A Generative Variational Model for Inverse Problems in Imaging

Andreas Habring, Martin Holler
[Habring et al. 2021]

Research group on Mathematics of Data Science (MathDS)
Institute of Mathematics and Scientific Computing, University of Graz

September 22, 2021

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Figure: Image reconstruction using (DIP). Inpainting. Left to right: Corrupted, (TGV), (DIP).

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Issues:

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Issues:

- Uses many heuristics in the optimization.

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| T f_\theta(\mu) - y \|_2^2.$$

Issues:

- Uses many heuristics in the optimization.
- Is not well-posed.

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Issues:

- Uses many heuristics in the optimization.
- Is not well-posed.

How can we eliminate these issues?

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| T f_\theta(\mu) - y \|_2^2.$$

Issues:

- Uses many heuristics in the optimization.
- Is not well-posed.

How can we eliminate these issues?

- Combine Deep Image Prior with regularization of the network.

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y \|_2^2.$$

Issues:

- Uses many heuristics in the optimization.
- Is not well-posed.

How can we eliminate these issues?

- Combine Deep Image Prior with regularization of the network.
- Design network in such a way, that it is amenable to analysis in function space.

Motivation

- Inverse problem: Given data $y = Tu + n$, with noise n, determine u.
- **Deep Image Prior (DIP):** [Ulyanov et al. 2018] Untrained CNN $f_\theta(\mu)$, $u = f_\theta(\mu)$ which *solves*

$$\min_\theta \| Tf_\theta(\mu) - y\|_2^2.$$

Issues:

- Uses many heuristics in the optimization.
- Is not well-posed.

How can we eliminate these issues?

- Combine Deep Image Prior with regularization of the network.
- Design network in such a way, that it is amenable to analysis in function space.

At the end, we will arrive at a generative prior $\mathcal{G}$, that is finite only on the range of the network

$$\mathcal{G}(u) = \min_{\mu,\theta} \|\mu\|_1, \quad \text{s.t.} \begin{cases} f_\theta(\mu) = u \\ \|\theta_i\|_2 \leq 1 \end{cases}$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

$$\min_{u,\theta,\mu} \| Tu - y \|_2^2 \quad \text{s.t. } u = f_\theta(\mu)$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

$$\min_{u,\theta,\mu} \| Tu - y \|_2^2 \quad \text{s.t. } u = f_\theta(\mu)$$

$$\mathcal{G}(u) = \begin{cases} 0 & \text{if } \exists \mu, \theta : \ u = f_\theta(\mu) \\ \infty & \text{else.} \end{cases}$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

$$\min_{u,\theta,\mu} \| Tu - y \|_2^2 \quad \text{s.t. } u = f_\theta(\mu) \qquad \mathcal{G}(u) = \begin{cases} 0 & \text{if } \exists \mu, \theta : \ u = f_\theta(\mu) \\ \infty & \text{else.} \end{cases}$$

- Not well-posed. We need regularization of the parameters.

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

$$\min_{u,\theta,\mu} \| Tu - y \|_2^2 \quad \text{s.t. } u = f_\theta(\mu) \qquad \mathcal{G}(u) = \begin{cases} 0 & \text{if } \exists \mu, \theta : \ u = f_\theta(\mu) \\ \infty & \text{else.} \end{cases}$$

- Not well-posed. We need regularization of the parameters.

$$\min_{u,\theta,\mu} \lambda \| Tu - y \|_2^2 + \|\mu\| \quad \text{s.t. } \begin{cases} u = f_\theta(\mu) \\ \|\theta\| \leq 1 \end{cases}$$

Informal Derivation of the Model

- From DIP we want to derive a well-posed model.
- We want to find a generative prior $\mathcal{G}$ such that the method can be written as

$$\min_u \lambda \| Tu - y \|^2 + \mathcal{G}(u).$$

- We derive:

$$\min_{\theta,\mu} \| Tf_\theta(\mu) - y \|_2^2$$

$$\min_{u,\theta,\mu} \| Tu - y \|_2^2 \quad \text{s.t. } u = f_\theta(\mu) \qquad \mathcal{G}(u) = \begin{cases} 0 & \text{if } \exists \mu, \theta : \ u = f_\theta(\mu) \\ \infty & \text{else.} \end{cases}$$

- Not well-posed. We need regularization of the parameters.

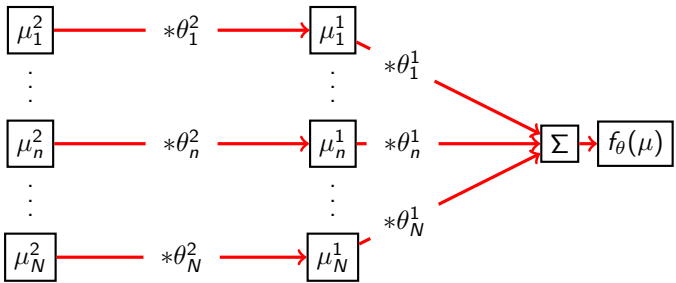
$$\min_{u,\theta,\mu} \lambda \| Tu - y \|_2^2 + \|\mu\| \quad \text{s.t.} \begin{cases} u = f_\theta(\mu) \\ \|\theta\| \leq 1 \end{cases} \qquad \mathcal{G}(u) = \min_{\mu,\theta} \|\mu\| \quad \text{s.t.} \begin{cases} u = f_\theta(\mu) \\ \|\theta\| \leq 1 \end{cases}$$

Network Architecture in Function Space

Generative network:

- L layers of successive convolutions of latent variables $\mu \in \mathcal{M}$ (Radon measures) with filter kernels $\theta \in L^2$ generate output $f_\theta(\mu)$.

$$\mu_n^{\ell-1} = \mu_n^\ell * \theta_n^\ell$$

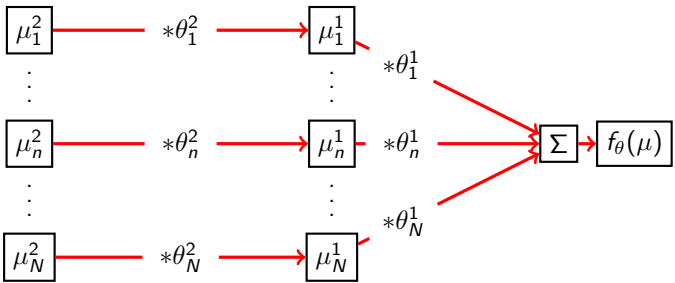


Network Architecture in Function Space

Generative network:

- L layers of successive convolutions of latent variables $\mu \in \mathcal{M}$ (Radon measures) with filter kernels $\theta \in L^2$ generate output $f_\theta(\mu)$.

$$\mu_n^{\ell-1} = \mu_n^\ell * \theta_n^\ell$$



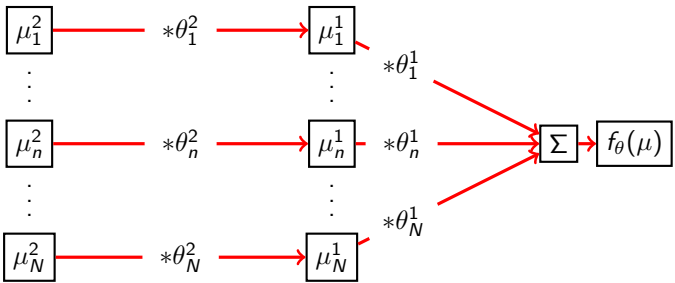
$$\mathcal{G}(u) = \min_{\mu,\theta} \|\mu\|_{\mathcal{M}} \quad \text{s.t.} \quad \begin{cases} u = f_\theta(\mu) \\ \|\theta\|_2 \leq 1 \end{cases}$$

Network Architecture in Function Space

Generative network:

- L layers of successive convolutions of latent variables $\mu \in \mathcal{M}$ (Radon measures) with filter kernels $\theta \in L^2$ generate output $f_\theta(\mu)$.

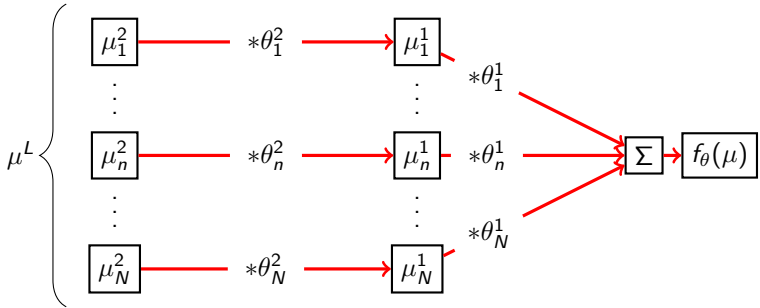
$$\mu_n^{\ell-1} = \mu_n^\ell * \theta_n^\ell$$



$$\mathcal{G}(u) = \min_{\mu,\theta} \|\mu\|_{\mathcal{M}} \quad \text{s.t.} \quad \begin{cases} u = f_\theta(\mu) \\ \|\theta\|_2 \leq 1 \end{cases}$$

Penalize μ on all layers as non-linearity.

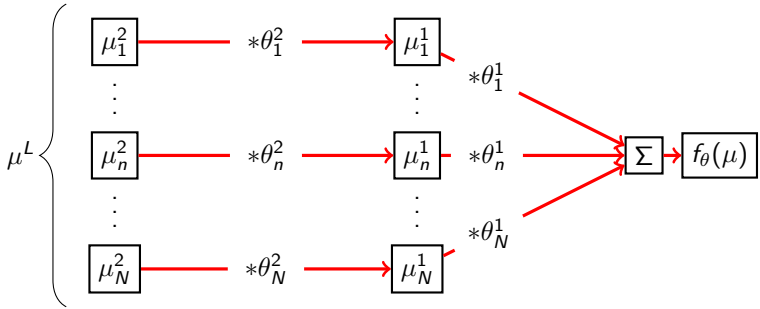
Network Regularity



Theorem

With the proposed network, if the number of layers $L \geq 2$, the input latent variables $\mu^L \in \mathcal{M}$ and $\theta \in L^2$, then $f_\theta(\mu) \in C(\overline{\Omega})$. That is, the network can only generate continuous outputs.

Network Regularity

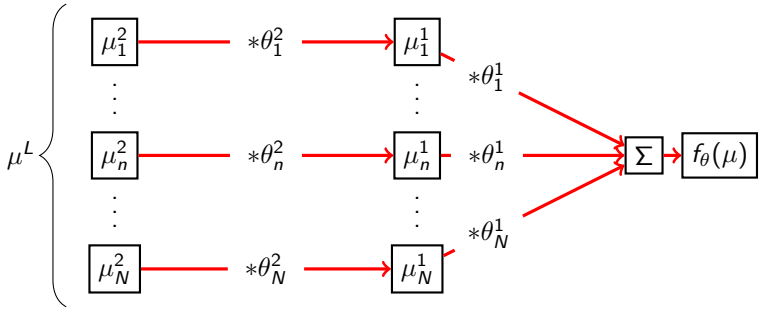


Theorem

With the proposed network, if the number of layers $L \geq 2$, the input latent variables $\mu^L \in \mathcal{M}$ and $\theta \in L^2$, then $f_\theta(\mu) \in C(\overline{\Omega})$. That is, the network can only generate continuous outputs.

- Network suited for denoising.

Network Regularity



Theorem

With the proposed network, if the number of layers $L \geq 2$, the input latent variables $\mu^L \in \mathcal{M}$ and $\theta \in L^2$, then $f_\theta(\mu) \in C(\overline{\Omega})$. That is, the network can only generate continuous outputs.

- Network suited for denoising.
- Network cannot reconstruct sharp edges.

The Proposed Model

Account for discontinuities by combining network with second regularization $\mathcal{R}$ allowing for jump discontinuities, e.g., TV.

image u

$=$

The Proposed Model

Account for discontinuities by combining network with second regularization $\mathcal{R}$ allowing for jump discontinuities, e.g., TV.

image u

$=$

$v = f_\theta(\mu)$

$+$

The Proposed Model

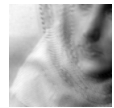
Account for discontinuities by combining network with second regularization $\mathcal{R}$ allowing for jump discontinuities, e.g., TV.

image u

$=$

$v = f_\theta(\mu)$

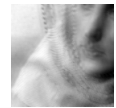
$+$



$u - v$

The Proposed Model

Account for discontinuities by combining network with second regularization $\mathcal{R}$ allowing for jump discontinuities, e.g., TV.


image u $=$ 
$v = f_\theta(\mu)$ $+$ 
$u - v$

(1) $$\min_{u,v} \lambda \mathcal{D}(\mathcal{T}u, y) + s_{\mathcal{G}}(\nu)\mathcal{G}(v) + s_{\mathcal{R}}(\nu)\mathcal{R}(u - v),$$

where

$$\mathcal{G}(v) = \inf_{\mu,\theta} \|\mu\|_{\mathcal{M}}, \quad \text{s.t.} \begin{cases} v & = f_\theta(\mu) \\ \|\theta_n^\ell\|_2 & \leq 1 \\ \int \theta_n^1 \, dx & = 0 \end{cases}$$

Properties of the Generative Prior

$$\mathcal{G}(v) = \inf_{\mu,\theta} \|\mu\|_{\mathcal{M}}, \quad \text{s.t.} \quad \begin{cases} v & = f_\theta(\mu) \\ \|\theta_n^\ell\|_2 & \leq 1 \\ \int \theta_n^1 \, dx & = 0 \end{cases} \tag{2}$$

Theorem (Properties of $\mathcal{G}$)

The generative prior $\mathcal{G} : L^q(\Omega) \to [0, \infty]$ as defined above

- *admits a minimum and*
- *is proper, coercive and weakly lower semi-continuous.*

Properties of the Generative Prior

$$\mathcal{G}(v) = \inf_{\mu,\theta} \|\mu\|_{\mathcal{M}}, \quad \text{s.t.} \quad \begin{cases} v & = f_\theta(\mu) \\ \|\theta_n^\ell\|_2 & \leq 1 \\ \int \theta_n^1 \, dx & = 0 \end{cases} \tag{2}$$

Theorem (Properties of $\mathcal{G}$)

The generative prior $\mathcal{G} : L^q(\Omega) \to [0, \infty]$ as defined above

- *admits a minimum and*
- *is proper, coercive and weakly lower semi-continuous.*

Properties of the Generative Prior

$$\mathcal{G}(v) = \inf_{\mu,\theta} \|\mu\|_{\mathcal{M}}, \quad \text{s.t.} \quad \begin{cases} v & = f_\theta(\mu) \\ \|\theta_n^\ell\|_2 & \leq 1 \\ \int \theta_n^1 \, dx & = 0 \end{cases} \tag{2}$$

Theorem (Properties of $\mathcal{G}$)

The generative prior $\mathcal{G} : L^q(\Omega) \to [0, \infty]$ as defined above

- *admits a minimum and*
- *is proper, coercive and weakly lower semi-continuous.*

Properties of the Generative Prior

$$\mathcal{G}(v) = \inf_{\mu,\theta} \|\mu\|_{\mathcal{M}}, \quad \text{s.t.} \quad \begin{cases} v & = f_\theta(\mu) \\ \|\theta_n^\ell\|_2 & \leq 1 \\ \int \theta_n^1 \, dx & = 0 \end{cases} \tag{2}$$

Theorem (Properties of $\mathcal{G}$)

The generative prior $\mathcal{G} : L^q(\Omega) \to [0, \infty]$ as defined above

- *admits a minimum and*
- *is proper, coercive and weakly lower semi-continuous.*

Main ingredient: Definition and weak*-continuity of the convolution

$$\mathcal{M} \times L^2 \ni (\mu, \theta) \mapsto \mu * \theta.$$

Well-posedness

$$\min_{u,v \in L^q(\Omega)} \lambda \mathcal{D}(Tu, y) + s_{\mathcal{R}}(\nu)\mathcal{R}(u - v) + s_{\mathcal{G}}(\nu)\mathcal{G}(v) \tag{3}$$

Theorem (Existence/stability/convergence)

Assume that $q \in (1, 2]$, Y Banach, $T \in \mathcal{L}(L^q(\Omega), Y)$, $\mathcal{D}(z, y) = \frac{1}{2}\|z - y\|_Y^2$ and $\mathcal{R} = \mathrm{TV}$. Then we have the following:

- *(Existence) Problem (3) admits a solution.*
- *(Stability) The solution is weakly, subsequentially stable up to shifts of u that are const. and in $\ker(T)$.*
- *(Convergence) Let $y_n \to y$, and $(u_n, v_n)_n$ be corresponding solutions with appropriate λ_n, then up to shifts of u_n that are const. and in $\ker(T)$, $(u_n, v_n)_n$ admits a weak accumulation point that solves $Tu = y$.*

This result can be proven more generally.

Well-posedness

$$\min_{u,v \in L^q(\Omega)} \lambda \mathcal{D}(Tu, y) + s_{\mathcal{R}}(\nu)\mathcal{R}(u - v) + s_{\mathcal{G}}(\nu)\mathcal{G}(v) \tag{3}$$

Theorem (Existence/stability/convergence)

Assume that $q \in (1, 2]$, Y Banach, $T \in \mathcal{L}(L^q(\Omega), Y)$, $\mathcal{D}(z, y) = \frac{1}{2}\|z - y\|_Y^2$ and $\mathcal{R} = \mathrm{TV}$. Then we have the following:

- *(Existence) Problem (3) admits a solution.*
- *(Stability) The solution is weakly, subsequentially stable up to shifts of u that are const. and in $\ker(T)$.*
- *(Convergence) Let $y_n \to y$, and $(u_n, v_n)_n$ be corresponding solutions with appropriate λ_n, then up to shifts of u_n that are const. and in $\ker(T)$, $(u_n, v_n)_n$ admits a weak accumulation point that solves $Tu = y$.*

This result can be proven more generally.

Well-posedness

$$\min_{u,v \in L^q(\Omega)} \lambda \mathcal{D}(Tu, y) + s_{\mathcal{R}}(\nu)\mathcal{R}(u - v) + s_{\mathcal{G}}(\nu)\mathcal{G}(v) \tag{3}$$

Theorem (Existence/stability/convergence)

Assume that $q \in (1, 2]$, Y Banach, $T \in \mathcal{L}(L^q(\Omega), Y)$, $\mathcal{D}(z, y) = \frac{1}{2}\|z - y\|_Y^2$ and $\mathcal{R} = \mathrm{TV}$. Then we have the following:

- *(Existence) Problem (3) admits a solution.*
- *(Stability) The solution is weakly, subsequentially stable up to shifts of u that are const. and in $\ker(T)$.*
- *(Convergence) Let $y_n \to y$, and $(u_n, v_n)_n$ be corresponding solutions with appropriate λ_n, then up to shifts of u_n that are const. and in $\ker(T)$, $(u_n, v_n)_n$ admits a weak accumulation point that solves $Tu = y$.*

This result can be proven more generally.

Well-posedness

$$\min_{u,v \in L^q(\Omega)} \lambda \mathcal{D}(Tu, y) + s_{\mathcal{R}}(\nu)\mathcal{R}(u - v) + s_{\mathcal{G}}(\nu)\mathcal{G}(v) \tag{3}$$

Theorem (Existence/stability/convergence)

Assume that $q \in (1, 2]$, Y Banach, $T \in \mathcal{L}(L^q(\Omega), Y)$, $\mathcal{D}(z, y) = \frac{1}{2}\|z - y\|_Y^2$ and $\mathcal{R} = \mathrm{TV}$. Then we have the following:

- *(Existence) Problem (3) admits a solution.*
- *(Stability) The solution is weakly, subsequentially stable up to shifts of u that are const. and in $\ker(T)$.*
- *(Convergence) Let $y_n \to y$, and $(u_n, v_n)_n$ be corresponding solutions with appropriate λ_n, then up to shifts of u_n that are const. and in $\ker(T)$, $(u_n, v_n)_n$ admits a weak accumulation point that solves $Tu = y$.*

This result can be proven more generally.

Experimental Results

- We used iPALM algorithm [Pock et al. 2016] and implemented the method in Python for GPUs based on PyOpenCL[Klöckner et al. 2012].

Experimental Results

- We used iPALM algorithm [Pock et al. 2016] and implemented the method in Python for GPUs based on PyOpenCL[Klöckner et al. 2012].
- We compare our method to:

Experimental Results

- We used iPALM algorithm [Pock et al. 2016] and implemented the method in Python for GPUs based on PyOpenCL[Klöckner et al. 2012].
- We compare our method to:
  - TGV regularization as a classical method.

Experimental Results

- We used iPALM algorithm [Pock et al. 2016] and implemented the method in Python for GPUs based on PyOpenCL[Klöckner et al. 2012].
- We compare our method to:
  - TGV regularization as a classical method.
  - Convex learning (CL) [Chambolle et al. 2020], which can be seen as a single layer version of the proposed method (only one convolutional layer).

Experimental Results

- We used iPALM algorithm [Pock et al. 2016] and implemented the method in Python for GPUs based on PyOpenCL[Klöckner et al. 2012].
- We compare our method to:
  - TGV regularization as a classical method.
  - Convex learning (CL) [Chambolle et al. 2020], which can be seen as a single layer version of the proposed method (only one convolutional layer).
  - The deep image prior (DIP) [Ulyanov et al. 2018], which is a state of the art convolutional neural network method.

Experimental Results - Inpainting

- We used images from the ImageNet ILSVRC2017 DET test dataset [Russakovsky et al. 2015].
- 100 random images, 26 handpicked texture images.
- Cropped and converted to grayscale.
- Inpainting from 30% known pixels.

		TGV	*DIP*	*proposed*
PSNR	Random images	24.58 ± 3.69	25.19 ± 4.08	$\mathbf{25.27} \pm 3.93$
	Texture images	22.04 ± 4.08	$\mathbf{23.94} \pm 4.78$	23.71 ± 4.28
SSIM	Random images	0.81 ± 0.081	0.83 ± 0.092	$\mathbf{0.84} \pm 0.078$
	Texture images	0.76 ± 0.075	$\mathbf{0.83} \pm 0.076$	$\mathbf{0.83} \pm 0.063$

Table: PSNR and SSIM values for inpainting as *mean* $\pm$ *standard deviation*. Bold indicates the best result.

Experimental Results - Inpainting

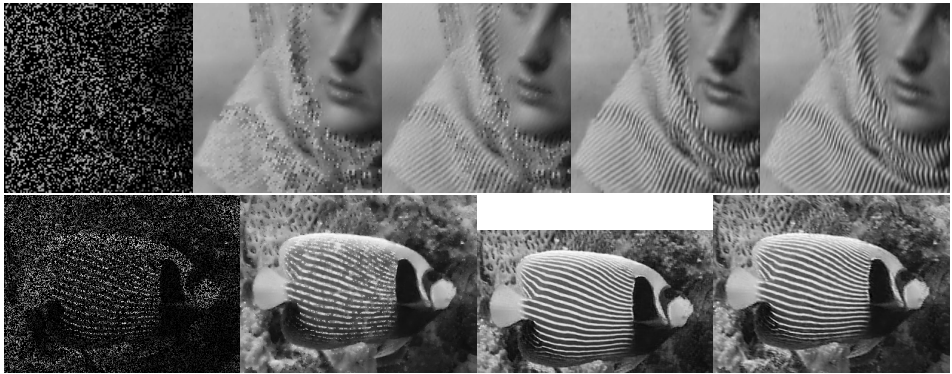


Figure: Inpainting with 30% known pixels. Left to right: Corrupted, (TGV), (CL), (DIP), proposed method, second row without (CL). *Fish* image: "Pomocanthus imperator facing right", by Albert kok, licensed under CC BY-SA 4.0)

Experimental Results - Denoising

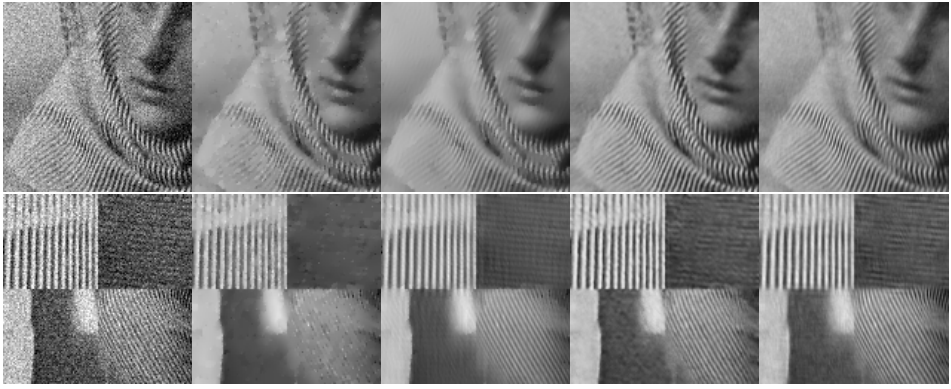


Figure: Denoising with Gaussian noise, standard deviation 0.1 times image range. Left to right: Corrupted, (TGV), (CL), (DIP), proposed method.

Experimental Results - Deconvolution

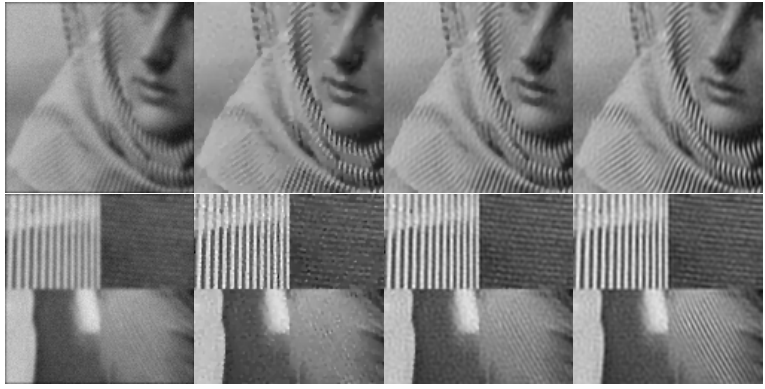


Figure: Deconvolution under noise. Left to right: Corrupted, (TGV), (CL), proposed method.

Experimental Results - JPEG Decompression

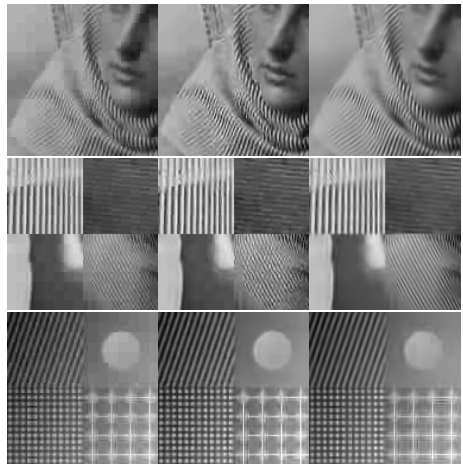


Figure: jpeg-decompression. Left to right: corrupted, (DIP), proposed.

Summary:

We have developed a neural network based regularization method that

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,
- uses a very simple architecture (no skip connections etc.) and very few parameters (factor 100 less than DIP, or factor 10 less counting auxiliary variables) and

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,
- uses a very simple architecture (no skip connections etc.) and very few parameters (factor 100 less than DIP, or factor 10 less counting auxiliary variables) and
- evidentially converges without the use of heuristics.

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,
- uses a very simple architecture (no skip connections etc.) and very few parameters (factor 100 less than DIP, or factor 10 less counting auxiliary variables) and
- evidentially converges without the use of heuristics.

Future work:

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,
- uses a very simple architecture (no skip connections etc.) and very few parameters (factor 100 less than DIP, or factor 10 less counting auxiliary variables) and
- evidentially converges without the use of heuristics.

Future work:

- Convex relaxation, structural properties of solutions, optimality conditions.

Summary:

We have developed a neural network based regularization method that

- delivers state of the art results (compared to DIP),
- comes with theoretical guarantees in the continuous setting,
- uses a very simple architecture (no skip connections etc.) and very few parameters (factor 100 less than DIP, or factor 10 less counting auxiliary variables) and
- evidentially converges without the use of heuristics.

Future work:

- Convex relaxation, structural properties of solutions, optimality conditions.
- Improved model: No splitting, different penalties/function spaces to account for sharp edges already with network.

References

- A. Chambolle et al. “A Convex Variational Model for Learning Convolutional Image Atoms from Incomplete Data”. In: *Journal of Mathematical Imaging and Vision* 62.3 (2020), pp. 417–444.
- Andreas Habring et al. *A Generative Variational Model for Inverse Problems in Imaging*. 2021.
- Andreas Klöckner et al. “PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation”. In: *Parallel Computing* 38.3 (2012), pp. 157–174.
- Thomas Pock et al. “Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex and Nonsmooth Problems”. In: *SIAM Journal on Imaging Sciences* 9.4 (2016), 1756–1787.
- Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.
- D. Ulyanov et al. “Deep Image Prior”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9446–9454.