Inverse Problems and Generative Models

Arthur Leclaire

TELECOM Paris
IP PARIS

MVA Generative Modeling
February, 6th, 2024

Today

- We will discuss imaging inverse problems.
- We will recall classical (simple) tools for solving inverse problems.
  In particular we will recall simple regularization techniques (Tychonov, smoothTV)
- We will discuss quantitative evaluation of image restoration.
- **We will see how to use generative models to solve inverse problems.**

Plan

Inverse problem with additive noise:

$$v = \mathcal{A}u_0 + w$$

where

- $u_0 \in \mathbf{R}^d$ is the clean image to recover
- $\mathcal{A} : \mathbf{R}^d \to \mathbf{R}^m$
- w is a noise

In many cases, the degradation operator $\mathcal{A}$ can be approximated with a linear operator A, and the noise model w is assumed to be Gaussian.

But, there are also inverse problems with non-linear $\mathcal{A}$ and non-Gaussian noise (e.g. Poisson noise).

Classical Inverse Problems

Application	Forward model	Notes
Denoising [58]	$A = I$	I is the identity matrix
Deconvolution [58, 59]	$\mathcal{A}(\boldsymbol{x}) = \boldsymbol{h} * \boldsymbol{x}$	$\boldsymbol{h}$ is a known blur kernel and $*$ denotes convolution. When $\boldsymbol{h}$ is unknown the reconstruction problem is known as blind deconvolution.
Superresolution [60, 61]	$A = SB$	S is a subsampling operator (identity matrix with missing rows) and B is a blurring operator cooresponding to convolution with a blur kernel
Inpainting [62]	$A = S$	S is a diagonal matrix where $S_{i,i} = 1$ for the pixels that are sampled and $S_{i,i} = 0$ for the pixels that are not.
Compressive Sensing [63, 64]	$A = SF$ or $A =$ Gaussian or Bernoulli ensemble	S is a subsampling operator (identity matrix with missing rows) and F discrete Fourier transform matrix.
MRI [3]	$A = SFD$	S is a subsampling operator (identity matrix with missing rows), F is the discrete Fourier transform matrix, and D is a diagonal matrix representing a spatial domain multiplication with the coil sensitivity map (assuming a single coil aquisition with Cartesian sampling in a SENSE framework [65]).
Computed tomography [58]	$A = R$	R is the discrete Radon transform [66].
Phase Retrieval [67–70]	$\mathcal{A}(\boldsymbol{x}) = \lvert A\boldsymbol{x}\rvert^2$	$\lvert\cdot\rvert$ denotes the absolute value, the square is taken elementwise, and A is a (potentially complex-valued) measurement matrix that depends on the application. The measurement matrix A is often a variation on a discrete Fourier transform matrix.

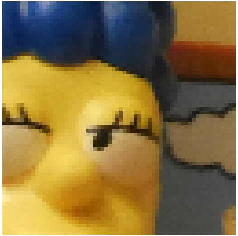
(source: [Ongie et al., 2020])

Gaussian denoising

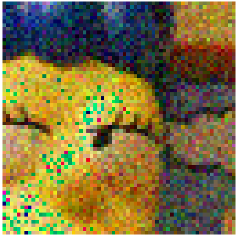
Let's start with the case $A = \mathrm{Id}$, i.e. **image denoising**:

$$v = u_0 + w \quad \text{where} \quad w \sim \mathcal{N}(0, \sigma^2).$$

We want to estimate u_0 from a single realization of v... need for some image model.



u_0



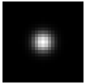
v

Deblurring

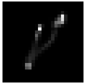
- A spatially invariant blur can be modeled by a convolution operator $Au = k * u$
- Several types of blur exist (motion, defocus)
- **Non-blind deblurring** consists in recovering u_0 from

$$v = k * u_0 + w.$$

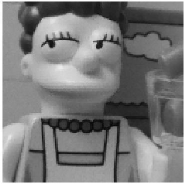
- We won't tackle blind deblurring here.



Isotropic blur



Motion blur



Original

Blurred

Example of motion blur

Super-Resolution

Super-résolution consists in finding another version of v at higher resolution.

This is an inverse problem corresponding to the subsampling operator with stride $s \in \mathbb{N}^*$:

$$u_{\downarrow s}(x, y) = u(sx, sy).$$

In practice, we often apply an (*anti-aliasing*) filter before subsampling.

With prefiltering, we obtain the operator

$$Au = (k * u)_{\downarrow s}.$$

Super-resolution consists in recovering u_0 from

$$v = (k * u)_{\downarrow s} + w.$$

The degraded image v is defined on a subgrid of stride s.

Inpainting

Inpainting consists in filling missing regions in images

The degradation operator then writes

$$Au = u\mathbf{1}_\omega$$

where $\omega \subset \Omega$ is the set of known pixels and $\Omega \setminus \omega$ the mask.

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

$$u = \mathcal{F}^{-1}\left(\frac{\hat{v}}{\hat{k}}\right) = \mathcal{F}^{-1}\left(\hat{u}_0 + \frac{\hat{w}}{\hat{k}}\right) \quad \longrightarrow \quad \text{but noise explodes !}$$

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

$$u = \mathcal{F}^{-1}\left(\frac{\hat{v}}{\hat{k}}\right) = \mathcal{F}^{-1}\left(\hat{u}_0 + \frac{\hat{w}}{\hat{k}}\right) \quad \longrightarrow \quad \text{but noise explodes !}$$

When the problem is ill-posed, there may be multiple solutions or erroneous solutions.

It is thus useful to adopt an *a priori* on the solution, e.g. imposing some kind of regularity.

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname*{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \text{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \text{Id}$),

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \operatorname{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^TA + \lambda I)^{-1}A^Tv$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \operatorname{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - y\|_2^2$ is

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\underset{u \in \mathbf{R}^\Omega}{\mathrm{Argmin}}\, F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \mathrm{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - y\|_2^2$ is $\nabla f(u) = A^T(Au - y)$.

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \operatorname{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^TA + \lambda I)^{-1}A^Tv$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - y\|_2^2$ is $\nabla f(u) = A^T(Au - y)$.

If $Au = k * u$ (periodic convolution), then $A^T u =$

Image Restoration by Optimization

We will therefore try to solve

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda R(u)$$

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname*{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \text{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

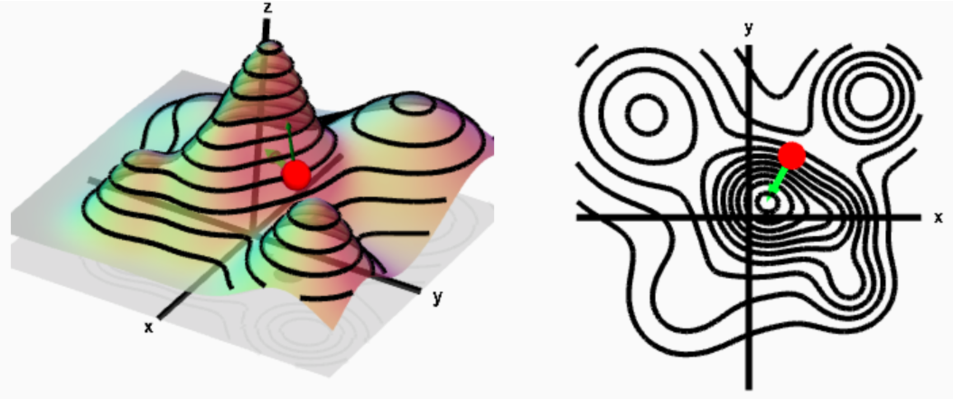
Example: for denoising ($A = \text{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - y\|_2^2$ is $\nabla f(u) = A^T(Au - y)$.

If $Au = k * u$ (periodic convolution), then $A^T u = \tilde{k} * u$ with $\tilde{k}(\mathbf{x}) = \overline{k(-\mathbf{x})}$.

The Steepest Descent



https://mathinsight.org/directional_derivative_gradient_introduction

Descent Lemma

Let $f : \mathbf{R}^d \to \mathbf{R}$ be differentiable with L-Lipschitz gradient. Then, for any $x, y \in \mathbf{R}^d$,

$$\begin{aligned} f(y) &= f(x) + \int_0^1 \nabla f(x + t(y-x)) \cdot (y-x)dt \\ &= f(x) + \nabla f(x) \cdot (y-x) + \int_0^1 \left(\nabla f(x+t(y-x)) - \nabla f(x)\right) \cdot (y-x)dt \\ &\leq f(x) + \nabla f(x) \cdot (y-x) + \int_0^1 \|\nabla f(x+t(y-x)) - \nabla f(x)\| \|y-x\| dt \\ &\leq f(x) + \nabla f(x) \cdot (y-x) + \int_0^1 Lt\|y-x\|^2 dt \\ &\leq f(x) + \nabla f(x) \cdot (y-x) + \frac{L}{2}\|y-x\|^2. \end{aligned}$$

Consequence: If we choose $\tau \in [0, \frac{2}{L})$, then

$$f(x - \tau \nabla f(x)) \leq f(x) - \tau\left(1 - \frac{\tau L}{2}\right)\|\nabla f(x)\|^2 \leq f(x).$$

Gradient Descent

We consider here the gradient descent method:

$$x_{n+1} = x_n - \tau_n \nabla f(x_n) \ ,$$

where $\tau_n > 0$ is a sequence of step sizes.

- For $\tau_n = \tau$ constant, we speak of fixed step size.
- We speak of optimal step size if, at each iteration n, we choose

$$\tau_n \in \operatorname*{Argmin}_{t \in \mathbf{R}} f(x_n - t\nabla f(x_n)).$$

The descent lemma gives that for f differentiable with L-Lipschitz gradient and $\underline{\tau < \frac{2}{L}}$,

$$f(x_{n+1}) \leqslant f(x_n)$$

Thus, if f is lower bounded, $f(x_n)$ converges.

Convexity and Minimum

The function $f : \mathbf{R}^d \to \mathbf{R}$ is convex if for all $x, y \in \mathbf{R}^d$,

$$\forall t \in (0,1), \quad f((1-t)x + ty) \leqslant (1-t)f(x) + tf(y).$$

It is said strictly convex if the inequality is strict.

If f is convex and differentiable, one can show that for any $x, y \in \mathbf{R}^d$,

$$f(y) \geqslant f(x) + \nabla f(x) \cdot (y - x).$$

Consequence : If f is convex and differentiable, then

$$x \in \text{Argmin} f \quad \Longleftrightarrow \nabla f(x) = 0.$$

The argmin is unique as soon as f is strictly convex.

Strong Convexity

We say that f is α-convex (with $\alpha > 0$) if $f - \frac{\alpha}{2}\|\cdot\|^2$ is convex.

When $\alpha > 0$, we say that f is **strongly convex**.

Remark : The convexity and the gradient Lipschitz constant can be read on the Hessian.

If $A, B \in \mathbf{R}^{d\times d}$ are symmetric, we write $A \succeq B$ if $A - B$ if semi-definite positive, i.e.

$$\forall x \in \mathbf{R}^d, \quad Ax \cdot x \geq Bx \cdot x.$$

For $f : \mathbf{R}^d \to \mathbf{R}$ of class $\mathscr{C}^2$,

$$\nabla f \text{ is } L\text{-lipschitz iff } \forall x \in \mathbf{R}^d, \quad -L\mathrm{Id} \preceq \nabla^2 f(x) \preceq L\mathrm{Id}.$$
$$\text{i.e. } \forall x \text{ the eigenvalues of } \nabla^2 f(x) \text{ have modulus} \leq L.$$

$$f \text{ is } \alpha\text{-convex iff } \nabla^2 f \succeq \alpha\mathrm{Id}$$
$$\text{i.e. } \forall x \text{ the eigenvalues of } \nabla^2 f(x) \text{ are all} \geq \alpha.$$

Convergence Guarantees, Convex Case

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be convex differentiable with ∇f L-Lipschitz. Assume that Argmin f is non-empty. Let $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$ and (x_n) the sequence defined by

$$x_{n+1} = x_n - \tau \nabla f(x_n) \; .$$

Then (x_n) converges towards an element of Argmin f.

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be differentiable and α-strongly convex with L-Lipschitz gradient. Then there exists a unique $x_* \in$ Argmin f, and for $\tau < \frac{1}{L} \leqslant \frac{1}{\alpha}$, we have

$$\|x_n - x_*\|^2 \leqslant (1 - \tau\alpha)^n \|x_0 - x_*\|^2.$$

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

$$F(u) = f(u) + g(u)$$

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

For a regularization $R(u) = \|Bu\|_2^2$, F is convex and differentiable.

We can thus minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

$$F(u) = f(u) + g(u)$$

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

For a regularization $R(u) = \|Bu\|_2^2$, F is convex and differentiable.

We can thus minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Good news: By automatic differentiation you need only coding $F(u)$...

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

$$F(u) = f(u) + g(u)$$

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

For a regularization $R(u) = \|Bu\|_2^2$, F is convex and differentiable.

We can thus minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Good news: By automatic differentiation you need only coding $F(u)$...

But ! in order to avoid instability problems, you'd better know what F does...
(for example, useful to have an idea of the Lipschitz constant of F)

Let us start with zero regularization!

Consider here

$$f(u) = \frac{1}{2}\|Au - v\|^2.$$

- We have an orthogonal decomposition $\mathbf{R}^d = K \oplus K^\perp$ with $K = \mathrm{Ker}[A]$ and $K^T = \mathrm{Im}[A^T]$
- Therefore $\mathrm{Argmin}_{\mathbf{R}^d} f$ is non-empty and we can define

$$A^+ v = \min_{u \in \mathrm{Argmin}\, f} \|u\|_2^2.$$

  It defines a linear operator A^+, called Moore-Penrose pseudo-inverse.
- The Moore-Penrose pseudo-inverse has a zero component in $\mathrm{Ker}[A]$.
- $A_{K^T} : K^T \to \mathrm{Im}(A)$ is invertible. Thus $A^+ = A_{K^T}^{-1} P$ (with P the orthogonal projection on $\mathrm{Im}(A)$).
- Actually, one can show that $A^+ v = \lim_{\lambda \to 0} (A^T A + \lambda I)^{-1} A^T v$.
- But $A^+ v$ is generally a bad solution for inverse problems because of bad conditioning.

Explicit Regularizations

We define the discrete derivatives of u by

$$\nabla u(x,y) = \begin{pmatrix} \partial_1 u(x,y) \\ \partial_2 u(x,y) \end{pmatrix} \quad \text{avec} \quad \begin{cases} \partial_1 u(x,y) = d_1 * u(x,y) = u(x+1,y) - u(x,y) \\ \partial_2 u(x,y) = d_2 * u(x,y) = u(x,y+1) - u(x,y) \end{cases} .$$

We define **Tychonov regularization** by

$$\|\nabla u\|_2^2 = \sum_{\mathbf{x}\in\Omega} \|\nabla u(\mathbf{x})\|^2 = \sum_{\mathbf{x}\in\Omega} |\partial_1 u(\mathbf{x})|^2 + |\partial_2 u(\mathbf{x})|^2.$$

We define the **total variation** by

$$\mathrm{TV}(u) = \|\nabla u\|_1 = \sum_{\mathbf{x}\in\Omega} \|\nabla u(\mathbf{x})\| = \sum_{\mathbf{x}\in\Omega} \sqrt{|\partial_1 u(\mathbf{x})|^2 + |\partial_2 u(\mathbf{x})|^2}.$$

Back to denoising

Let us minimize

$$F(u) = \frac{1}{2}\|u - v\|^2 + \lambda R(u)$$

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$**.**

Back to denoising

Let us minimize

$$F(u) = \frac{1}{2}\|u - v\|^2 + \lambda R(u)$$

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$**.**

We have $\nabla R(u) = 2\nabla^T \nabla u$.

Back to denoising

Let us minimize

$$F(u) = \frac{1}{2}\|u - v\|^2 + \lambda R(u)$$

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$**.**

We have $\nabla R(u) = 2\nabla^T \nabla u$. As F is convex,

$$u \in \operatorname{Argmin} F \iff \nabla F(u) = 0 \iff u - v + 2\lambda\nabla^T\nabla u = 0 \iff u = (I + 2\lambda\nabla^T\nabla)^{-1} v$$

Back to denoising

Let us minimize

$$F(u) = \frac{1}{2}\|u - v\|^2 + \lambda R(u)$$

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$.

We have $\nabla R(u) = 2\nabla^T \nabla u$. As F is convex,

$$u \in \operatorname{Argmin} F \iff \nabla F(u) = 0 \iff u - v + 2\lambda\nabla^T\nabla u = 0 \iff u = (I + 2\lambda\nabla^T\nabla)^{-1} v$$

For $p : \Omega \to \mathbf{R}^2$, $\nabla^T p$ is given by

$$\nabla^T p(x, y) = p_1(x-1, y) - p_1(x, y) + p_2(x, y-1) - p_2(x, y).$$

Actually, $\operatorname{div}(p) := -\nabla^T p$ is a discrete divergence and $\Delta u := -\nabla^T \nabla u$ is a discrete Laplacian.

Explicit Solution: Wiener filtering

Theorem

Let $v \in \mathbb{C}^\Omega$ and $\lambda > 0$. The function $F : \mathbb{C}^\Omega \to \mathbf{R}_+$ defined by

$$\forall u \in \mathbb{C}^\Omega, \quad F(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda \|\nabla u\|_2^2$$

has a minimum attained at a unique $u_* \in \mathbb{C}^\Omega$, which is given in Fourier domain:

$$\forall (\xi, \zeta) \in \Omega, \quad \hat{u}_*(\xi, \zeta) = \frac{\hat{v}(\xi, \zeta)}{1 + 2\lambda\, \hat{L}(\xi, \zeta)}$$

where $\hat{L}(\xi, \zeta) = |\hat{d}_1(\xi, \zeta)|^2 + |\hat{d}_2(\xi, \zeta)|^2 = 4\left(\sin^2\left(\frac{\pi\xi}{M}\right) + \sin^2\left(\frac{\pi\zeta}{N}\right)\right)$.

Remarks:

- d_1, d_2 are the kernel derivatives, e.g. $d_1 = \delta_{(-1,0)} - \delta_{(0,0)}$. So $\hat{L}$ is the kernel of $-\Delta$ filter.
- The theorem adapts for deblurring with Tychonov regularization:

$$\forall (\xi, \zeta) \in \Omega, \quad \hat{u}_*(\xi, \zeta) = \frac{\overline{\widehat{k}(\xi, \zeta)}\hat{v}(\xi, \zeta)}{|\widehat{k}(\xi, \zeta)|^2 + 2\lambda\, \hat{L}(\xi, \zeta)}$$

Link with an evolution model

The gradient descent on

$$F(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda\|\nabla u\|_2^2$$

writes as

$$u_{n+1} - u_n = -\tau(u_n - v) + 2\lambda\tau\Delta u_n \ .$$

The sequence (u_n) converges to u_* as soon as $\tau < \frac{2}{L}$ with $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$.

Link with an evolution model

The gradient descent on

$$F(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda\|\nabla u\|_2^2$$

writes as

$$u_{n+1} - u_n = -\tau(u_n - v) + 2\lambda\tau\Delta u_n \ .$$

The sequence (u_n) converges to u_* as soon as $\tau < \frac{2}{L}$ with $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$.

If we drop the data-fidelity... then gradient descent on $u \mapsto \|\nabla u\|_2^2$ gives

$$u_{n+1} - u_n = 2\tau\Delta u_n$$

This is a discretization of the heat equation $\partial_t u = c\Delta u$ with initial condition u_0.

Smoothed Total Variation

What if we want to minimize

$$F(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda \mathrm{TV}(u).$$

Problem: The total variation is not differentiable.

A simple solution: Consider a smoothed variant: For $\varepsilon > 0$, let

$$\mathrm{TV}_\varepsilon(u) = \sum_{(x,y)\in\Omega} \sqrt{\varepsilon^2 + \partial_1 u(x,y)^2 + \partial_2 u(x,y)^2}\ .$$

One can see that

$$\nabla \mathrm{TV}_\varepsilon(u) = \nabla^T \left(\frac{\nabla u}{\sqrt{\varepsilon^2 + \|\nabla u\|_2^2}} \right).$$

And one can show that $\nabla \mathrm{TV}_\varepsilon$ is $\frac{8}{\varepsilon}$-Lipschitz.
We can thus minimize F by gradient descent with $\tau < \frac{2}{1+\frac{8\lambda}{\varepsilon}}$.

Denoising Examples

Noisy
PSNR = 19.93

Tychonov denoising
PSNR = 25.89

TV_ε denoising
PSNR = 27.21

Projected Gradient Descent

Imagine that we want to constrain the solution into a convex closed set $C \subset \mathbf{R}^d$:

$$\underset{u \in C}{\text{Argmin}}\, F(u)$$

For that, we can use the orthogonal projection $p_C : \mathbf{R}^d \to C$.

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be convex differentiable such that ∇f is L-Lipschitz.
Let $C \subset \mathbf{R}^d$ be a closed convex set. Assume that $\text{Argmin}_C f$ is non-empty.
For $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$, let (x_n) be defined by

$$x_{n+1} = p_C(x_n - \tau \nabla f(x_n)) \ .$$

Then (x_n) converges to an element of $\text{Argmin}_C f$.

Example : For inpainting, we can deal with the noiseless problem $v = Au$.
In this case, we can perform constrained minimization of only the regularization term:

$$\min_{v = Au} R(u).$$

Plan

Euclidean metrics

- Given two images u and v of size $M \times N$ with graylevels between 0 and 255.
- Denote $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ the pixel domain
- Mean Square Error $\downarrow$:

$$\mathrm{MSE} = \frac{1}{MN} \sum_{\mathbf{x} \in \Omega} (u(\mathbf{x}) - v(\mathbf{x}))^2$$

- Root Mean Square Error $\downarrow$:

$$\mathrm{RMSE} = \left(\frac{1}{MN} \sum_{\mathbf{x} \in \Omega} (u(\mathbf{x}) - v(\mathbf{x}))^2 \right)^{\frac{1}{2}}$$

- Peak Signal to Noise Ratio $\uparrow$:

$$\mathrm{PSNR} = 20 \log_{10} \left(\frac{255}{\mathrm{RMSE}} \right)$$

- Useful for inverse problems such as denoising.
- Not ideal when one hopes to generate new content.

Structural similarity index measure (SSIM ↑) [Wang et al., 2004]

Between patches:

- Given two patches x, y (typically of size 8×8 or 11×11 with a Gaussian windowing)

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \in [-1, 1]$$

with:

- μ_x the pixel sample mean of x
- μ_y the pixel sample mean of y
- σ_x^2 the variance of x
- σ_y^2 the variance of y
- σ_{xy} the covariance of x and y
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator, with the range $L = 255$ or 1 and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

- $\mathrm{SSIM}(x, y)$ is the product of three terms:

Luminance	Contrast	Structure
$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$	$c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$	$s(x, y) = \frac{\sigma_{xy} + c_2/2}{\sigma_x\sigma_y + c_2/2}$

Structural similarity index measure (SSIM $\uparrow$) [Wang et al., 2004]

Between images:

- Given two images u and v of size $M \times N$ with gray-level between 0 and $L = 255$, define the Mean-SSIM by averaging over all patches:

$$\text{(M)SSIM}(u, v) = \text{mean}(\{\text{SSIM}(P_{\mathbf{x}}(u), P_{\mathbf{x}}(v)),\ \mathbf{x} + \omega \subset \Omega\})$$

  where $P_{\mathbf{x}}(u)$ is the restriction of u on the patch $\mathbf{x} + \omega$.
- There are also multiscale variants.
- SSIM is not a distance, its range is $[-1, 1]$.
- SSIM is closer to a perceptual distance, especially regarding local textures.

LPIPS $\downarrow$ [Zhang et al., 2018]

LPIPS: Learned Perceptual Image Patch Similarity

- Previous works on texture synthesis [Gatys et al., 2015] and style transfer [Gatys et al., 2016]
- have shown the importance of the VGG [Simonyan and Zisserman, 2015] features for perceptual similarity between images.
- This means that intermediate features of classification CNN are useful in their own: "**a good feature is a good feature**. Features that are good at semantic tasks are also good at self-supervised and unsupervised tasks, and also provide good models of both human perceptual behavior and macaque neural activity."

LPIPS model: Define a perceptual distance between 64×64 patches by computing a Euclidean norm between features:

$$d(x, x_0)^2 = \sum_{\text{layers } \ell} \frac{1}{H_\ell W_\ell} \sum_{i,j} \| w_\ell \odot (V^\ell(x)_{i,j} - V^\ell(x_0))_{i,j} \|_2^2$$

where for each layer the channel weights w_ℓ are learned to reproduce human evaluation of distortion between patches.

Plan

Generative prior for inverse problems in imaging

- Instead of computing an explicit regularization R, one can add a constraint

$$\min_{x\in\Sigma} \|Ax - y\|^2$$

where $\Sigma \subset \mathbf{R}^d$ is a "low-dimensional" model [Candes et al., 2006, Bourrier et al., 2014].

- Adopting a **generative prior** consists in considering the model

$$\Sigma = \{G(z), z \in \mathbf{R}^k\}$$

parameterized by a pre-trained generative network.

- We then solve the inverse problem by computing

$$\hat{x} = G(\hat{z}) \quad \text{where} \quad \hat{z} \in \operatorname*{Argmin}_{z\in\mathbf{R}^k} \|A(G(z)) - y\|^2.$$

This can be seen as a "pseudo-inverse" with a "manifold constraint".

Relation with GAN inversion

- Generative priors demonstrated to be effective for compressed sensing [Bora et al., 2017].
- Denoising with a generative prior amounts to solving

$$\min_{z \in \mathbf{R}^k} \|G(z) - y\|^2.$$

  This can be reformulated as GAN inversion: finding the latent code z such that $y = G(z)$. GANs are less appropriate for that than VAE or normalizing flows.
- Adopting a generative prior implicitly assumes that the GAN inversion is effective.
- Recovery guarantees can be formulated with hypotheses on A and G [Bora et al., 2017]. In practice, Bora et al. also add a latent regularization $\|z\|^2$.
- But GANs may suffer from mode collapse, or limited generator capacity.
- Only works when the generator is learned on appropriate data (related to the observation).

Deep Image Prior [Ulyanov et al., 2018]

- "Image statistics are implicitly captured by the structure of CNN"
- Fix a random latent code z and "fine-tune" the parameters of the network:

$$\min_{\theta} \|AG_{\theta}(z) - y\|^2$$

- Results highly depend on the chosen architecture for G_{θ}.
  Ulyanov et al. chose a U-Net architecture with skip connections with millions of parameters, and $z, x = G_{\theta}(z)$ with same spatial dimension.
- Convergence guarantee: descent lemma as soon as function has Lipschitz gradient.
- Iterating too much conducts to fit also the noise!
  $\rightarrow$ Regularization by early stopping the optimization algorithm...
- Same technique also used with SinGAN [Shaham et al., 2019] for image editing or restoration.

Deep Image Prior [Ulyanov et al., 2018]

4× super-resolution

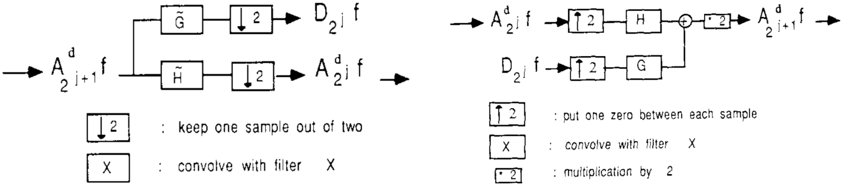
	Baby	Bird	Butterfly	Head	Woman	**Avg.**
No prior	30.16	27.67	19.82	29.98	25.18	26.56
Bicubic	31.78	30.2	22.13	31.34	26.75	28.44
TV prior	31.21	30.43	24.38	31.34	26.93	28.85
Glasner et al.	**32.24**	31.10	22.36	**31.69**	26.85	28.84
Ours	31.49	**31.80**	**26.23**	31.04	**28.93**	**29.89**
LapSRN	33.55	33.76	27.28	32.62	30.72	31.58
SRResNet-MSE	33.66	35.10	28.41	32.73	30.6	32.10

8× super-resolution

	Baby	Bird	Butterfly	Head	Woman	**Avg.**
No prior	26.28	24.03	17.64	27.94	21.37	23.45
Bicubic	27.28	25.28	17.74	28.82	22.74	24.37
TV prior	27.93	25.82	18.40	28.87	23.36	24.87
SelfExSR	**28.45**	26.48	18.80	29.36	24.05	25.42
Ours	28.28	**27.09**	**20.02**	**29.55**	**24.50**	**25.88**
LapSRN	28.88	27.10	19.97	29.76	24.79	26.10

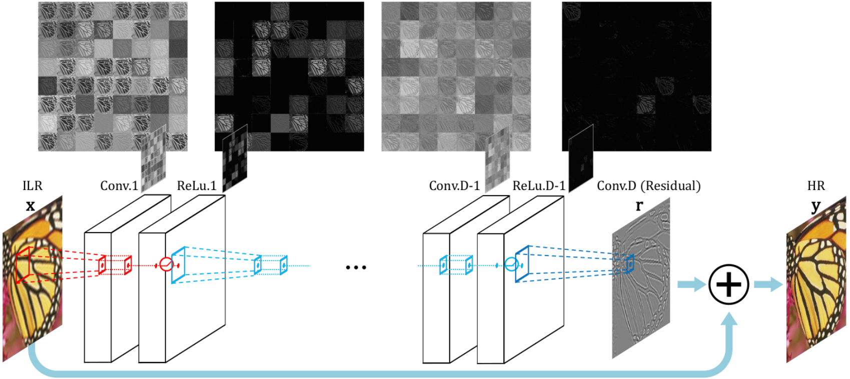
U-Net and Skip Connections

- A U-net can be trained to produce an image aligned with the input image.
- Combine images processed at different scales.
- Skip connections for residual learning [Kim et al., 2016]
- U-nets were used for several imaging tasks:
  - · Segmentation [Ronneberger et al., 2015]
  - · Denoising, inverse problems: [Kim et al., 2016], [Ongie et al., 2020], DRUNet [Zhang et al., 2021]
  - · Image to image translation: Pix2Pix [Isola et al., 2017]
- Multi-resolution combinations already at the core of wavelet processing [Mallat, 1989]...



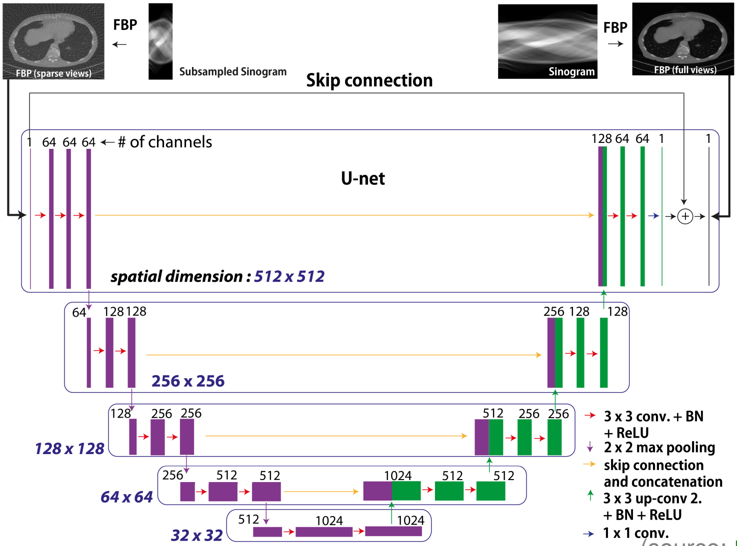
U-Net and Skip Connections

Skip connections: learn $y = x + N(x)$ instead of $y = N(x)$.



(source: [Kim et al., 2016])

U-Net and Skip Connections



(source: [Jin et al., 2017])

Deep Generative Priors

- Acting both on network parameters θ and latent code z improves restoration

$$\min_{z,\theta} \|AG_\theta(z) - y\|^2$$

- ... especially if we adopt a more appropriate loss $\mathcal{L}$

$$\min_{z,\theta} \mathcal{L}(AG_\theta(z), y)$$

- In order to have more photo-realistic results, one can
  · include a term evaluating the quality of deep features (VGG)
  · exploit the discriminator used for training G (with a few adaptations)

(a) Target (b) Zhu *et al.* [37] (c) Bau *et al.* [5] (d) Perceptual loss (e) Discriminator (f) Discriminator + progressive

(source: [Pan et al., 2022])

Super-Resolution with GANs (SRGAN) [Ledig et al., 2017]

Goal: From couples of training images (x_n^{HR}, x_n^{LR}) (high-res, low-res), train a feed-forward network G to predict the HR from LR:

$$\min_{\theta_G} \sum_{n=1}^{N} \mathcal{L}(G_{\theta_G}(x_n^{LR}), x_n^{HR}).$$

Actually, the SRGAN loss has an adversarial formulation, which includes a "content loss":

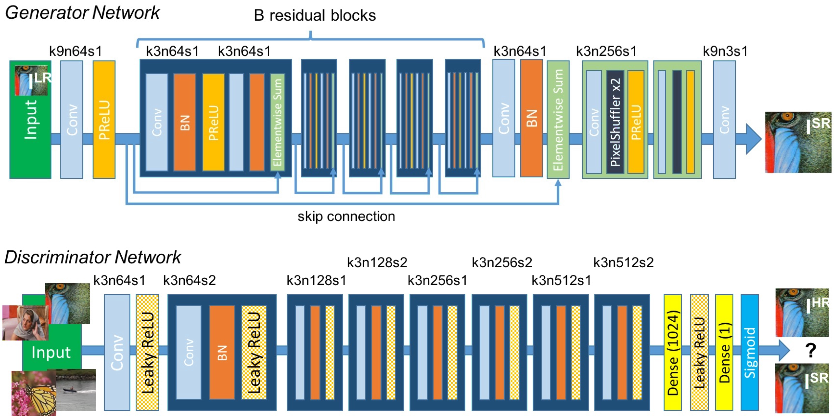
$$\min_{\theta_\mathrm{G}} \max_{\theta_\mathrm{D}} \sum_{n=1}^{N} \log D_{\theta_D}(x_n^{\mathrm{HR}}) + \log(1 - D_{\theta_D}(G_{\theta_G}(x_n^{\mathrm{LR}}))) + \lambda \mathcal{L}_{\mathrm{content}}(G_{\theta_G}(x_n^{\mathrm{LR}}), x_n^{\mathrm{HR}})$$

Content loss between the VGG feature tensors of $x^{SR} = G_{\theta_G}(x^{\mathrm{LR}})$ and x^{HR} at a layer ℓ:

$$\mathcal{L}_{\mathrm{content}}(x^{\mathrm{SR}}, x^{\mathrm{HR}}) = \|\mathrm{VGG}^{\ell}(x^{\mathrm{SR}}) - \mathrm{VGG}^{\ell}(x^{\mathrm{HR}})\|_2^2$$

- Force images to have similar high level feature tensors (closer to perceptual similarity)
- Training by alternating gradient-based updates of θ_G, θ_D.

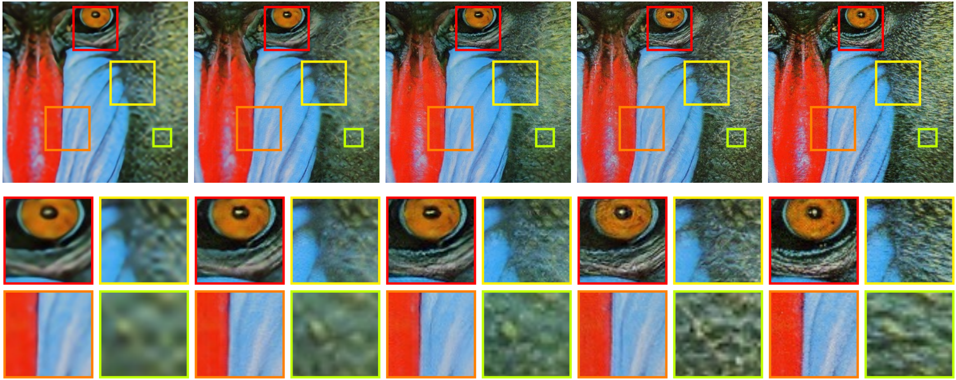
Super-Resolution with GANs (SRGAN) [Ledig et al., 2017]



Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

Super-Resolution with GANs (SRGAN) [Ledig et al., 2017]

SRResNet SRGAN-MSE SRGAN-VGG22 **SRGAN-VGG54** original HR image



×4 upsampling (16× more pixels)

- SRResNet: generator trained only with MSE (no adversarial loss)
- SRGAN-MSE: generator and discriminator with MSE content loss,
- SRGAN-VGG22: generator and discriminator with VGG22 content loss,
- SRGAN-VGG54: generator and discriminator with VGG54 content loss.

Super-Resolution with GANs (SRGAN) [Ledig et al., 2017]

bicubic
(21.59dB/0.6423)

SRResNet
(23.53dB/0.7832)

SRGAN
(21.15dB/0.6868)

original

$\times$4 upsampling (16$\times$ more pixels)

- Even though some details are lost, they are replaced by “fake” but photo-realistic objects
- Of course, SRResNet achieves better PSNR, but is blurrier.

Deep learning techniques for inverse problems in imaging [Ongie et al., 2020]

- One can simply solve $y = Ax + w$ by training a network $\hat{x} = N(y)$...
- This is supervised learning given a training set $\mathcal{D} = \{(x_n, y_n),\ i = 1, \dots, N\}$.
- Many possible architectures: **denoising auto-encoders, U-Nets, unrolled optimization,...**

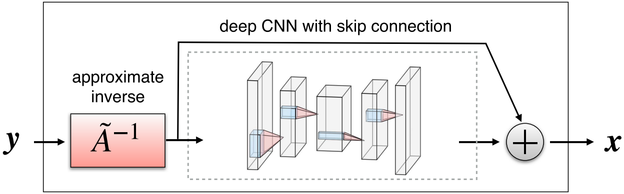
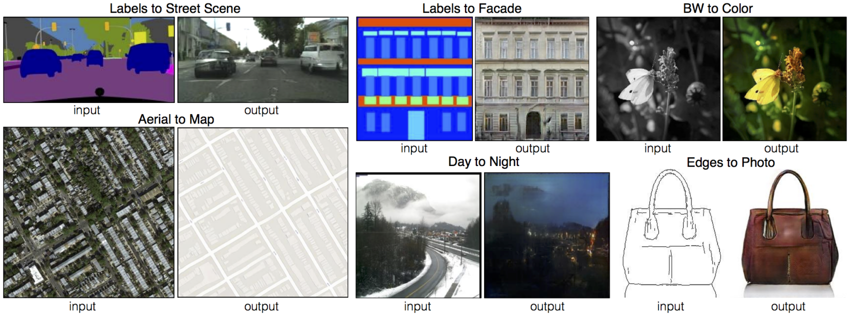


Fig. 7. When an approximate inverse $\tilde{A}^{-1}$ of the forward model is known, a common approach in the supervised setting is to train a deep CNN to remove noise and artifacts from an initial reconstruction obtained by applying $\tilde{A}^{-1}$ to the measurements.

(source: [Ongie et al., 2020])

Image-to-image translation

Pix2pix: Image-to-Image Translation with Conditional Adversarial Nets [Isola et al., 2017]



- Training using a set of image pairs (x_i, y_i)
- GAN conditioned on input image x to produce $y = G(x)$.
- Opens the way for new creative tools

Image-to-image translation

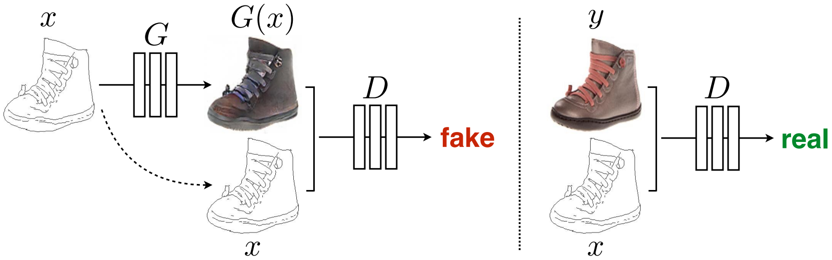


Figure 2: Training a conditional GAN to map edges→photo. The discriminator, D, learns to classify between fake (synthesized by the generator) and real {edge, photo} tuples. The generator, G, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map.

(source: From [Isola et al., 2017])

Conditional GANs

Conditional GANs: Train the generator and the discriminator by passing a class information:

- **Generator:** Generate a fake “3”.
- **Discriminator:** Is it a real or a fake “3”?

Unconditional training:

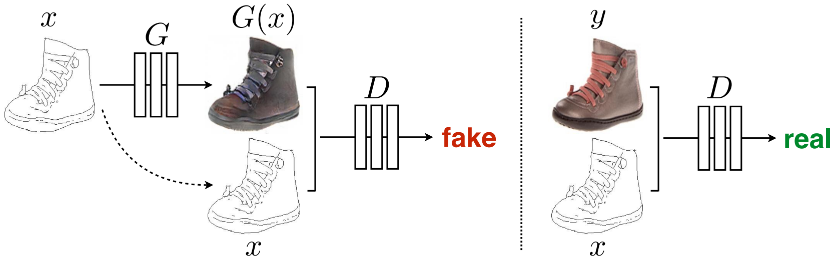
$$\min_{\theta_G} \max_{\theta_D} \sum_{x \in \mathcal{D}_{\text{real}}} \log D_{\theta_D}(x) + \sum_{z \in \mathcal{D}_{\text{rand}}} \log(1 - D_{\theta_D}(\underbrace{G_{\theta_G}(z)}_{\text{fake}}))$$

Class conditional training:

$$\min \theta_G \max \theta_D \sum_{(x,c) \in \mathcal{D}_{\text{real}}} \log D_{\theta_D}(x, c) + \sum_{(z,c) \in \mathcal{D}_{\text{rand}}} \log(1 - D_{\theta_D}(\underbrace{G_{\theta_G}(z, c)}_{\text{fake}}, c))$$

Needs a distribution model for drawing c to generate $G_{\theta_G}(z, c)$.

Conditional GANs: image-to-image translation

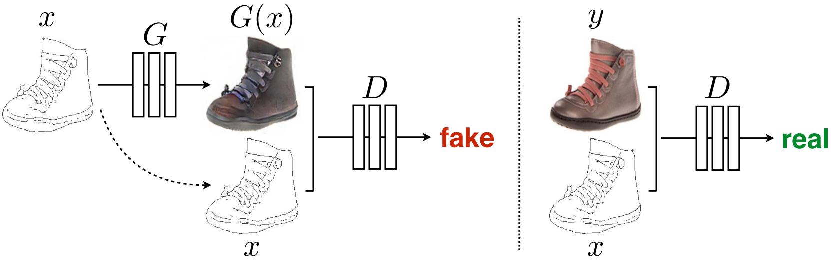


(source: From [Isola et al., 2017])

Architecture details:

- Generator: **U-net architecture**
- Discriminator applied to each 70×70 patch and spatially averaged
- Both are fully convolutional so after training, larger images can be generated
- No latent code z in the generator, but randomness thanks to dropout in the network.

Conditional GANs: image-to-image translation



(source: From [Isola et al., 2017])

Training loss:

$$\min_{\theta_G} \max_{\theta_D} \sum_{(x,y)\in\mathcal{D}} \log D_{\theta_D}(y,x) + \log(1 - D_{\theta_D}(\underbrace{G_{\theta_G}(x)}_{\text{fake}}, x)) + \| \underbrace{G_{\theta_G}(x)}_{\text{fake}} - y\|_1$$

The discriminator looks at generated patches while the ℓ_1 loss is global.

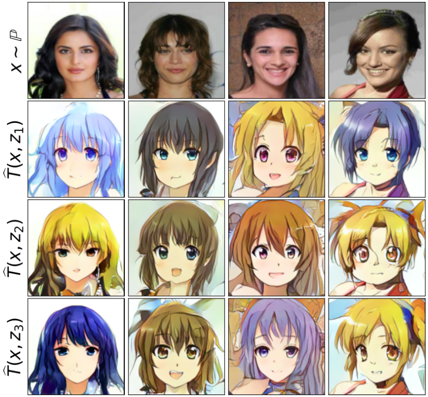
Pix2Pix results

Figure 4: Different losses induce different quality of results. Each column shows results trained under a different loss. Please see `https://phillipi.github.io/pix2pix/` for additional examples.

(source: From [Isola et al., 2017])

"Style transfer" with weak optimal transport [Korotin et al., 2023]

Weak optimal transport allows for style transfer with $y = T(x, z)$ and z truly stochastic.



(source: [Korotin et al., 2023])

Take-home Messages

- We have seen optimization methods for solving imaging inverse problems.
- This can be adapted to use a generative prior (related to GAN inversion).
- Generative priors are useful for tasks where one has access to very few measurements.
- Such deep prior may hallucinate details. Use with care in scientific context.
  Crucial need for uncertainty quantization!
- For explicit regularizations based on deep denoisers, see courses on Plug-and-Play imaging.

THANK YOU FOR YOUR ATTENTION!

References I

- Bora, A., Jalal, A., Price, E., and Dimakis, A. G. (2017).
  Compressed sensing using generative models.
  In *International Conference on Machine Learning*, pages 537–546. PMLR.
- Bourrier, A., Davies, M., Peleg, T., Perez, P., and Gribonval, R. (2014).
  Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems.
  Information Theory, IEEE Transactions on, 60(12):7928–7946.
- Candes, E. J., Romberg, J. K., and Tao, T. (2006).
  Stable signal recovery from incomplete and inaccurate measurements.
  Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59(8):1207–1223.
- Gatys, L. A., Ecker, A. S., and Bethge, M. (2015).
  Texture synthesis using convolutional neural networks.
  In *Advances in Neural Information Processing Systems*, pages 262–270.

References II

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016).
Image style transfer using convolutional neural networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017).
Image-to-image translation with conditional adversarial networks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. (2017).
Deep convolutional neural network for inverse problems in imaging.
IEEE transactions on image processing, 26(9):4509–4522.

Kim, J., Lee, J. K., and Lee, K. M. (2016).
Accurate image super-resolution using very deep convolutional networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654.

Korotin, A., Selikhanovych, D., and Burnaev, E. (2023).
Kernel neural optimal transport.
arXiv preprint arXiv:2205.15269.

References III

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017).
Photo-realistic single image super-resolution using a generative adversarial network.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

Mallat, S. (1989).
A theory for multiresolution signal decomposition: the wavelet representation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674–693.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. (2020).
Deep learning techniques for inverse problems in imaging.
IEEE Journal on Selected Areas in Information Theory, 1(1):39–56.

Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C., and Luo, P. (2022).
Exploiting deep generative prior for versatile image restoration and manipulation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11):7474–7489.

References IV

Ronneberger, O., Fischer, P., and Brox, T. (2015).
U-net: Convolutional networks for biomedical image segmentation.
In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Shaham, T. R., Dekel, T., and Michaeli, T. (2019).
SinGAN: Learning a Generative Model from a Single Natural Image.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580.

Simonyan, K. and Zisserman, A. (2015).
Very deep convolutional networks for large-scale image recognition.
In Bengio, Y. and LeCun, Y., editors, *Proceedings of the International Conference on Learning Representations*.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018).
Deep image prior.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

References V

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004).
Image quality assessment: from error visibility to structural similarity.
IEEE transactions on image processing, 13(4):600–612.

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, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018).
The unreasonable effectiveness of deep features as a perceptual metric.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.