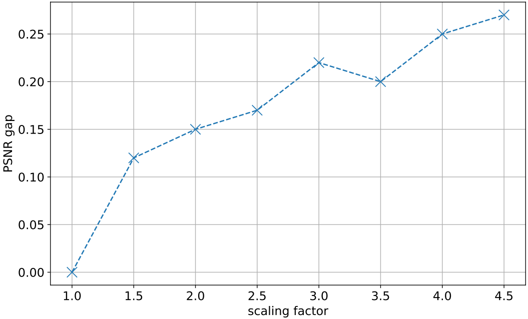


Figure: Performance gap for scale invariance

Convergence of the Optimization algorithm

Vibrant Leaves images exhibit natural properties, but they all share similar features.
$\rightarrow$ Not as diverse as natural images.
Intuitively, the data distribution has a **smaller support**, and more **data points.**
Question: *Is convergence to a good solution faster when training on Vibrant Leaves images?*

Optimization landscape

Convergence of the Optimization algorithm

Vibrant Leaves images exhibit natural properties, but they all share similar features.
$\rightarrow$ Not as diverse as natural images.
Intuitively, the data distribution has a **smaller support**, and more **data points.**
Question: *Is convergence to a good solution faster when training on Vibrant Leaves images?*

Can we measure the dimensionality of the training set?

Optimization landscape

Convergence of the Optimization algorithm

Vibrant Leaves images exhibit natural properties, but they all share similar features.
$\rightarrow$ Not as diverse as natural images.
Intuitively, the data distribution has a **smaller support**, and more **data points.**
Question: *Is convergence to a good solution faster when training on Vibrant Leaves images?*

Can we measure the dimensionality of the training set?

Somehow.

- For each dataset $(x_{i,d})_{i \leq N}$, project the images on a lower dimension space, using the Inception features used for FID $y_{i,d} = FC_{\text{Inception}}(x_{i,d})$.
- Compute the covariance matrix of the features $\Sigma_d = \frac{1}{N} \sum_i (y_{i,d} - \bar{y}_d)(y_{i,d} - \bar{y}_d)^T$
- Compute the eigenvalues of Σ_d, observe their profiles.

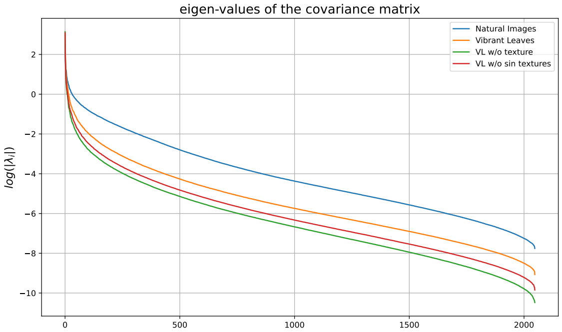
The larger the eigenvalues, the more diverse the dataset, the higher its dimensionality.

Optimization landscape

Convergence of the Optimization algorithm

Vibrant Leaves images exhibit natural properties, but they all share similar features.
$\rightarrow$ Not as diverse as natural images.
Intuitively, the data distribution has a **smaller support**, and more **data points.**
Question: *Is convergence to a good solution faster when training on Vibrant Leaves images?*

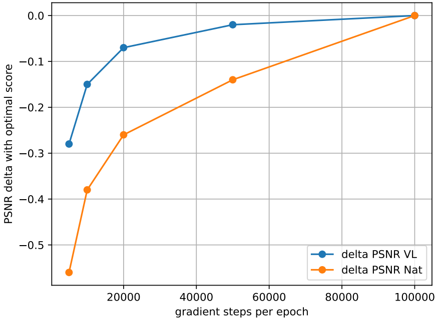
Can we measure the dimensionality of the training set?



Training Algorithm:

- Initial conditions: $\eta = 1.10^{-4}$, batch size 16, Adam optimizer.
- Perform K gradient steps, by forming mini-batches randomly on the training set.
- Divide the learning rate by 2, Repeat until $\eta = 5.10^{-7}$

Experiments: Progressively reduce K from 100K to 5K, observe the performance.



We test each model at $\sigma = 25$ and report the PSNR delta with respect to the model trained with $K = 100K$.
The optimization converges faster with VL images.

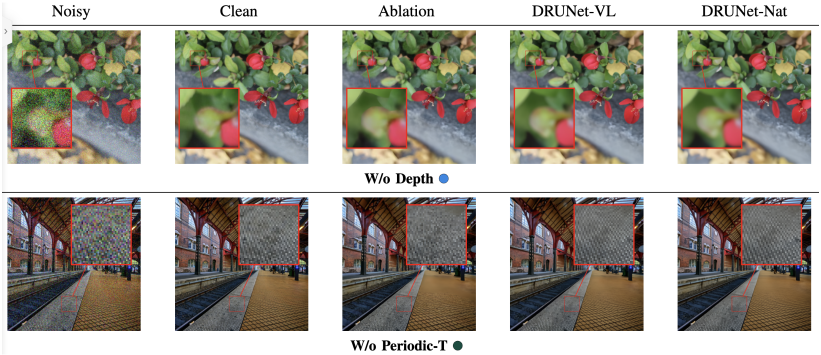
VibrantLeaves Ablation Study

Can we identify which properties of the VL model are responsible for the improvements?

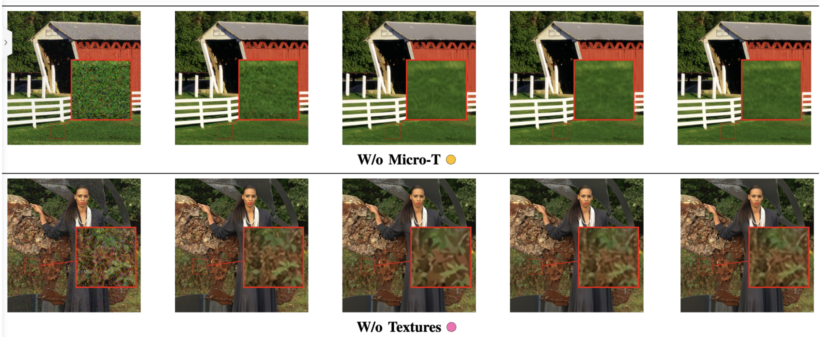
Protocol: We remove one property at a time → train DRUNet on the modified images → test on natural images

- **Vibrant Leaves**: all properties combined.
- **Without Depth**: W/o depth-of-field modeling and perspective.
- **Without micro-textures** : semi-periodic texture generator only.
- **Without periodic textures**: micro-textures only.
- **Without textures**

Ablation Study: Results



Ablation Study: Results



Ablation Study: Results

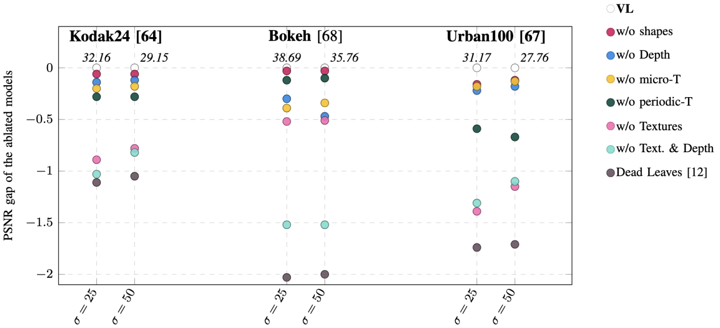


Fig. 16: Ablation study - Numerical results. We report the PSNR gap of the ablated models with respect to the results of DRUNet trained on VibrantLeaves. We test each models on two different noise values ($\sigma \in \{25, 50\}$) and different natural image datasets with different properties: Kodak24 [64], BokehDB [68], Urban100 [67]. The score on top of each column refers to the PSNR of DRUNet trained on VibrantLeaves.

Goal: Visualize the prior learned by the denoising network.
Methods:

- Feature visualization by activation maximization / Deep Dream → doesn't work well for denoisers
- Sampling the implicit prior using Langevin sampling algorithms [Kadkhodaie and Simoncelli, 2021, Leclaire et al., 2025]

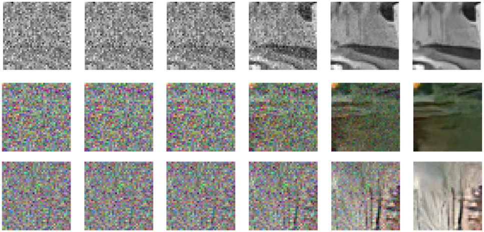


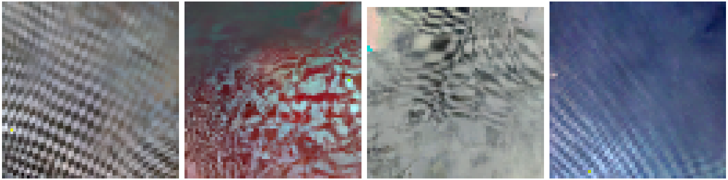
Figure: Images sampled from $\tilde{p}(x)$ from a blind denoising network trained on natural images

→ Somewhat natural patterns, but not clear what is learned.

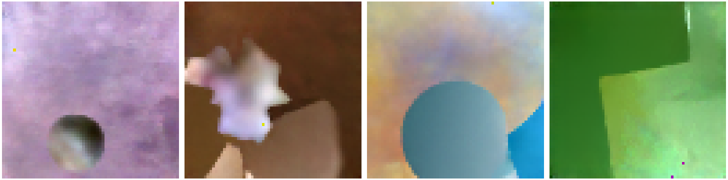
Better insights on the functionning of NNs: Prior Sampling

Does the prior implicitly learnt by the network incorporate all the properties we added to the dead leaves model?

- ☐ occlusions of the dead leaves model,
- ☐ shapes,
- ☐ textures,
- ☐ depth



(a) Pseudo-Periodic textures



(b) Occlusions and complex shapes

(a) Gaussian Micro-textures (colored noise)

(b) Bi-level textures

Learnt properties from Vibrant Leaves images:

- ✓ occlusions of the dead leaves model,
- ✓ shapes,
- ✓ textures,
- ✓ depth

Better insights on the functionning of NNs: Prior Sampling

Learnt properties from Vibrant Leaves images:

- ✓ occlusions of the dead leaves model,
- ✓ shapes,
- ✓ textures,
- ✓ depth

Can we use this prior in the context of inverse problems?

Plug-and-Play applications

Plug-and-Play

- Inverse problems formulation: $x^\star = \arg\min_x ||y - \phi(x)||_2^2 + \lambda\rho(x)$.
- Proximal splitting algorithms: alternate descent between $f(x) = ||y - \phi(x)||_2^2$ and $\rho(x) \sim log(p(x))$.
- **Plug-and-Play** [Venkatakrishnan et al., 2013]: replace the proximal operator of ρ by a denoiser D.

Spectacular results using deep denoisers [Zhang et al., 2021], and many theoretical results [Laumont et al., 2022, Hurault et al., 2022, Goujon et al., 2024] etc ...

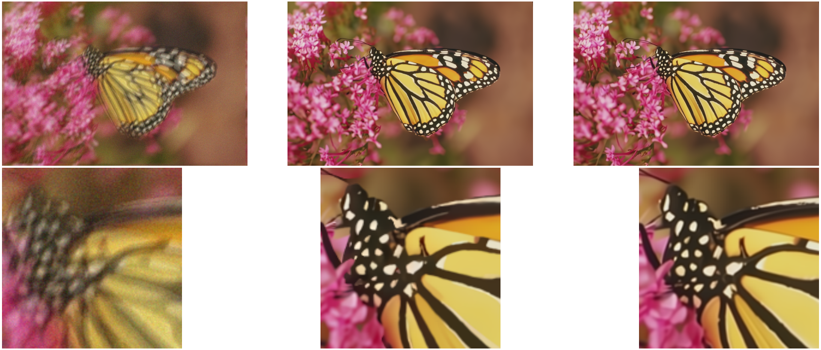


Figure: Plug-and-Play Deblurring of the Monarch image. From left to right: Blurred image, PnP with DRUNet trained on Vibrant Leaves (PSNR = 28.80dB), PnP with DRUNet trained on natural images(PSNR = 29.24dB).

CONCLUSION AND PERSPECTIVES

Summary

Contributions

- Training deep image restoration networks on **Dead Leaves** images → good performance, even for difficult real-world tasks
- **Vibrant Leaves:** a new parametric image model, built upon the dead leaves model, with more natural image properties *(depth, textures, shapes)*
- Experimental validation of the model: from -2db to -0.6 db compared to DL
- A deeper dive into the advantages of synthetic training:
  - Invariance properties
  - Faster convergence of the training algorithm
  - Ablation study: identify crucial image properties for image restoration NNs
  - Sampling from the learnt prior
  - Plug-and-Play applications

Perspectives

- (In Progress): make the model differentiable: optimization of the generation parameters to fit a specific target distribution
- Extend to other modalities: medical images, microscopy, SAR

Achddou, R., Gousseau, Y., and Ladjal, S. (2021).
Synthetic images as a regularity prior for image restoration neural networks.
In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 333–345. Springer.

Achddou, R., Gousseau, Y., and Ladjal, S. (2023a).
Fully synthetic training for image restoration tasks.
Computer Vision and Image Understanding, 233:103723.

Achddou, R., Gousseau, Y., and Ladjal, S. (2023b).
Learning raw image denoising using a parametric color image model.
In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2690–2694. IEEE.

Achddou, R., Gousseau, Y., Ladjal, S., and Süsstrunk, S. (2025).
Vibrantleaves: A principled parametric image generator for training deep restoration models.
arXiv preprint arXiv:2504.10201.

Bibliography II

Alvarez, L., Gousseau, Y., and Morel, J.-M. (1999).
The size of objects in natural and artificial images.
In *Advances in Imaging and Electron Physics*, volume 111, pages 167–242. Elsevier.

Artmann, U. (2015).
Image quality assessment using the dead leaves target: experience with the latest approach and further investigations.
In *Digital Photography XI*, volume 9404, pages 130–144. SPIE.

Baradad, M., Chen, R., Wulff, J., Wang, T., Feris, R., Torralba, A., and Isola, P. (2022).
Procedural image programs for representation learning.
Advances in Neural Information Processing Systems, 35:6450–6462.

Baradad, M., Wulff, J., Wang, T., Isola, P., and Torralba, A. (2021).
Learning to see by looking at noise.
Advances in Neural Information Processing Systems, 34.

Cao, F., Guichard, F., and Hornung, H. (2009).
Measuring texture sharpness of a digital camera.
In *Digital Photography V*, volume 7250, page 72500H. International Society for Optics and Photonics.

Chen, C., Chen, Q., Xu, J., and Koltun, V. (2018).
Learning to see in the dark.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300.

Elad, M., Kawar, B., and Vaksman, G. (2023).
Image denoising: The deep learning revolution and beyond—a survey paper.
SIAM Journal on Imaging Sciences, 16(3):1594–1654.

Galerne, B., Gousseau, Y., and Morel, J.-M. (2010).
Random phase textures: Theory and synthesis.
IEEE Transactions on image processing, 20(1):257–267.

Goujon, A., Neumayer, S., and Unser, M. (2024).
Learning weakly convex regularizers for convergent image-reconstruction algorithms.
SIAM Journal on Imaging Sciences, 17(1):91–115.

Hurault, S., Leclaire, A., and Papadakis, N. (2022).
Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization.
In *International Conference on Machine Learning*, pages 9483–9505. PMLR.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. (2017).
Clevr: A diagnostic dataset for compositional language and elementary visual reasoning.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Kadkhodaie, Z. and Simoncelli, E. (2021).
Stochastic solutions for linear inverse problems using the prior implicit in a denoiser.
Advances in Neural Information Processing Systems, 34:13242–13254.

Kataoka, H., Okayasu, K., Matsumoto, A., Yamagata, E., Yamada, R., Inoue, N., Nakamura, A., and Satoh, Y. (2020).
Pre-training without natural images.
In *Proceedings of the Asian Conference on Computer Vision*.

Laumont, R., Bortoli, V. D., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2022).
Bayesian imaging using plug & play priors: when langevin meets tweedie.
SIAM Journal on Imaging Sciences, 15(2):701–737.

Leclaire, A., Guez, E., and Galerne, B. (2025).
Backward Diffusion iterates Noising-Relaxed Denoising.
working paper or preprint.

Lee, A. B., Mumford, D., and Huang, J. (2001).
Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model.
International Journal of Computer Vision, 41(1-2):35–59.

Lu, L., Achddou, R., and Susstrunk, S. (2025).
Dark noise diffusion: Noise synthesis for low-light image denoising.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

Madhusudana, P. C., Lee, S.-J., and Sheikh, H. R. (2021).
Revisiting dead leaves model: Training with synthetic data.
IEEE Signal Processing Letters.

Matheron, G. (1968).
Modele séquentiel de partition aléatoire.
Technical report, Technical report, CMM.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016).
Playing for data: Ground truth from computer games.
In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer.

Rudin, L. I., Osher, S., and Fatemi, E. (1992).
Nonlinear total variation based noise removal algorithms.
Physica D: nonlinear phenomena, 60(1-4):259–268.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013).
Plug-and-play priors for model based reconstruction.
In *2013 IEEE global conference on signal and information processing*, pages 945–948. IEEE.

Wei, K., Fu, Y., Zheng, Y., and Yang, J. (2021).
Physics-based noise modeling for extreme low-light photography.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

Yu, G. and Sapiro, G. (2011).
DCT Image Denoising: a Simple and Effective Image Denoising Algorithm.
Image Processing On Line, 1:292–296.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. (2021).
Plug-and-play image restoration with deep denoiser prior.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017).
Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising.
IEEE transactions on image processing, 26(7):3142–3155.

Zhang, K., Zuo, W., and Zhang, L. (2018).
Ffdnet: Toward a fast and flexible solution for cnn-based image denoising.
IEEE Transactions on Image Processing, 27(9):4608–4622.