Image Denoising

... Not What You Think

Michael Elad

Computer Science Department
The Technion - Israel Institute of Technology
Haifa 32000, Israel

September 10th 2021
Removal of noise from images is a heavily studied problem in image processing.

In this talk we expand on recent discoveries and developments around this seemingly dead topic.
Our Agenda

1. Brief Introduction & History
2. Image Denoising: The Classic Era
3. The Deep Learning Revolution
4. Synergy: Classic + Deep Learning
5. Our Focus Today: Denoising for ...
   - Solving general inverse problems
   - Image Synthesis
   - High perceptual quality recovery
6. Summary
Introduction & History
So, Let’s Talk About …

Image Denoising

or more accurately

Removal of **White Additive Gaussian Noise** from an Image

Original (clean) Image

Noisy image

Gaussian Noise: $\mathcal{N}(0, \sigma^2 I)$

Image Denoiser

Denoised image
Image Denoising is Challenging

Image denoising is far from trivial task! **Why?**

- Because our goal is to remove noise as much as possible while **preserving** the details in the image
- Denoising is essentially a highly ill-posed separation task

Original (clean) Image  
Noisy image  
Gaussian Noise: $\mathcal{N}(0, \sigma^2 \mathbf{I})$  
Image Denoiser  
Denoised image
1. **Practical:** It is a real-world problem, arising in all cameras,
Why Work on Image Denoising?

1. **Practical:** It is a real-world problem, arising in all cameras,
Why Work on Image Denoising?

1. **Practical:** It is a real-world problem, arising in all cameras,
Why Work on Image Denoising?

1. Practical: It is a real-world problem, arising in all cameras,
Why Work on Image Denoising?

1. **Practical:** It is a real-world problem, arising in all cameras,

2. **Front-Gate to Image Processing:** Being the simplest inverse problem, it is a platform for assessing new ideas in our field, &

3. **Other Uses for the Denoiser Engine:** Recent work has shown that given a denoiser, there are other fascinating uses for it that go far beyond noise removal.
Why Assume Gaussian Noise?

- The Gaussian case is more common and much more important.

- When considering a Poisson noise,
  - High count of photons – The distribution gets closer and closer to the Gaussian case.
  - Low-count Poisson-distributed image can be converted to a Gaussian-noisy one by Anscomb - Variance Stabilizing Transform.

- Many of the developed ideas for the Gaussian case can be converted to other noise models.

- MMSE denoisers for the Gaussian case are of extreme theoretical value (see later).
Image Denoising: Little bit of History

Roughly speaking, there are ~25,000 papers* on this subject, offering algorithms, theoretical analysis and so much more

My speculation

* Search done on June 1st in WoS, topic: ((image or video) and (denoising or (noise and remov) or clean))
Image Denoising: Little bit of History

This research comes from all over the globe

Citing Articles:
USA: 140524
China: 45284
Germany: 29272
France: 35585
England: 24090
Canada: 18325
Spain: 17880
Israel: 13988
Australia: 13358
Switzerland: 12504
Japan: 12389
Italy: 11754
Netherlands: 10455
India: 8830
Finland: 7842
Korea: 7558
Belgium: 5027
Singapore: 4964
Brazil: 4849
Taiwan: 4134
Iran: 3112
Russia: 2595

... and it is heavily cited
The Classic Era
How can we design a denoiser?

The classic Bayesian approach (1960-2014):

- Model image content with a **prior** expression (e.g., forcing smoothness, sparsity, low-rank, self-similarity, ...), and
- Formulate the denoising task as an optimization problem

\[
\hat{x} = \min_x \|x - y\|^2 + \rho(x)
\]

- **Likelihood**
- **Prior**

\(y\) : Given noisy image
\(\hat{x}\) : Denoised result

Design an iterative or a direct algorithm for getting \(\hat{x}\) from \(y\)
Image Denoising: Evolution


- **L₂-based Regularization**
  - Wiener

- **Robust statistics**
  - Hubber-Markov

- **PDE-Methods**
  - Anisotropic Diffusion
  - Beltrami
  - TV

- **Sparsity Methods**
  - NCSR
  - BM3D
  - KSVD

- **Patch-Methods**
  - Kernel-Regr.
  - EPLL

**Heuristic**
- Spatially adaptive filtering
- Wavelet Thresholding
- Cycle-Spinning
- GSM
- SURE

**Self-Similarity Methods**
- NLM
- NLM-PCA
- NL-Bayes

**Low-Rank**
- WNNM
- NL-LR
End of an Era?

This evolution of algorithms and the tendency of different methods to perform very similarly has led to a feeling that “Denoising is Dead.”
End of an Era?

This evolution of algorithms and the tendency of different methods to perform very similarly has led to a feeling that “Denoising is Dead”
End of an Era?

And so, somewhere around 2010-2012, the general feeling in our community was that ...

We are currently touching the ceiling in denoising performance and chances of improving them are very slim

There is no point in devising new denoising methods

Work in this field has diminishing returns

Well, We Were Wrong!
End of an Era?

Wrong? How?

The past decade has taught us that image denoising is still very much alive and kicking due to several branches of novel activity on:

- Obtaining better performing denoisers with deep learning
- New frontiers in denoising:
  - Better adaptation to image content
  - Denoising strategies that go beyond PSNR
  - Identifying alternative methods for designing/training denoisers
  - Extending the denoising task to realistic noise, and
- Discovering new ways for leveraging denoisers for other needs
The Deep Learning Revolution
How can we ALTERNATIVELY design a denoiser?

The machine learning approach (2012-Now):

- Gather a LARGE dataset of clean images \( \{x_k\}_{k=1}^{N} \)
- Add AWGN these images: \( \{y_k = x_k + n_k\}_{k=1}^{N} \)
- Define a parametric denoising machine \( D_\theta(y) \)
- Train \( D_\theta(y) \) by setting its parameters \( \theta \): 

\[
\min_\theta \sum_{k=1}^{N} \| x_k - D_\theta(y_k) \|^2
\]
Image Denoising: A Paradigm Shift

How can we design a denoiser?

By modeling image content and leveraging it for noise filtering:

- Sparse Representation
- Scale Invariance
- Low Rank
- Low dimensionality
- GMM
- Piecewise Smoothness
- Non-Local Self-Similarity

Observe that with this trend, all the knowledge and knowhow accumulated carefully over decades in image processing became TOTALLY OBSOLETE.
Image Denoising: Recent Evolution

Initial Steps - MLP
(Burger et. al)
CVPR 2012

Deep Shrinkage
Isogawa et. al
IEEE-SPL 2017

Deep Image Prior
(Ulyanov & Vedaldi)
CVPR 2018

Denoising Auto-
Encoder
(Cho)
ICML 2013

TNRD
(Chen & Pock)
IEEE-TPAMI 2016

CNLNet
(Lefkimmiatis)
CVPR 2017

Noise2Void
(Krull et. al)
CVPR 2019

Learned Prox
(Meinhardt et. al)
ICCV 2017

FFDNet (blind)
(Zhang et. al)
IEEE-TIP 2017

Batch Renormalization
(Tian et. al)
Neural Networks
2020

CBDNet (blind)
(Guo et. al)
CVPR 2019

Attention
(Tian et. al)
Neural Networks
2020

DnCNN
(Zhang et. al)
IEEE-TIP 2017

GCBD
GAN-based
(Chen et. al)
CVPR 2018

Michael Elad
The Computer-Science Department
The Technion

21
Synergy: Classics + Deep Learning
Image Denoising: Return of the Classics

- In recent years deep learning is ruling the image denoising domain, pushing aside all the classical methods, along with their great achievements.
- Recently, however, we do see a synergy between the two paradigms.
- Recall: In building a supervised deep learning denoiser solution, we operate along the following lines:
  1. Gather training data to use
  2. Define an architecture for the Denoiser
  3. Define a cost function (loss) to optimize
  4. Train and hope for good generalization
So, how do we choose an architecture for a given task?

**Option 1** - Copy an existing network that has shown good results in earlier work (VGG, U-Net, ...), and slightly modify it

**Option 2** – Pile and Guess a series of steps that mix known pieces such as convolutions, fully connected layer, batch-norm, ReLU, pooling, stride, skips, upscale/downscale, connections, ... and add new “tricks”

**Option 3** – Neural Architecture Search
Here are several paper examples from CVPR/NIPS 2019 that illustrate these architectures:

- **CVPR 2019**: U-Net-based with $3.8 \times 10^6$ params
- **CVPR 2019**: DnCNN-based with $5.5 \times 10^5$ params
- **CVPR 2019**: U-Net-based with $5.3 \times 10^6$ params
- **NIPS 2019**: U-Net-based with $1.1 \times 10^6$ params
- **CVPR 2019**: DnCNN-based with $1.2 \times 10^6$ params
Message: Do far better in choosing architectures by relying on unfolding algorithms from the classics of image processing

The benefits in such architectures:
- They are far more concise yet just as effective as leading methods
- They are easier to train because they are lighter
- They have the potential to break current performance barriers
- They may bring better understanding and explainability
- They enable better adaptation to out of distribution images

Here are few representative examples:
- Rethinking the CSC Model [Simon & Elad, NIPS `19]
- Non-Local & Multi-Scale Denoising [Vaksman, Milanfar & Elad, CVPR (NTIRE) `20]
- Deep KSVD Denoising [Scetbon, Milanfar & Elad, IEEE-TIP `21]
- PatchCraft: Non-Local Video Denoising [Vaksman, Elad & Milanfar, ArXiV `21]
Our Focus Today: Recent Discoveries
Recent findings on using denoisers for other tasks:

- **Discovery 1:** Solving general inverse problems [2013-]
- **Discovery 2:** Image Synthesis [2019-]
- **Discovery 3:** High perceptual quality recovery [2021-]

We turn to describe these results.
Discovery 1: Solving Inverse Problems

Inverse Problems: Recovery of images from corrupted measurements

- De-Noising
- De-Blurring
- In-Painting
- De-Mosaicing
- Tomography
- Image Scale-Up & super-resolution
- ... and more ...

\[ y = Hx + n \]
How can we solve inverse problems?

We can return to the classic Bayesian approach:

- Model image content with a prior expression (e.g., forcing smoothness, sparsity, low-rank, self-similarity, ...), and
- Formulate the inversion task as an optimization problem

\[
\hat{x} = \min_x \|Hx - y\|^2 + \rho(x)
\]

- This is known as MAP estimation
- It is an extension of the classic path for denoising, tailoring methods for inverse problems
- This approach leads to iterative algorithm for getting \(\hat{x}\) from \(y\)
- Is there a supervised learning alternative? Definitely!
Discovery 1: Solving Inverse Problems

**Question:** Given a denoiser \( D(y, \sigma) \) how can one solve inverse problems with it?

**Answer:** Use \( D(y, \sigma) \) as a *regularizer*

**Practical Implication:** Iterated use of \( D(\cdot, \sigma) \)

---

**Plug-and-play priors for model based reconstruction**
SV Venkatakrishnan, CA Bouman, B Wohlberg
2013 IEEE Global Conference on Signal and Information Processing, 945-948

**The little engine that could: Regularization by denoising (RED)**
Y Romano, M Elad, P Milanfar
SIAM Journal on Imaging Sciences 10 (4), 1804-1844
Here is (roughly) the PnP Perspective in a nutshell:

- Recall: Inverse problems can be formulated as optimization tasks:
  \[ \hat{x} = \min_x \frac{1}{2} \|Hx - y\|^2 + \rho(x) \]

- Let’s do something “stupid” and split the unknown:
  \[ \hat{x} = \min_{x,v} \frac{1}{2} \|Hx - y\|^2 + \rho(v) \text{ s.t. } x = v \]

- Now, turn the constraint into a penalty*:
  \[ \hat{x} = \min_{x,v} \frac{1}{2} \|Hx - y\|^2 + \rho(v) + \beta \|x - v\|^2 \]

- And solve by alternating between x and v
  - Least-Squares: \( \hat{x} = \min_x \frac{1}{2} \|Hx - y\|^2 + \beta \|x - v\|^2 \)
  - A denoiser: \( \hat{v} = \min_v \rho(v) + \beta \|x - v\|^2 \)

... and this way we got an iterated algorithm that keeps calling to a denoiser, for solving the inverse problem

* The PnP uses the Augmented Lagrange which is more accurate and less sensitive to the choice of \( \beta \)
Let’s start again with the formulated optimization task, and suggest a very specific regularization term:

\[
\hat{x} = \min_x \frac{1}{2} \|Hx - y\|^2 + \rho(x) = \min_x \frac{1}{2} \|Hx - y\|^2 + \lambda x^T [x - D(x, \sigma)]
\]

Under mild conditions* the gradient of this is \([x - D(x, \sigma)]\)

\[
\hat{x}_{k+1} = \hat{x}_k - \mu [H^T (H\hat{x}_k - y) + \lambda [\hat{x}_k - D(\hat{x}_k, \sigma)]]
\]

... and this way we got an iterated algorithm that keeps calling to a denoiser, and is guaranteed to achieve the minimum

* Differentiability, local homogeneity, passivity and symmetric Jacobian (MMSE)
Discovery 1: Solving Inverse Problems

Here are some results for Deblurring and Super-Resolution

(a) Ground Truth  (b) Input 20.8

(a) Bicubic 20.68dB  (b) NCSR 26.79dB

(d) RED: SD-TNRD 27.39dB

(c) P^3-TNRD 26.61dB  (d) P^3-TNRD 28.39dB

(e) NCSR 28.39dB
Discovery 1: Solving Inverse Problems

- PnP and RED are heavily cited and extensively studied, owing to their generality and elegance.

- Follow-up work focuses on:
  - Proving convergence to the desired solution and tying these to properties of the permissible denoisers (e.g. MMSE ...)
  - Deployment in various applications
  - Creation of new variants of these two methods ... and ...

- PnP/RED can be used to define well-motivated architectures for solving general inverse problems, built around a core learned denoising engine.
In recent years, and with the deep-learning revolution, there is a growing interest in synthesizing images “out of thin air”.

The popular tool of interest is called GAN – Generative Adversarial Network, built of two competing networks – a generator and a critic.

Why synthesize? Because:
- We can, and it is fascinating.
- It testifies that we have seized the distribution of images, and
- It could be used for other needs.

Could we synthesize images differently?
Discovery 2: Image Synthesis

**Question:** Given a denoiser $D(y, \sigma)$ how can one synthesize images with it?

**Answer:** Use $D(y, \sigma)$ as a *Projector* onto the image manifold

**Practical Implication:** Iterated use of $D(\cdot, \sigma)$ with varying $\sigma$

---

Michael Elad  
The Computer-Science Department  
The Technion
Here is the core idea in a nutshell:

Our goal: draw a sample from the distribution of images $P(x)$

- Start with a random noise image $\hat{x}_0$
- Climb to a more probable image by the iterative equation:

$$\hat{x}_{k+1} = \hat{x}_k + a \cdot \nabla \log P(\hat{x}_k) + b \cdot z_k$$ (Langevin Dynamics)

This is known as the Score Function and it is approximately proportional to $[\hat{x}_k - D(\hat{x}_k, \sigma)]$ for a small value of $\sigma$

This suggests an implicit relation between MMSE denoisers and Priors:

$D(x, \sigma) \leftrightarrow P(x)$

... and this way we got an iterated algorithm that keeps calling to a denoiser, and is guaranteed to obtain a sample from $P(x)$
In practice, instead of the plain Langevin with a fixed (and small) value of $\sigma$ we use the **Annealed Langevin Algorithm** that considers a sequence of blurred priors:

$$
P(x + v) \text{ for } v \sim \mathcal{N}(0, \sigma_k^2 I)
$$

$$
= P(x) \otimes c \cdot \exp \left\{ -\frac{1}{2\sigma^2} \|x\|^2 \right\}
$$

with $\sigma_0 > \sigma_1 > \sigma_2 \cdots > \sigma_N > 0$

The core idea: start by drawing from a wider distribution and gradually narrow it, leading to a faster sampling performance.
Discovery 2: Image Synthesis

Does it work? Here are some results

Kadkhodaie & Simoncelli

Song & Ermon

Michael Elad
The Computer-Science Department
The Technion
Discovery 2: Image Synthesis

Claim: diffusion-based methods are the best in image synthesis

BigGAN (FID 6.95)  Diffusion (FID 4.59)

Abstract

We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction
Suppose that we need to denoise the following image:

Should we be pleased with this result? It seems to be a bit ... blurry, no? Why?

Minimum Mean-Squared-Error (MMSE) denoisers are great for MSE result, but their result falls outside the manifold.
Question: How can we denoise an image while targeting “High Perceptual Quality”?

Answer: Denoise by sampling from the posterior $P(x|y)$

Practical Implication: We consider 2 methods

- Training a deep denoiser via CGAN, or
- Iterated use of an MMSE denoiser $D(\cdot, \sigma)$

These methods produce a multitude of possible solutions
Discovery 3: Targeting Perceptual Quality

Let’s have a closer look at the Stochastic Image Denoiser:

Task: Draw a sample from $P(x|y)$ where $[y = x + n, n \sim \mathcal{N}(0, \sigma_0^2 I)]$

- Start with a random noise image $\hat{x}_0$
- Climb to a more probable image by the iterative equation:
  
  $$\hat{x}_{k+1} = \hat{x}_k + a \cdot \nabla \log P(\hat{x}_k|y) + b \cdot z_k$$

  \[\leftarrow\text{Langevin with a conditional Score}\]

  
  Bayes rule

  $$\hat{x}_k - D(\hat{x}_k, \sigma) + \nabla \log P(y|\hat{x}_k)$$

  \[\leftarrow\text{Approx. Score}\quad \leftarrow\text{A Gaussian Distribution}\]
Discovery 3: Targeting Perceptual Quality

Let’s have a closer look at the 

\[ \nabla \log P(\hat{x}_k|y) = \hat{x}_k - D(\hat{x}_k, \sigma) + \nabla \log P(y|\hat{x}_k) \]

- As we use the Annealed Langevin algorithm, there are two noise signals to consider:
  - Measurement’s noise: \( n \sim \mathcal{N}(0, \sigma_0^2 I) \)
  - Synthetic annealing noise: \( v \sim \mathcal{N}(0, \sigma_k^2 I) \) for \( \sigma_0 > \sigma_1 > \sigma_2 \ldots > \sigma_N > 0 \)

- Implication: We recover a sequence of gradually less noisy images \( \hat{x}_k \) where their noise \( v \) is assumed to be a portion of \( n \)
Discovery 3: Targeting Perceptual Quality

Stochastic Image Denoiser:

- We start from a noisy image \( \sigma \approx 150 \) in this example
- Then gradually denoise it using (conditional) annealed Langevin dynamics
Discovery 3: Targeting Perceptual Quality

Stochastic Image Denoiser:
Discovery 3: Targeting Perceptual Quality

Stochastic Image Denoiser:
Discovery 3: Targeting Perceptual Quality

Stochastic Image Denoiser:
Discovery 3: Targeting Perceptual Quality

Stochastic Image Denoiser:
Let’s have a closer look at the **Conditional GAN Denoiser**:

- **Typical design approach**: Optimize a distortion measure (e.g. MSE) between the denoised and the ideal images.

- **Adversarial loss** could be added to improve the perceptual quality.

- However, when used together, we get a compromise between distortion and perceptual quality.
Discovery 3: Targeting Perceptual Quality

- For ill-posed restoration tasks, perceptual quality performance comes at the expense of its distortion [Blau & Michaeli 2017]
- We aim for best perceptual quality
- The posterior distribution attains perfect perceptual quality, compromising 3dB compared to the MMSE [Blau & Michaeli 2017]
- We propose to sample from the posterior via a Conditional GAN mechanism (PSCGAN)

\[ x \sim P_X \quad y \sim P_{Y|X=x} \]

Samples from \( P_{X|Y=y} \)
The PSCGAN Architecture:

Why use $y$ in the critic? Without it, we check only the perceptual quality of the denoised result. With it, we also assess its denoising validity.
Discovery 3: Targeting Perceptual Quality

What about the Loss?

- CGAN optimization leads to posterior sampling [Adler et al. 2018]:
  $\min_{\theta} \max_{\omega} \mathbb{E}_{X,Y}[C_{\omega}(x,y)] - \mathbb{E}_{D_{\theta},Y,Z}[C_{\omega}(D_{\theta},y)]$

- However, this requires an unavailable balanced dataset to train on (many $x$’s for each $y$ and many $y$’s for each $x$)

- On the other hand, we would like to avoid a penalty of the form
  $\mathbb{E}_{X,Y,Z}[\|x - D_{\theta}(y, z)\|^2_2]$

- Our remedy: adding an MMSE oriented penalty term:
  $\mathbb{E}_{X,Y}[\|x - \mathbb{E}_z[D_{\theta}|y]\|^2_2]$

- This gives the MMSE result “for free” (averaging many instances)
Discovery 3: Targeting Perceptual Quality

CGAN:
Discovery 3: Targeting Perceptual Quality

CGAN:
Oh ... and One Last Thing
Back to Inverse Problems

- Goal: Recovery from corrupted measurements
  - De-Noising
  - De-Blurring
  - In-Painting
  - De-Mosaicing
  - Tomography
  - Image Scale-Up & super-resolution

- Can we suggest a “sampler” from $P(x|y)$ for handling all these problems, in order to obtain “perfect looking” results?

- Answer: Yes! Use Langevin dynamics again, in an adapted form

**SNIPS: Solving Noisy Inverse Problems Stochastically**
B Kawar, G Vaksman, M Elad
arXiv preprint arXiv:2105.14951
The idea is similar to our high-perceptual denoising, with necessary changes for considering the degradation operator $H$ ...

Starting naively, using Bayes theorem, we need to calculate

$$\nabla \log P(x_i|y) = \nabla \log P(x_i) + \nabla \log P(y|x_i)$$

We know that $y = Hx + n$ and thus:

$$\nabla \log P(y|x_i) = \nabla \log P(y - Hx_i|x_i) =$$

$$\nabla \log P(Hx + n - Hx - Hv_i|x_i) = \nabla \log P(n - Hv_i|x_i)$$

However, ... while $n - Hv_i$ is a simple Gaussian, it’s dependency on $x_i$ is non-trivial, so how do we proceed from here?
Step 1: Use SVD for decoupling the measurements $H = UΣV^T$:

\[
UTy = UT[UΣVT(x_i - v_i) + n] = ΣVT(x_i - v_i) + UTn
\]

Thus, we can apply the Langevin dynamics algorithm on $\hat{x}_T = V^Tx_i$ given $y_T = UTy$ and sample from the conditional $\tilde{v}_T[k]$ to be a portion of $n_T[k]$

Bottom line: An MMSE denoiser is used for a novel solution of inverse problems, this time targeting best perceptual quality.
Back to Inverse Problems

Noisy Inpainting: A portion missing and noise with $\sigma_0 \approx 25$
Back to Inverse Problems

Super resolution: downscaling by 4 with additive noise of $\sigma_0 \approx 25$
Super resolution: downscaling by 4 with additive noise of $\sigma_0 \approx 12$
Back to Inverse Problems

Deblurring: uniform $5 \times 5$ blur with additive noise of $\sigma_0 \approx 25$
Compressive sensing (12.5%) with additive noise of $\sigma_0 \approx 25$
Compressive sensing (12.5%) with additive noise of $\sigma_0 \approx 25$

<table>
<thead>
<tr>
<th>Original</th>
<th>Degraded</th>
<th>Sample</th>
<th>Mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
And just to remind you ...

The proposed diffusion-based sampling scheme, while quite appealing, suffers from several key shortcomings:

- It is rather S L O W (many denoising activations)
- It is limited (as of now) to specific families of images
- Relying on SVD is cumbersome
Time to Summarize
Image Denoising

... Not What You Think

1. There are so many opportunities and challenges in better understanding, designing, and proposing creative usage of image denoisers.

2. Despite the fact that this has not been a talk about Deep-Learning, the presence of this field in the topics covered is prominent.