Trends in mathematics of information: Deep learning, artificial intelligence and compressed sensing

Anders C. Hansen (Cambridge and UiO)
Vegard Antun (UiO)

Joint work with:

B. Adcock (SFU) A. Bastounis (Cambridge)
B. Roman (Cambridge) C. Poon (Cambridge)
V. Vlacic (ETH)

Oslo, May 2018

Layout of the Talks

Layout of the talks

Day I : Deep learning in classification, artificial intelligence, computational issues

Day II : Compressed sensing and deep learning in inverse problems

Day III : Practical tutorial, convolutional networks, DeepFool, instabilities, implementation of compressed sensing and deep learning in inverse problems

Sneak peak from the book "Structured compressed sensing, imaging and learning" (Cambridge University Press (2019))

Introduction to deep learning for classification

Given a classification function $f : \mathbb{R}^d \to \{0, 1\}$, find an approximation $\tilde{f} : \mathbb{R}^d \to \{0, 1\}$ to f.

Construct $\tilde{f}$ based on a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$ for which we know $f(x^j)$ for $j = 1, \ldots, r$.

Test $\tilde{f}$ on a classification (or a test) set $\mathcal{C} = \{y^1, \ldots, y^s\}$. Success is measure by

$$\frac{|\{y^j \in \mathcal{C} \mid f(y^j) = \tilde{f}(y^j)\}|}{s}$$

Best performances of deep learning on image classification is about $3 - 5\%$ failure rate. This is often referred to as super human performance.

Neural networks

Let $\mathcal{NN}_{\mathbf{N},L,d}$, with $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0 = d)$ denote the set of all L-layer neural networks. That is, all mappings $\phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ of the form

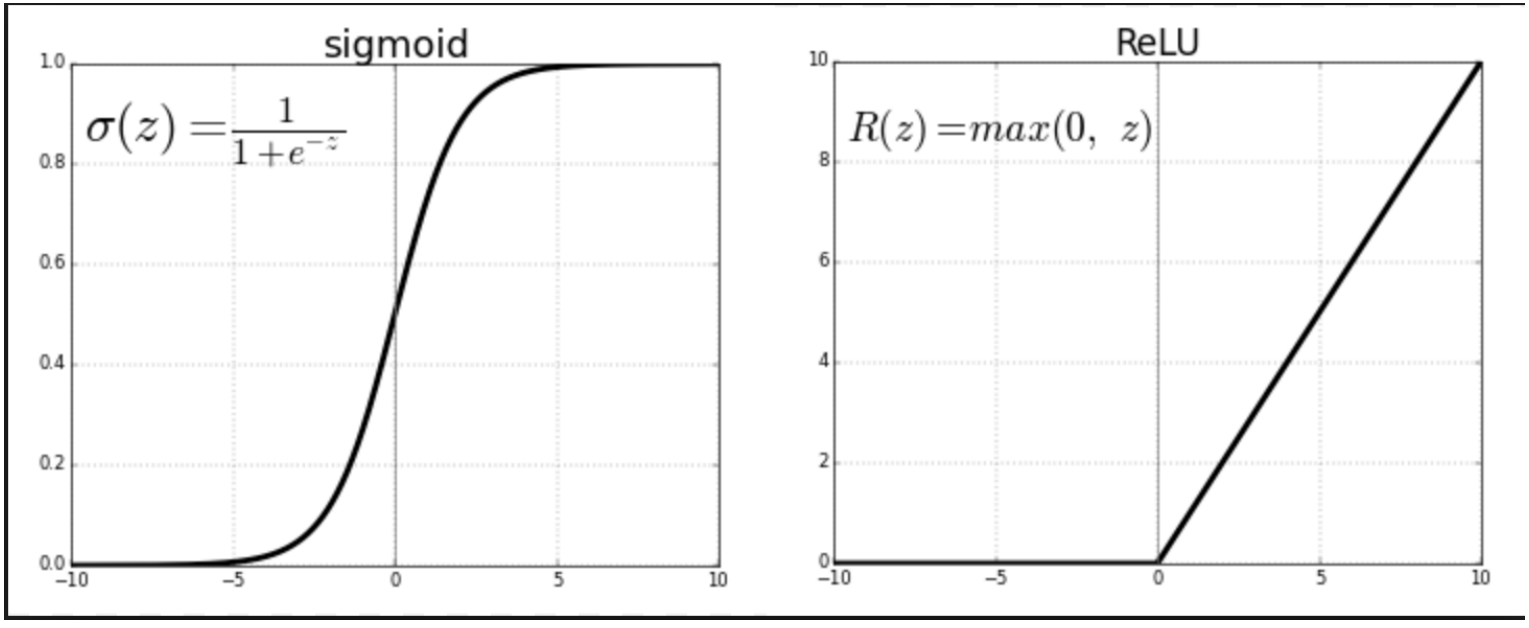
$$\phi(x) = W_L(\rho(W_{L-1}(\rho(\ldots \rho(W_1(x)))))), \quad x \in \mathbb{R}^d.$$

$$W_j y = A_j y - b_j, \qquad A_j \in \mathbb{R}^{N_j \times N_{j-1}}, \quad b_j \in \mathbb{R}^{N_j}$$

$$\rho : \mathbb{R} \to \mathbb{R}$$

is some non-linear function that acts pointwise on a vector.

Choices of ρ



Approximation qualities of neural nets

The universal approximation theorem:

Theorem 1 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$. Then the set of neural networks is dense in $C(\mathbb{R}^d)$ in the topology of uniform convergence on compact sets, if and only if ρ is not a polynomial.

The interpolation theorem:

Theorem 2 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$ and assume that ρ is not a polynomial. For any k distinct points $\{x_j\}_{j=1}^k \subset \mathbb{R}^d$ and associated data $\{\alpha_j\}_{j=1}^k \subset \mathbb{R}$. Then there exists a neural network ϕ such that

$$\phi(x_j) = \alpha_j, \qquad j = 1, \ldots, k.$$

Training neural nets

Given a classification function $f : \mathbb{R}^d \to \{0, 1\}$, a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$, a classification set $\mathcal{C} = \{y^1, \ldots, y^s\}$, and a cost function $C : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}_+$, compute

$$\phi \in \underset{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L,d}}{\operatorname{argmin}} \; C(v, w),$$

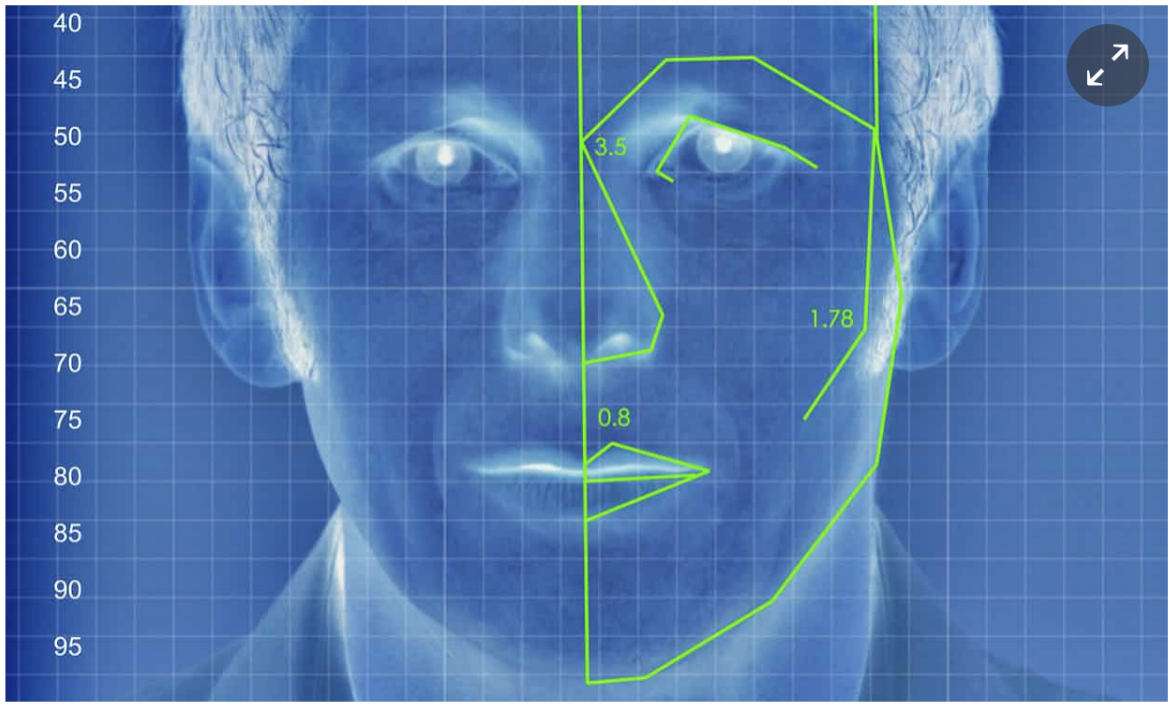
with

$$v = \{\tilde{\phi}(x^j)\}_{j=1}^r, \quad w = \{f(x^j)\}_{j=1}^r.$$

Typical choice is $C(v, w) = \|v - w\|_p^p$.

New AI can guess whether you're gay or straight from a photograph

An algorithm deduced the sexuality of people on a dating site with up to 91% accuracy, raising tricky ethical questions



An illustrated depiction of facial analysis technology similar to that used in the experiment. Illustration: Alamy

Artificial intelligence can accurately guess whether people are gay or straight based on photos of their faces, according to new research that suggests machines can have significantly better “gaydar” than humans.

The study from Stanford University - which found that a computer algorithm

Key questions about deep learning

- Does it work?
- Have we achieved artificial intelligence?
- Do we know what we are doing?
- Does the computational optmization problem for training neural nets make mathematical sense?

Have we achieved artificial intelligence?

Smale's 18th problem

What are the limits of intelligence, both artificial and human?

Smale's 18th problem

Smale:

"Penrose (1991) attempts to show some limitations of artificial intelligence. Involved in his argumentation is the interesting question, is the Mandelbrot set decidable? (see problem 14) and implications of the Gödel incompleteness theorem. However a broader study is called for, one which involves deeper models of the brain, and of the computer, in a search of what artificial and human intelligence have in common, and how they differ."

"Finally problem solving as exemplified by Turing and real number machines is only part of the story of intelligence. Continual interaction with the environment must be incorporated into a good model. Learning is a part of human intelligent activity. The corresponding mathematics is suggested by the theory of repeated games, neural nets and genetic algorithms."

Turing's imitation game

Turing:

"I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words."

Turing's imitation game

Turing:

"The new form of the problem can be described in terms of a game which we call the 'imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart front the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A." The interrogator is allowed to put questions to A and B."

" We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?""

The imitation game and deep learning

What would be a key feature that humans have when classifying images that computers may not have?

The imitation game and deep learning

What would be a key feature that humans have when classifying images that computers may not have?

Answer: Stability

The First Paradox (stability)

There is an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0, 1\}$ such that for any neural network dimensions $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ with $N_0, L \geq 2$ and any $0 < \epsilon < 1/(K + M)$ where M is arbitrarily large and $K \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$ we have the following. There exist uncountably many training sets $\mathcal{T} = \{x^1, \ldots, x^K\}$ and uncountably many classification sets $\mathcal{C} = \{y^1, \ldots, y^M\}$ such that there is a

$$\tilde{\phi} \in \underset{\phi \in \mathcal{NN}_{\mathbf{N},L}}{\operatorname{argmin}} C(v, w), \quad v_j = \phi(x^j), \quad w_j = f(x^j),$$

where $1 \leq j \leq K$ such that

$$\tilde{\phi}(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{C}.$$

However, there exists uncountably many $v \in \mathbb{R}^{N_0}$ such that

$$|\tilde{\phi}(v) - f(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}.$$

Moreover, there is a neural network $\hat{\phi}$, not necessarily trained, such that

$$\hat{\phi}(x) = f(x) \quad \forall x \in \mathcal{B}_\epsilon^\infty(\mathcal{T} \cup \mathcal{C}).$$

Intriguing properties of neural networks

Christian Szegedy
Google Inc.

Wojciech Zaremba
New York University

Ilya Sutskever
Google Inc.

Joan Bruna
New York University

Dumitru Erhan
Google Inc.

Ian Goodfellow
University of Montreal

Rob Fergus
New York University
Facebook Inc.

Abstract

Deep neural networks are highly expressive models that have recently achieved state of the art performance on speech and visual recognition tasks. While their expressiveness is the reason they succeed, it also causes them to learn uninterpretable solutions that could have counter-intuitive properties. In this paper we report two such properties.

First, we find that there is no distinction between individual high level units and random linear combinations of high level units, according to various methods of unit analysis. It suggests that it is the space, rather than the individual units, that contains of the semantic information in the high layers of neural networks.

Second, we find that deep neural networks learn input-output mappings that are fairly discontinuous to a significant extend. Specifically, we find that we can cause the network to misclassify an image by applying a certain imperceptible perturbation, which is found by maximizing the network's prediction error. In addition, the specific nature of these perturbations is not a random artifact of learning: the same perturbation can cause a different network, that was trained on a different

The Paradoxes in Practice: Deep Fool

Deep Fool was established at EPFL in order to study the stability of neural networks.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard

The Robustness of Deep Networks

A geometrical perspective

The Paradoxes in Practice: Deep Fool

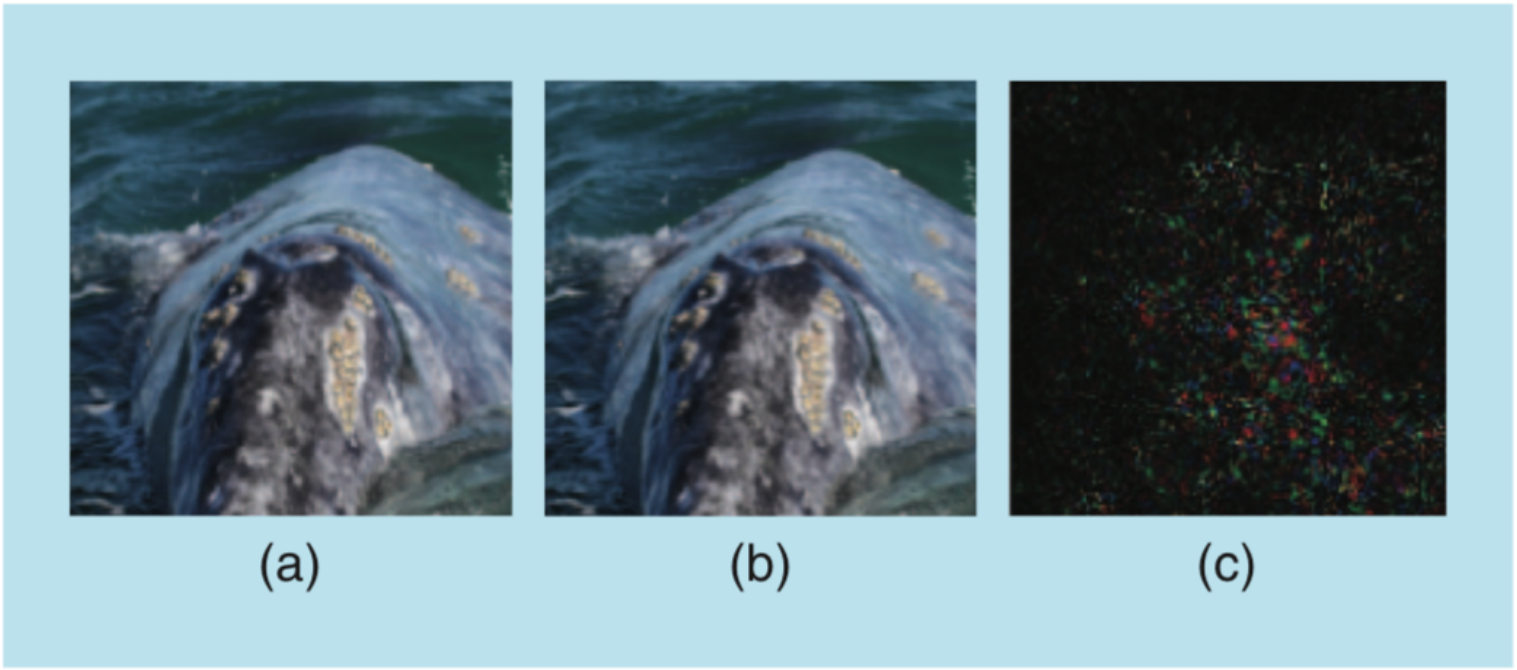


FIGURE 1. An example of an adversarial perturbations in state-of-the-art neural networks. (a) The original image that is classified as a "whale," (b) the perturbed image classified as a "turtle," and (c) the corresponding adversarial perturbation that has been added to the original image to fool a state-of-the-art image classifier [5].

Deep Fool: Universal perturbations

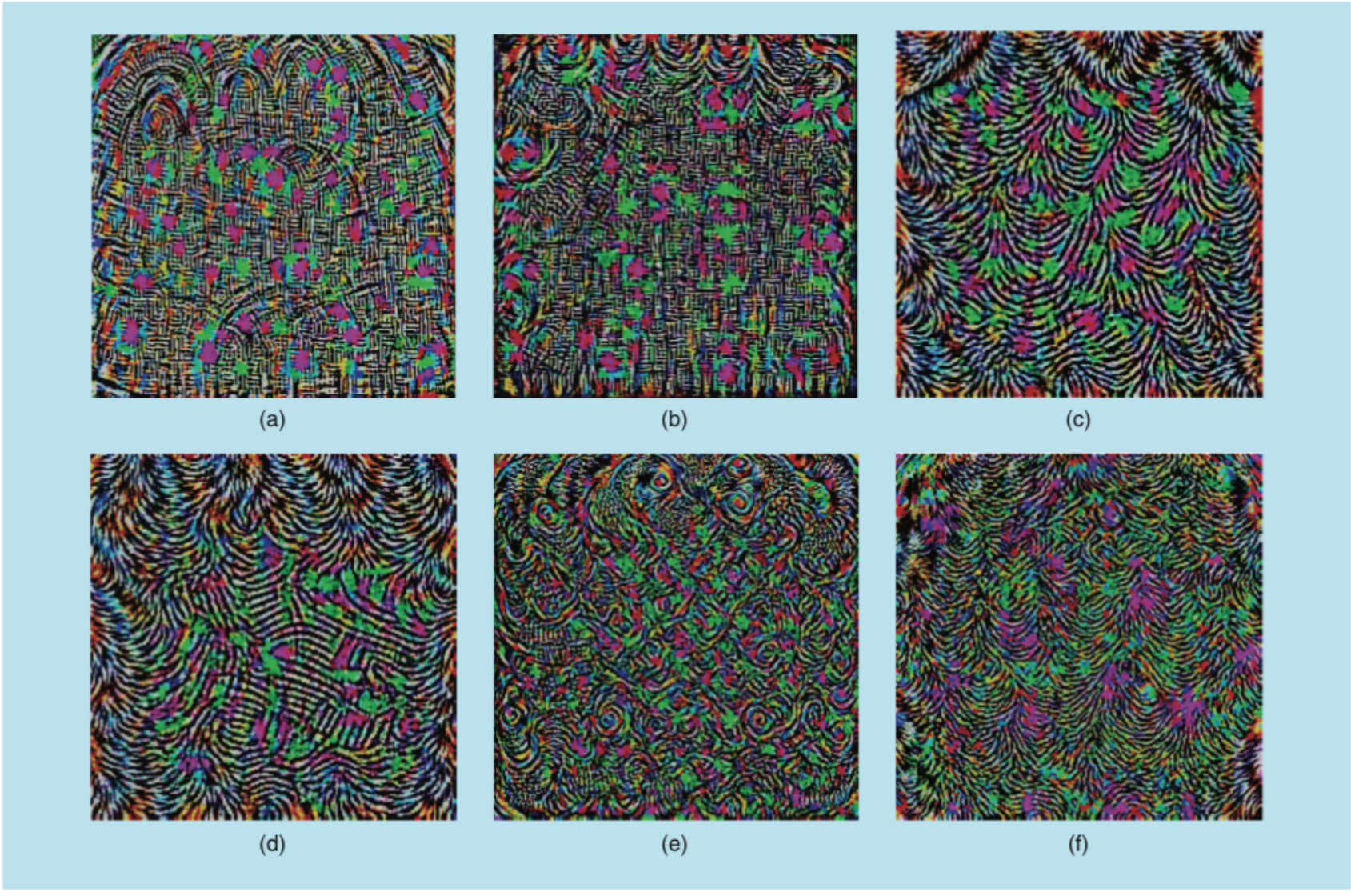


FIGURE 3. Universal perturbations computed for different deep neural network architectures. The pixel values are scaled for visibility. (a) CaffeNet, (b) VGG-F, (c) VGG-16, (d) VGG-19, (e) GoogLeNet, and (f) ResNet-152.

Deep Fool: Examples

FIGURE 4. Examples of natural images perturbed with the universal perturbation and their corresponding estimated labels with GoogLeNet. (a)–(h) Images belonging to the ILSVRC 2012 validation set. (i)–(l) Personal images captured by a mobile phone camera. (Figure used courtesy of [22].)

Deep Fool: Examples

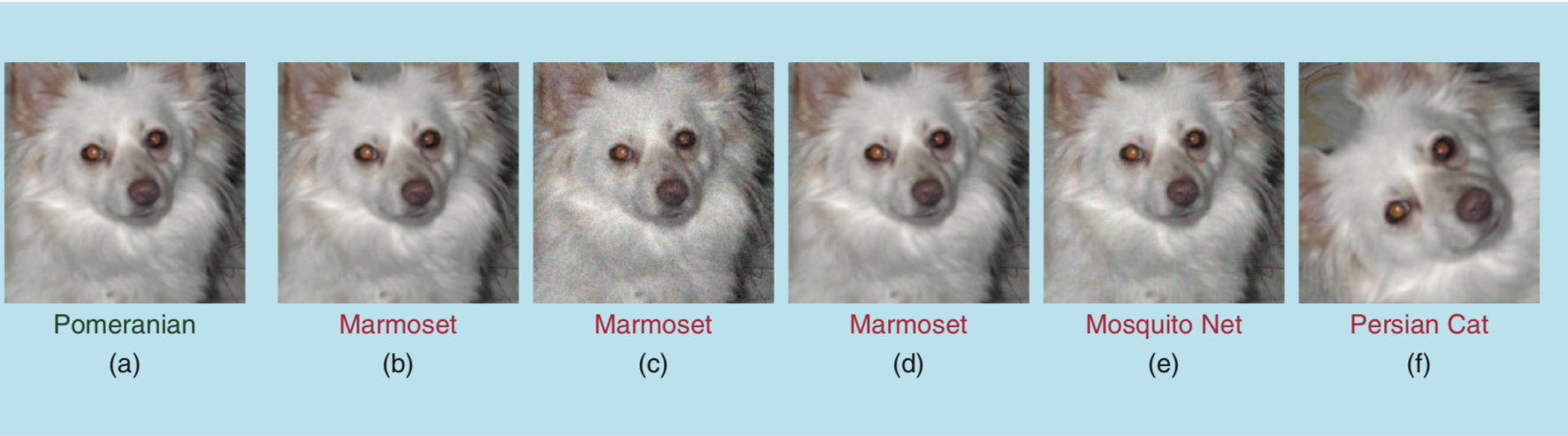


FIGURE 5. (a) The original image. The remaining images are minimally perturbed images (along with the corresponding estimated label) that misclassify the CaffeNet deep neural network. (b) Adversarial perturbation, (c) random noise, (d) semirandom noise with $m = 1{,}000$, (e) universal perturbation, (f) affine transformation. (Figure used courtesy of [17].)

Consequences of instabilities of deep learning

Practical consequences:

- Surveillance
- Security
- Driverless cars

Philosophical consequences:

- Can this be fixed?
- Legal implications
- FDA approval of medical equipment?

Consequences of instabilities of deep learning

Roboter og kunstig intelligens (KI) er allerede i omfattende bruk i helsevesenet. Men det er langt igjen til at legene kan erstattes. Foto: Jeff Pachoud/AFP photo/NTB scanpix

Nyheter Helse

Kunstig intelligens: Forsøk stoppet - foreslo livsfarlig medisin

Dagens Næringsliv
Publisert: 23.10.2017 – 08:45 Oppdatert: 23.10.2017 – 09:03

Uber's self-driving car saw the pedestrian but didn't swerve – report

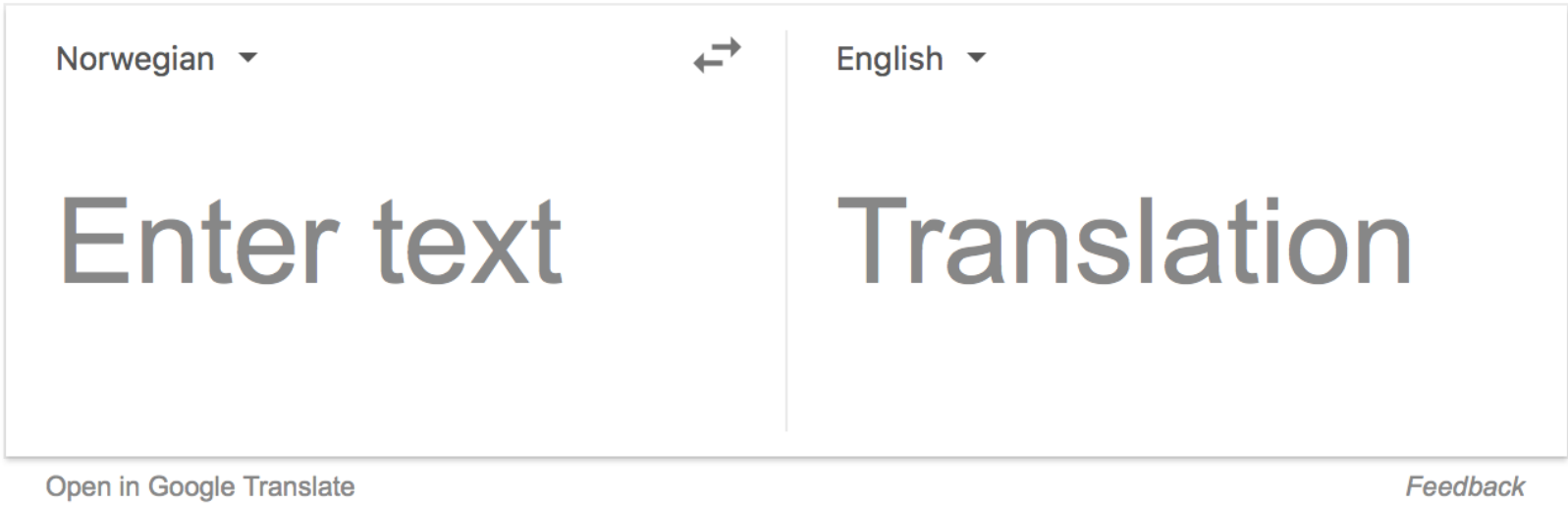
Tuning of car's software to avoid false positives blamed, as US National Transportation Safety Board investigation continues

▲ Uber's modified Volvo XC90 SUV detected but did not react to the crossing pedestrian in first self-driving car fatality, report says. Photograph: Volvo

An Uber self-driving test car which killed a woman crossing the street detected her but decided not to react immediately, a report has said.

The car was travelling at 40mph (64km/h) in self-driving mode when it collided with 49-year-old Elaine Herzberg at about 10pm on 18 March. Herzberg was pushing a bicycle across the road outside of a crossing. She later died from her injuries.

Consequences of instabilities of deep learning



Key questions about deep learning

- Does it work? Yes, except that it becomes universally unstable.
- Have we reached artificial intelligence? No, we are still very far.
- Do we know what we are doing? Not really.
- Can the instability issue be fixed?
- Does the computational optmization problem for training neural nets make mathematical sense?

Can the instability issue be fixed?

There is an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0, 1\}$ such that for any neural network dimensions $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ with $N_0, L \geq 2$ and any $0 < \epsilon < 1/(K + M)$ where M is arbitrarily large and $K \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$ we have the following. There exist uncountably many training sets $\mathcal{T} = \{x^1, \ldots, x^K\}$ and uncountably many classification sets $\mathcal{C} = \{y^1, \ldots, y^M\}$ such that there is a

$$\tilde{\phi} \in \underset{\phi \in \mathcal{NN}_{\mathbf{N},L}}{\operatorname{argmin}} C(v, w), \quad v_j = \phi(x^j), \quad w_j = f(x^j),$$

where $1 \leq j \leq K$ such that

$$\tilde{\phi}(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{C}.$$

However, there exists uncountably many $v \in \mathbb{R}^{N_0}$ such that

$$|\tilde{\phi}(v) - f(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}.$$

Moreover, there is a neural network $\hat{\phi}$, not necessarily trained, such that

$$\hat{\phi}(x) = f(x) \quad \forall x \in \mathcal{B}_\epsilon^\infty(\mathcal{T} \cup \mathcal{C}).$$

Key questions about deep learning

- Does it work? Yes, except that it becomes universally unstable.
- Have we reached artificial intelligence? No, we are still very far.
- Do we know what we are doing? Not really.
- Can the instability issue be fixed? Potentially, but it is unlikely that we can have a fixed architecture of the neural network.
- Does the computational optmization problem for training neural nets make mathematical sense?

Does the computational optmization problem for training neural nets make mathematical sense?

The key computational problem (35)

- Neural Networks

  Given a classification function $f : \mathbb{R}^d \to \{0, 1\}$, a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$, a classification set $\mathcal{C} = \{y^1, \ldots, y^s\}$, and a cost function $C : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}_+$, compute

$$\{\phi(y^j)\}_{j=1}^s, \quad \phi \in \underset{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L,d}}{\operatorname{argmin}} \; C(v, w),$$

  with

$$v = \{\tilde{\phi}(x^j)\}_{j=1}^r, \quad w = \{f(x^j)\}_{j=1}^r,$$

  where, $\mathcal{NN}_{\mathbf{N},L,d}$, with $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0 = d)$ denotes the set of all L-layer neural networks. That is, all mappings $\phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ of the form

$$\phi(x) = W_L(\rho(W_{L-1}(\rho(\ldots \rho(W_1(x)))))), \quad x \in \mathbb{R}^d.$$

- Hilbert's question on the existence of algorithms for decision problems, solved by Turing in

  On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. London Math. Soc.* (1936)

- Smale's question on the existence of purely iterative generally convergent algorithms for polynomial root-finding, solved by McMullen and Doyle & McMullen in

  Families of rational maps and iterative root-finding algorithms *Annals of Math.* (1987) and Solving the quintic by iteration, *Acta Math.* (1989).

The basic question

Do algorithms exist for our problems?

Smale writes in the list of problems for the 21st century:

> *"But real number computations and algorithms which work only in exact arithmetic can offer only limited understanding. Models which process approximate inputs and which permit round-off computations are called for."*

We must be able to handle inaccurate input as $\sqrt{2}$, $\cos(3)$ or $e^{2\pi i/5}$ will never be represented accurately.

The basic question

Question 1 (Existence of algorithms)

Given any of the problems in (35), where the input may be given with some inaccuracy controlled by $\hat{\epsilon} > 0$, does there exists an algorithm that can compute an approximate solution, such that, for an arbitrary $\epsilon > 0$, the output will be no further than ϵ away from a true solution? The algorithm can choose $\hat{\epsilon}$ to be as small as desired (as a function of ϵ and the input) to produce the output.

A negative answer to this question is what Turing defined as non-computable.

The Second Paradox (algorithm)

There is an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0, 1\}$ such that for any neural network dimensions

$$\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0),$$

with $N_0, L \geq 2$ and any $K \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$ there exist uncountably many training sets $\mathcal{T}$ of size K such that for any cost function C such that $C(v, w) = 0$ iff $v = w$ then

the answer to Question 1 is no

for the computational problem (35) with $\mathcal{C} = \mathcal{T}$.

MATLAB tests of linear programming

- Linear Programming

$$z \in \operatorname*{argmin}_x \langle x, c \rangle \text{ such that } Ax = y, \quad x \geq 0,$$

where $A \in \mathbb{R}^{m \times N}$, $y \in \mathbb{R}^m$, $c \in \mathbb{R}^N$.

The exitflag

The `linprog` command has an `EXITFLAG` parameter that determines if MATLAB is happy with the computed solution. This parameter can take the following values

```
1  linprog converged to a solution X.
0  Maximum number of iterations reached.
-2  No feasible point found.
-3  Problem is unbounded.
-4  NaN value encountered during execution of algorithm.
-5  Both primal and dual problems are infeasible.
-7  Magnitude of search direction became too small;
     no further progress can be made. The problem is
     ill-posed or badly conditioned.
```

MATLAB tests

```
for k = 1:10
x = 10^(-k); A = [1-x,1]; b = 1; c = [1,1]
[c_soln, EXITFLAG] = linprog(c,[],[],A,b,[0,0],[100,100]);
error(k) = norm(c_soln-exact_soln);
flag(k) = EXITFLAG;
end
```

MATLAB tests

```
for k = 1:10
x = 10^(-k); A = [1-x,1]; b = 1; c = [1,1]
[c_soln, EXITFLAG] = linprog(c,[],[],A,b,[0,0],[100,100]);
error(k) = norm(c_soln-exact_soln);
flag(k) = EXITFLAG;
end

error =  8.7e-10 1.1e-10 3.0e-06 8.7e-11 6.6e-5
         7.0e-6 7.0e-6 0.2 0.6 0.7
```

MATLAB tests

```
for k = 1:10
x = 10^(-k); A = [1-x,1]; b = 1;
[c_soln, EXITFLAG] = linprog(c,[],[],A,b,[0,0],[100,100]);
error(k) = norm(c_soln-exact_soln);
flag(k) = EXITFLAG;
end

error =  8.7e-10 1.1e-10 3.0e-06 8.7e-11 6.6e-5
         7.0e-6 7.0e-6 0.2 0.6 0.7

flag = 1 1 1 1 1 1 1 1 1 1
```

Karmarkar's algorithm

From Wikipedia, the free encyclopedia

Karmarkar's algorithm is an algorithm introduced by Narendra Karmarkar in 1984 for solving linear programming problems. It was the first reasonably efficient algorithm that solves these problems in polynomial time. The ellipsoid method is also polynomial time but proved to be inefficient in practice.

Denoting n as the number of variables and L as the number of bits of input to the algorithm, Karmarkar's algorithm requires $O(n^{3.5}L)$ operations on $O(L)$ digit numbers, as compared to $O(n^6 L)$ such operations for the ellipsoid algorithm. The runtime of Karmarkar's algorithm is thus

$$O(n^{3.5}L^2 \cdot \log L \cdot \log\log L)$$

using FFT-based multiplication (see Big O notation).

Karmarkar's algorithm falls within the class of interior point methods: the current guess for the solution does not follow the boundary of the feasible set as in the simplex method, but it moves through the interior of the feasible region, improving the approximation of the optimal solution by a definite fraction with every iteration, and converging to an optimal solution with rational data.[1]

The answer to Question 1

Theorem 3

Fix the dimensions $m, N \in \mathbb{N}$ where $N \geq 4$. Consider the LP listed in (41). Choose any integer $K > 2$. There exists a domain Ω of inputs $(A, y, c) \in \mathbb{R}^{m \times N} \times \mathbb{R}^m \times \mathbb{R}^N$ such that:

(i) *No algorithm, even randomised, can halt and produce K correct digits for all inputs in Ω.*

(ii) *There does exist an algorithm that will provide $K - 1$ correct digits. However, any algorithm will need arbitrarily long time to reach $K - 1$ correct digits.*

(iii) *The problem of producing $K - 2$ digits is in P (polynomial in n, the number of variables).*

This statement is valid for any model of computation.

Can one compute the exit flag?

Given any of the problems in (41), it is impossible to design an algorithm to check whether the algorithm will fail on a given input.

This is in fact strictly harder than solving the original problem.

Thus, even if given an oracle for solving LP accurately, one cannot determine if a given algorithm for solving LP produces nonsense on its input.

Compressed sensing and deep learning in inverse problems

The basic inverse problem

Solve

$$Ax = y, \qquad A \in \mathbb{C}^{m \times N}, \quad x \in \mathbb{C}^{N}, \quad y \in \mathbb{C}^{m}.$$

Typical inverse problem where deep learning and compressed sensing is/may be used:

(i) Magnetic Resonance Imaging (MRI)
(ii) X-ray Computed Tomography
(iii) Thermoacoustic and Photoacoustic Tomography
(iv) Single Photon Emission Computerized Tomography
(v) Electrical Impedance Tomography
(vi) Electron Microscopy
(vii) Reflection seismology
(viii) Radio interferometry
(ix) Helium atom scattering

(i) Single pixel camera
(ii) Lensless camera
(iii) Compressive video
(iv) Fluorescence Microscopy

Most of these problems are modelled by the Fourier transform

$$\mathcal{F}f(\omega) = \int_{\mathbb{R}^d} f(x) e^{-2\pi i \omega \cdot x} \, dx,$$

or the Radon transform $\mathcal{R}f : \mathbf{S} \times \mathbb{R} \to \mathbb{C}$ (where $\mathbf{S}$ denotes the circle)

$$\mathcal{R}f(\theta, p) = \int_{\langle x, \theta \rangle = p} f(x) \, dm(x),$$

where dm denotes Lebesgue measure on the hyperplane $\{x : \langle x, \theta \rangle = p\}$.

- Fourier slice theorem $\Rightarrow$ both problems can be viewed as the problem of reconstructing f from pointwise samples of its Fourier transform.

$$g = \mathcal{F}f, \quad f \in L^2(\mathbb{R}^d). \tag{1}$$

Does Deep Learning work in inverse problems?

Time saving experiment in MRI

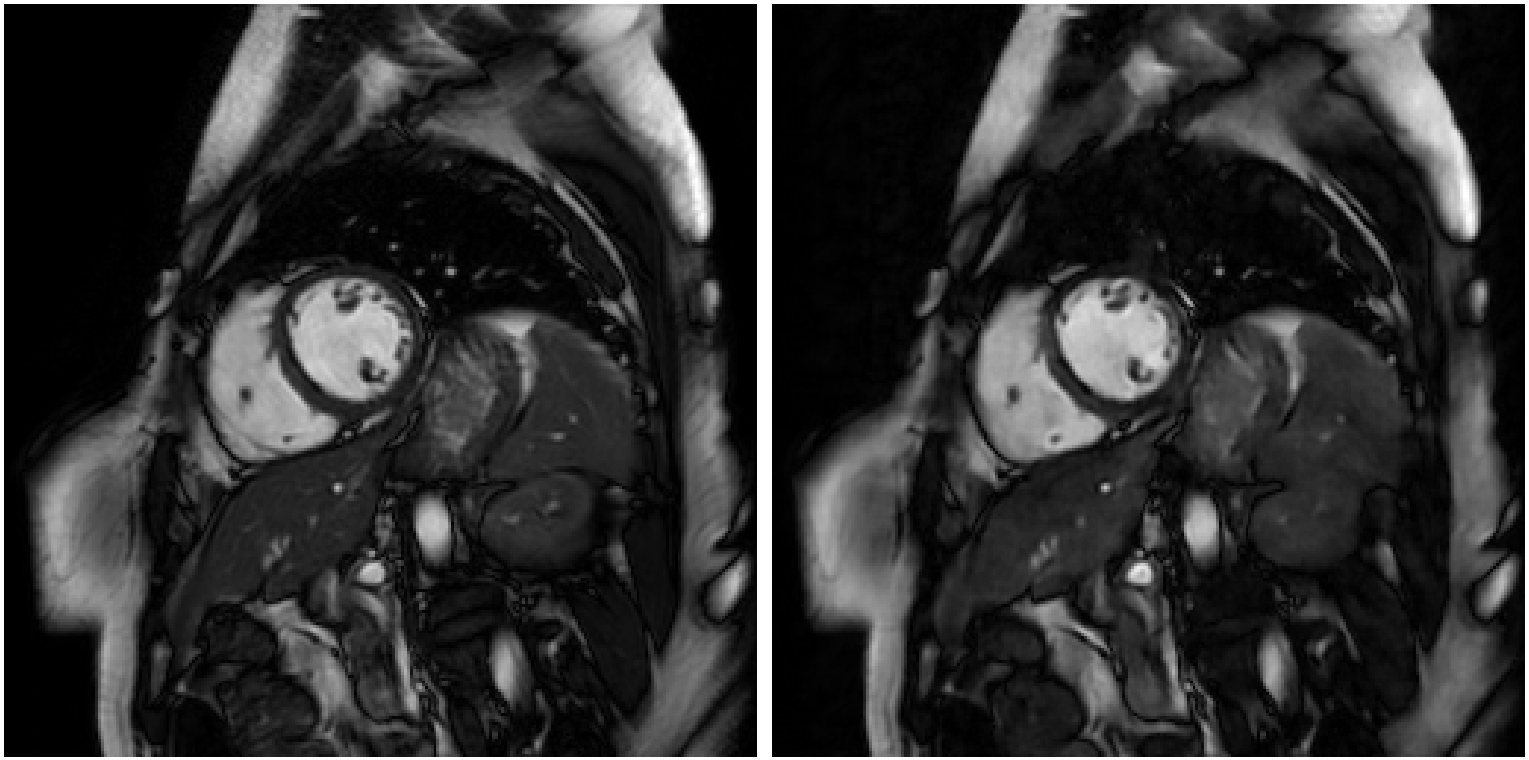


Figure : Left: Original image. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Does Compressed Sensing Work?

Resolution enhancing experiment in MRI

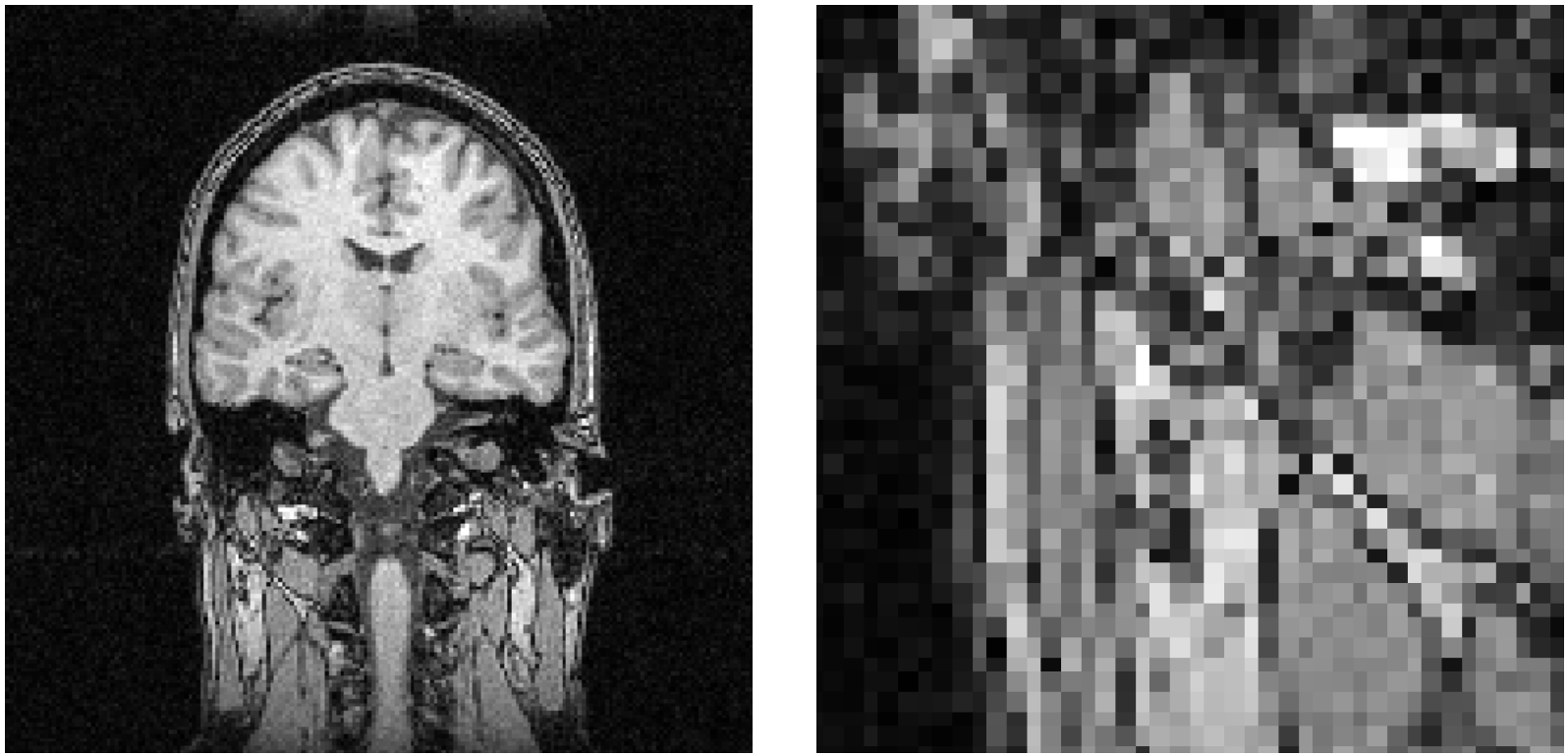


Figure : Standard 3D MRI headscan. Scanning time = 15 min.

Experiment from "Undersampling improves fidelity of physical imaging and the benefits grow with resolution", B. Roman, R. Calderbank, B. Adcock D. Nietlispach, M. Bostock, I. Calvo-Almazn, M. Graves A. Hansen, *PNAS* (in revision).

Does Compressed Sensing Work?

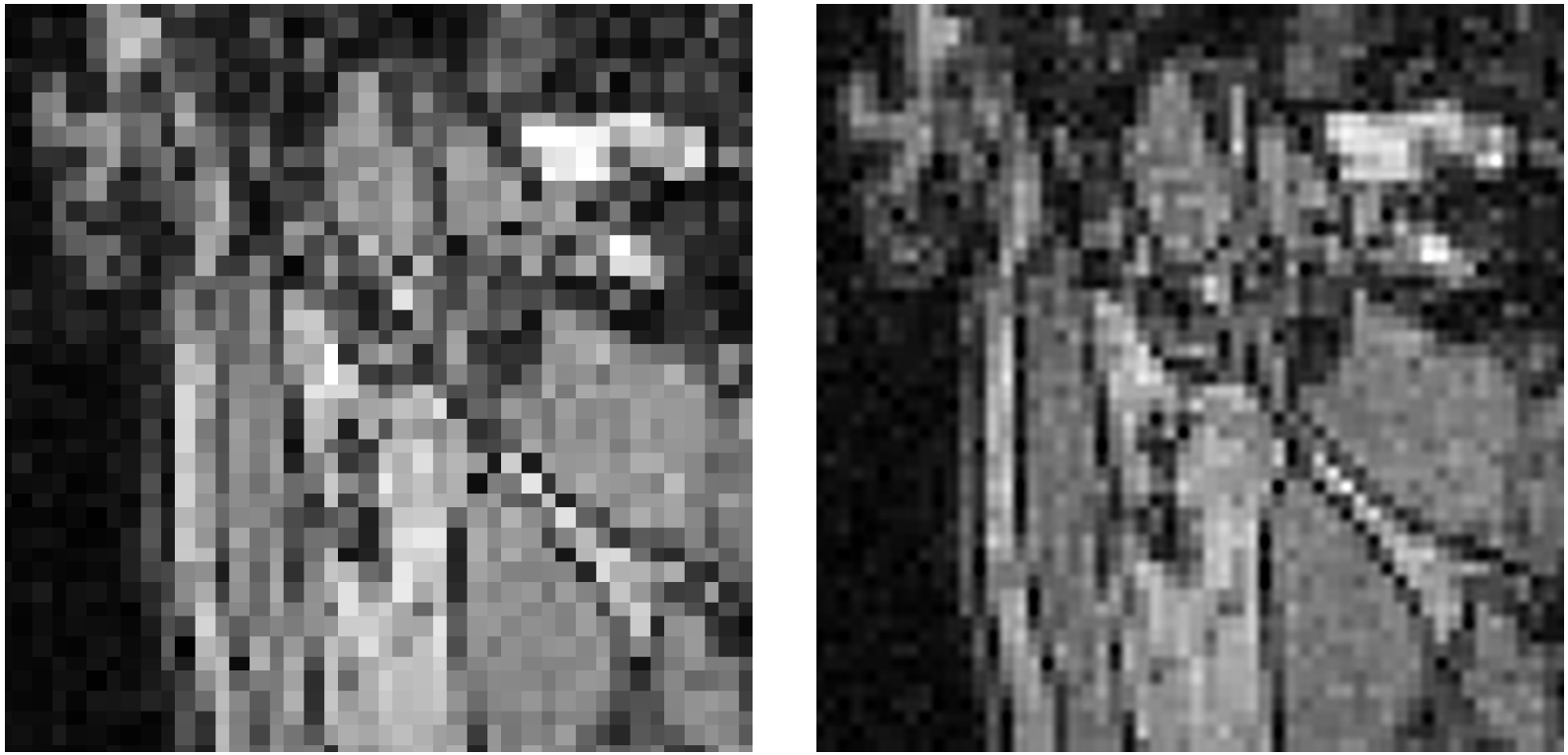


Figure : Left: Standard full sampling. Right: One type of compressed sensing approaches to resolution enhancing. Scanning time for both = 15 min.

Experiment from "Undersampling improves fidelity of physical imaging and the benefits grow with resolution", B. Roman, R. Calderbank, B. Adcock D. Nietlispach, M. Bostock, I. Calvo-Almazn, M. Graves A. Hansen, *PNAS* (in revision).

Does Compressed Sensing Work?

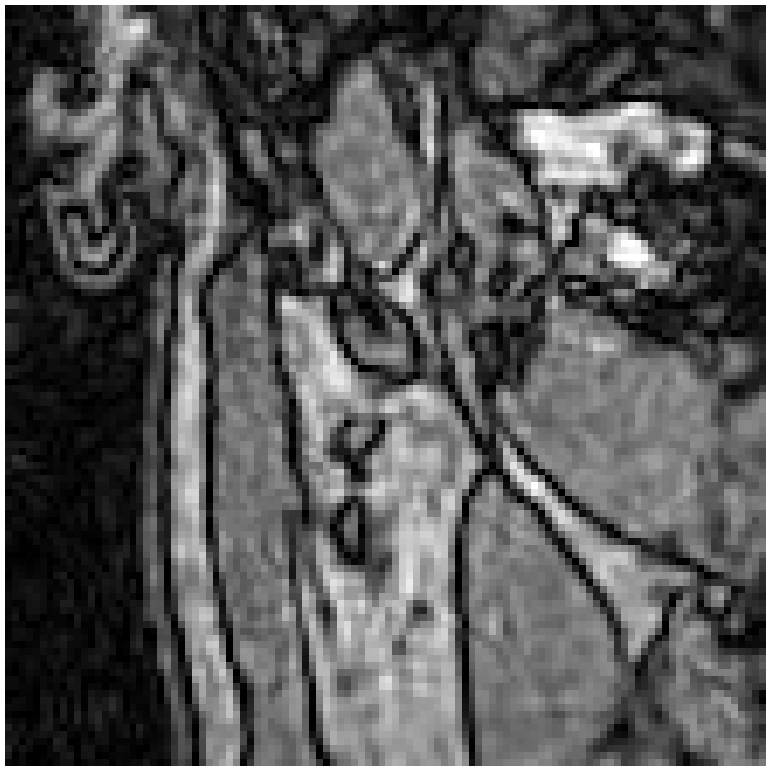



Figure : Two different compressed sensing approaches to resolution enhancing. Scanning time = 15 min

Experiment from "Undersampling improves fidelity of physical imaging and the benefits grow with resolution", B. Roman, R. Calderbank, B. Adcock D. Nietlispach, M. Bostock, I. Calvo-Almazn, M. Graves A. Hansen, *PNAS* (in revision).

Does Compressed Sensing Work?

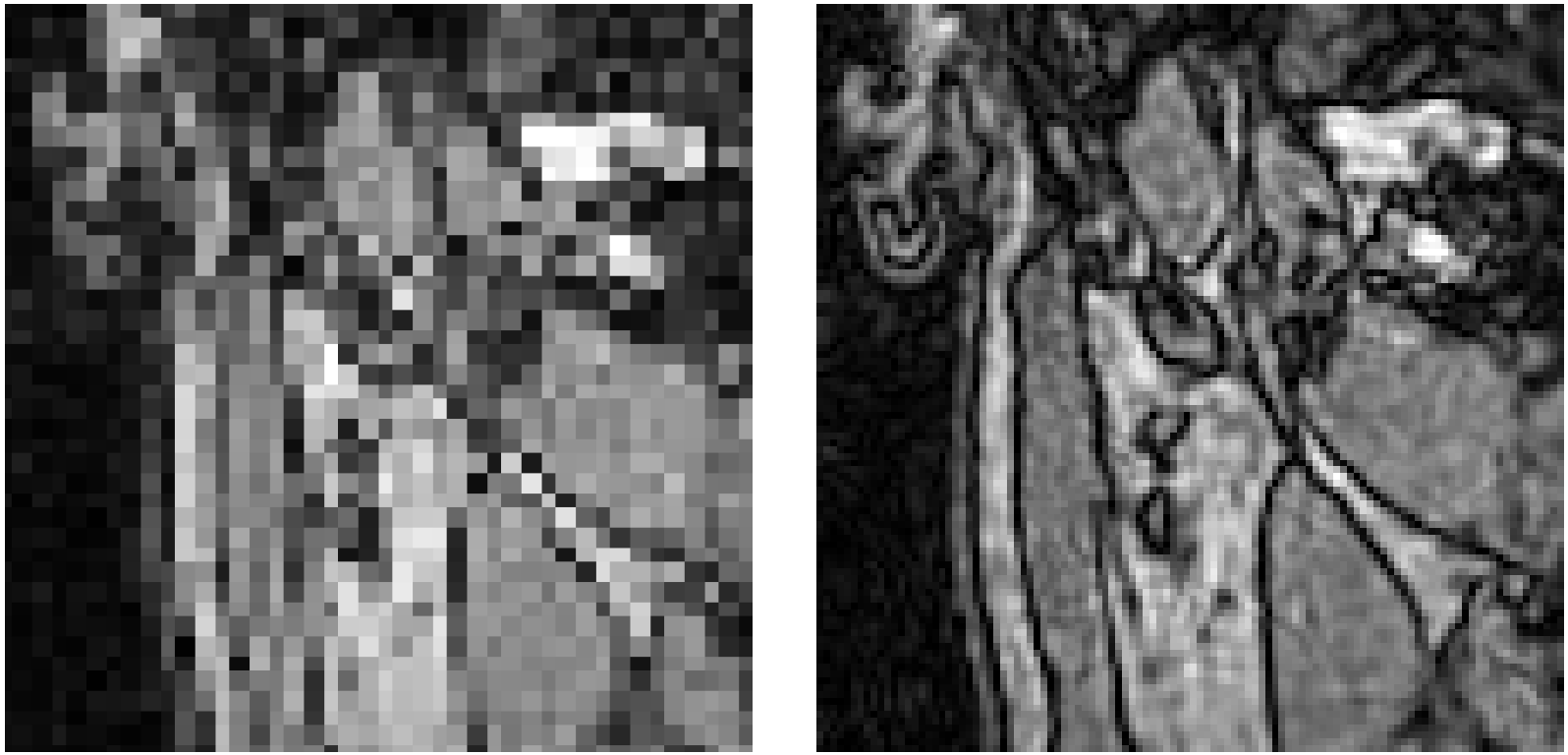


Figure : Left: Standard full sampling. Right: The correct type of compressed sensing approaches to resolution enhancing. Scanning time for both = 15 min

Experiment from "Undersampling improves fidelity of physical imaging and the benefits grow with resolution", B. Roman, R. Calderbank, B. Adcock D. Nietlispach, M. Bostock, I. Calvo-Almazn, M. Graves A. Hansen, *PNAS* (in revision).

Deep learning in inverse problems

Given a crappy images $x \in \mathbb{R}^d$, train a neural network $\phi \in \mathcal{NN}_{\mathbf{N},L,d}$ to get a good images

$$y = \phi(x).$$

In practice, one tries to learn the noise and use

$$y = x - \phi(x).$$

Denoising experiment with deep learning

Original

Noisy version

Denoised with Neur. Net.

DL in Inverse Problems: 1st Step

```
>> I = phantom(512); theta_1 = [0:1:179];
>> R = radon(I, theta_1);
>> imshow(I); imagesc(R)
```

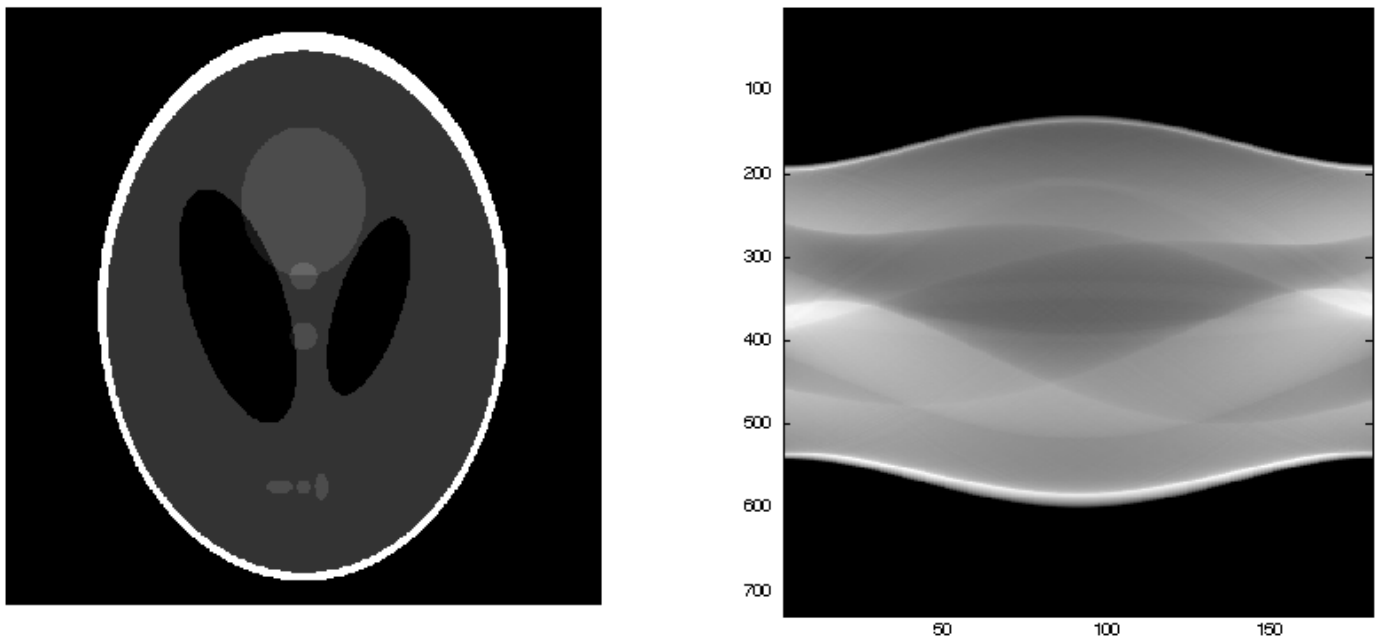


Figure : Left: Logan-Shepp Phantom. Right: The image under the Radon transform (sinogram)

DL in Inverse Problems: 1st Step

```
>> I = phantom(512); theta_3 = [0:3:179];
>> R = radon(I, theta_3); II = iradon(R,theta_3);
>> imshow(I); imagesc(II)
```

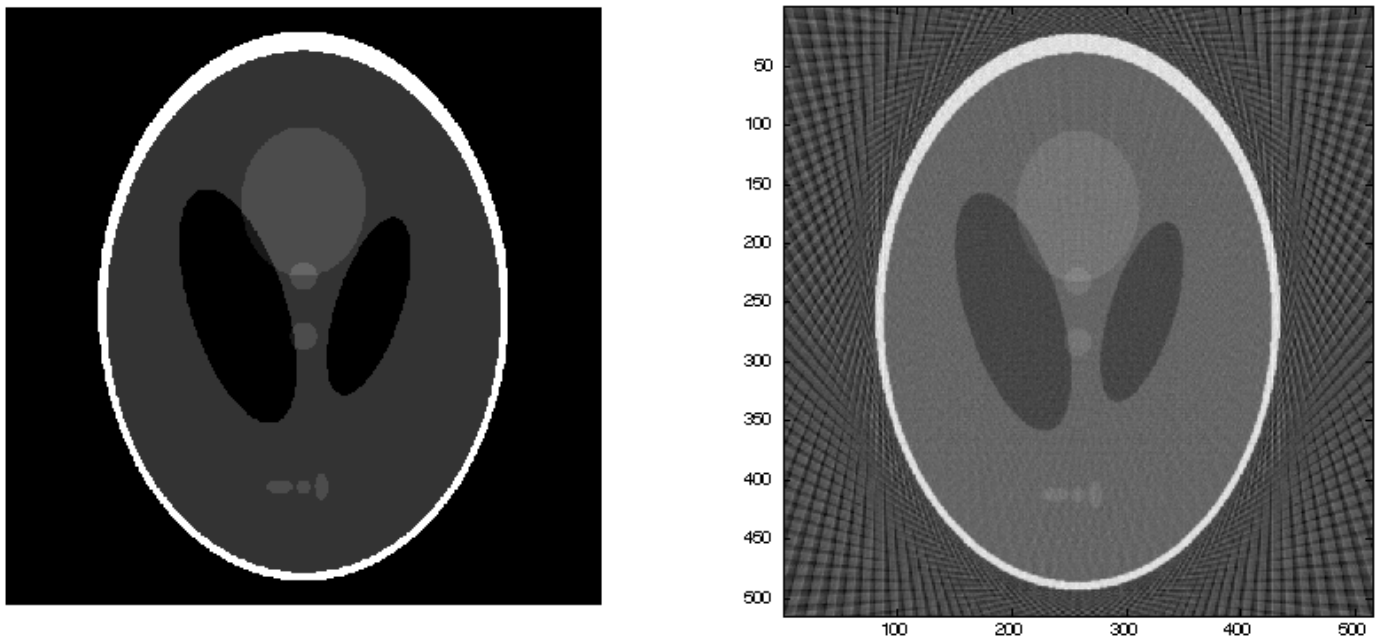


Figure : Left: Logan-Shepp Phantom. Right: Reconstruction with the filtered back projection using 60 lines.

DL in Inverse Problems: 1st Step

Crazy idea: The filtered back projection gives a noisy image.

Why don't we try deep learning to denoise the image. In particular, we train a neural network ϕ such that

$$x \approx \text{iradon}(\text{radon}(x)) - \phi(\text{iradon}(\text{radon}(x))$$

Deep Convolutional Neural Network for Inverse Problems in Imaging

Kyong Hwan Jin, Michael T. McCann, *Member, IEEE*, Emmanuel Froustey, Michael Unser, *Fellow, IEEE*

Abstract

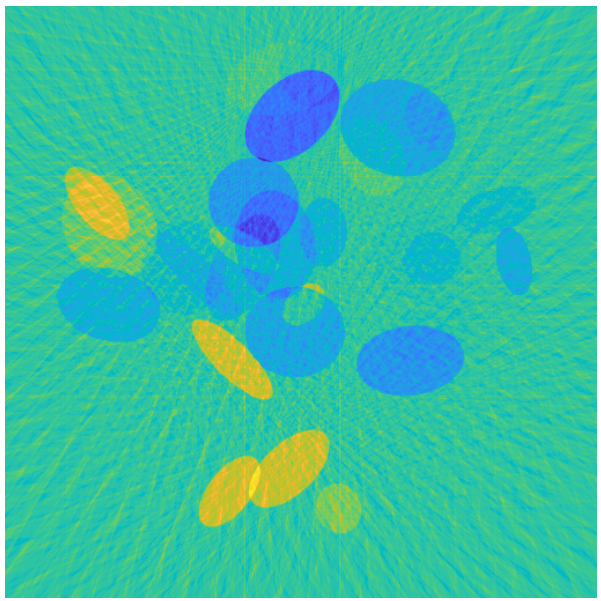
In this paper, we propose a novel deep convolutional neural network (CNN)-based algorithm for solving ill-posed inverse problems. Regularized iterative algorithms have emerged as the standard approach to ill-posed inverse problems in the past few decades. These methods produce excellent results, but can be challenging to deploy in practice due to factors including the high computational cost of the forward and adjoint operators and the difficulty of hyper parameter selection. The starting point of our work is the observation that unrolled iterative methods have the form of a CNN (filtering followed by point-wise non-linearity) when the normal operator (H^*H, the adjoint of H times H) of the forward model is a convolution. Based on this observation, we propose using direct inversion followed by a CNN to solve normal-convolutional inverse problems. The direct inversion encapsulates the physical model of the system, but leads to artifacts when the problem is ill-posed; the CNN combines multiresolution decomposition and residual learning in order to learn to remove these artifacts while preserving image structure. We demonstrate the performance of the proposed network in sparse-view reconstruction (down to 50 views) on parallel beam X-ray computed tomography in synthetic phantoms as well as in real experimental sinograms. The proposed network outperforms total variation-regularized iterative reconstruction for the more realistic phantoms and requires less than a second to reconstruct a 512×512 image on the GPU.

DL in Inverse Problems: 1st Step (Experiments)

Computerised Tomography (CT) experiment with deep learning

Original

Recon-FBP (50 lines)

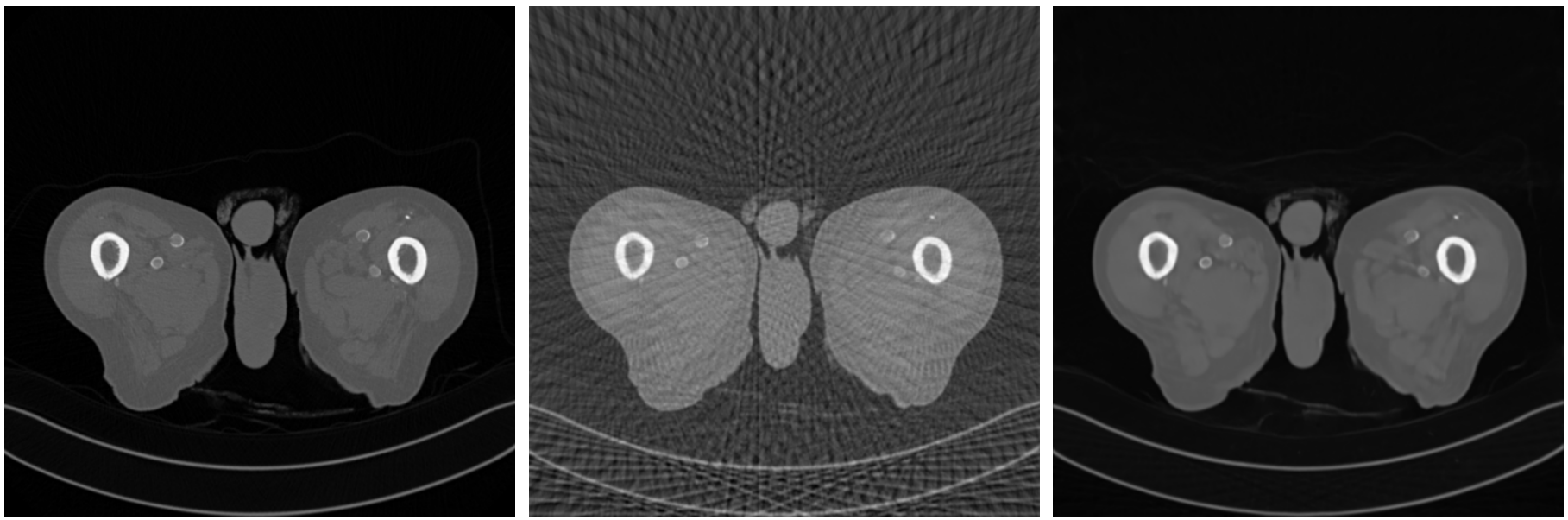


Recon-NeurNet (50 lines)

DL in Inverse Problems: 1st Step (Experiments)

Computerised Tomography (CT) experiment with deep learning

Original | Recon-FBP (50 lines) | Recon-NeurNet (50 lines)



DL in Inverse Problems: 1st Step (Experiments)

CT experiment with deep learning - capturing details

Figure : Original image (with small detail added).

DL in Inverse Problems: 1st Step (Experiments)

CT experiment with deep learning - capturing details

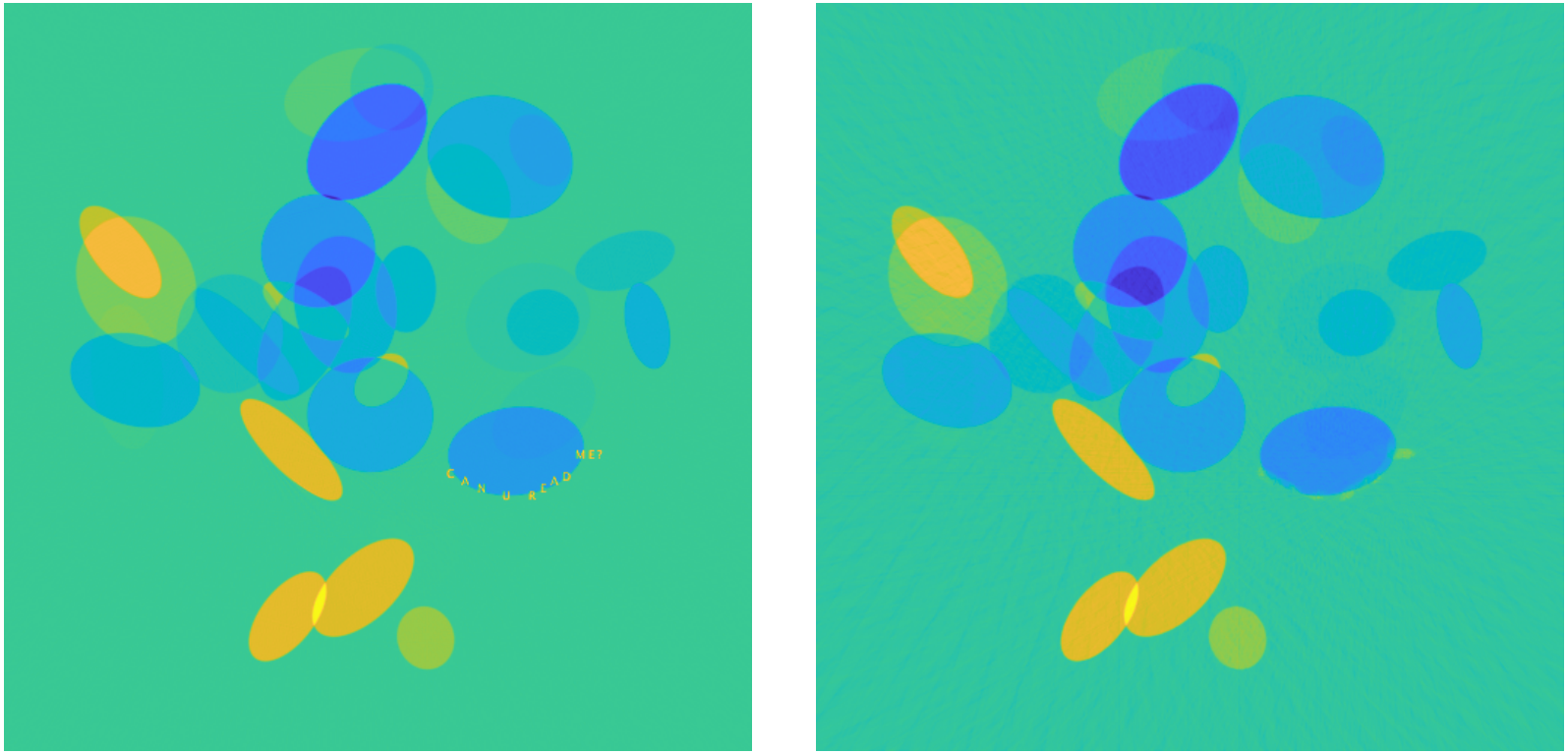


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.

DL in Inverse Problems: 1st Step (Experiments)

CT experiment with deep learning - capturing details

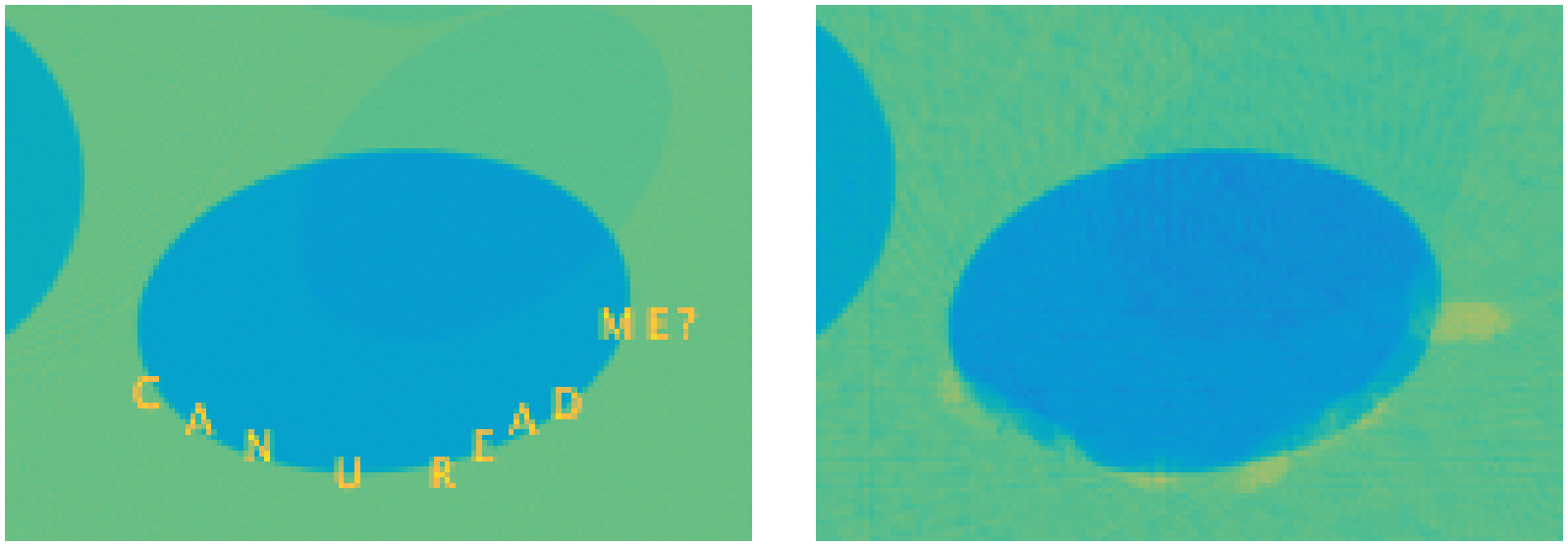


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.

Compressed sensing in inverse problems

- Given the linear system

$$Ux_0 = y.$$

- Solve

$$\min \|z\|_1 \quad \text{subject to } P_\Omega U z = P_\Omega y,$$

where P_Ω is a projection and $\Omega \subset \{1, \ldots, N\}$ is subsampled with $|\Omega| = m$.

Compressed Sensing: Why l_1 optimisation?

Let $U \in \mathbb{C}^{n \times n}$, $x_0 \in \mathbb{C}^n$ and consider

$$y = Ux_0.$$

We want to recover x_0 from y.

- This is obvious if U is invertible and we know y.
- What if we do not know y, but rather

$$P_\Omega y,$$

  where P_Ω is the projection onto $\operatorname{span}\{e_j\}_{j \in \Omega}$ and $\Omega \subset \{1, \ldots, n\}$ with $|\Omega| = m$ and Ω is randomly chosen.
- Can we recover x_0 from $P_\Omega y$?

Given $x_0 \in \mathbb{C}^n$ let

$$\Delta = \operatorname{supp}(x_0) = \{k \in \mathbb{N} : \langle x_0, e_k \rangle \neq 0\}.$$

Want to find a strategy so that x_0 can be reconstructed from $P_\Omega U x_0$, where $|\Omega| = m$, with high probability. In particular we would like to know how large m must be as a function of n and $|\Delta|$.

Why sparsity?

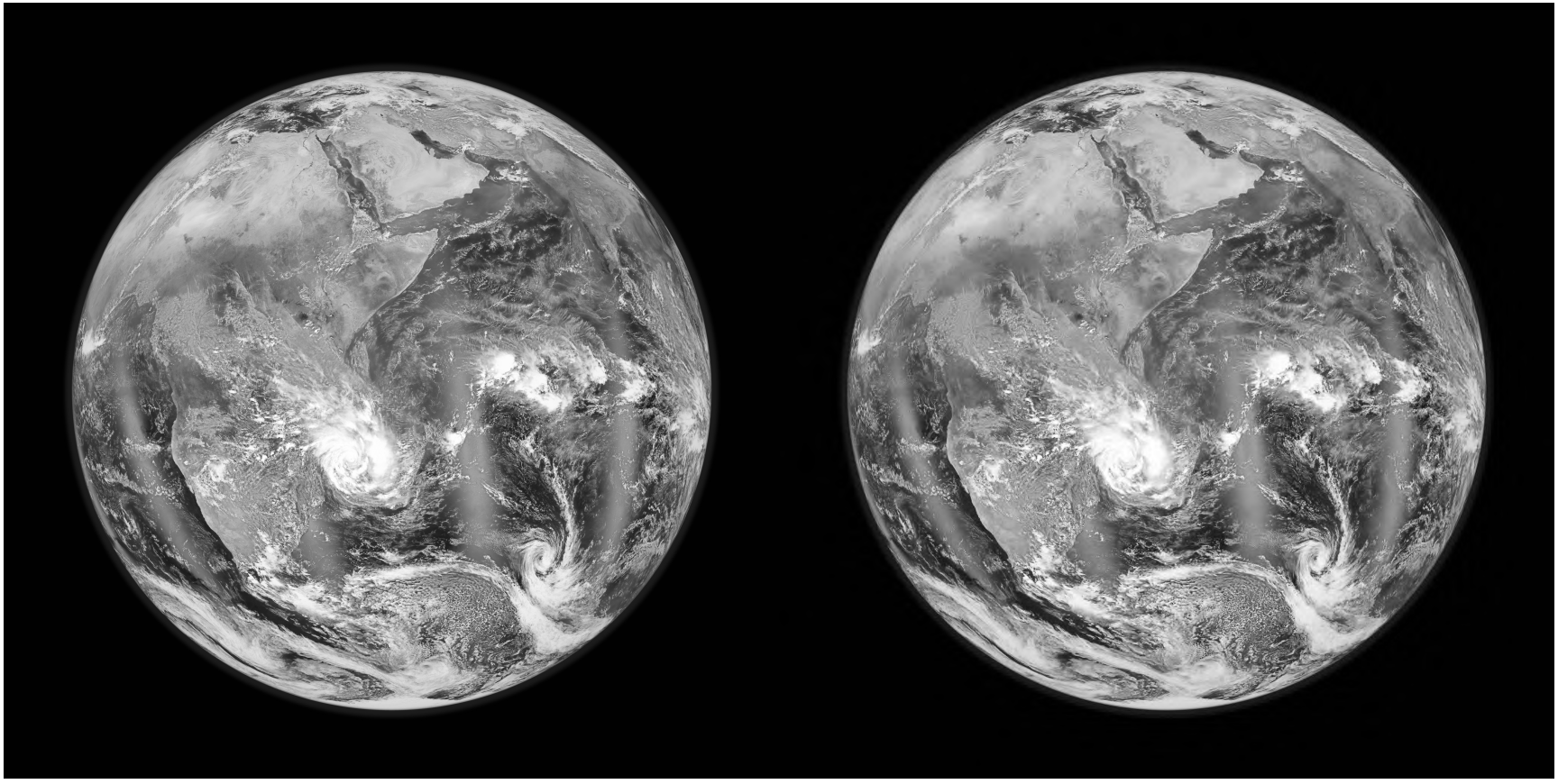
The world is sparse in "you name it"-lets.

Wavelets and their cousins curvelets, contourlets, shearlets etc. yield sparse expansions of natural images. Suppose that f is an image and $\{\varphi_n\}_{n\in\mathbb{N}}$ is a "you name it"-let system of functions, then

$$f = \sum_{j=1}^{\infty} \alpha_n \varphi_n,$$

where only few of the α_ns are important. One keeps these and sets the rest to zero. This is the key to modern compression.

Compression



The right image is compressed where 99% of the coefficients (Daubechies 4 wavelet) are set to zero.

Want to recover x_0 from $P_\Omega U x_0$ by finding

$$\inf_x \|x\|_{l^0}, \qquad P_\Omega U x = P_\Omega U x_0 \tag{2}$$

where $\|x\|_{l^0} = |\{j : x_j \neq 0\}|$ or

$$\inf_x \|x\|_{l^1}, \qquad P_\Omega U x = P_\Omega U x_0, \tag{3}$$

where $\|x\|_{l^1} = \sum_{j=1}^n |x_j|$. Note that (2) is a non-convex optimization problem and (3) is a convex optimization problem.

MATLAB tests

MATLAB tests (Basis Pursuit)

```
>> y = 1
for k = 1:10
x = 10^(-k);
A = [1-x,1];
[z,info] = spgl1( A, y, [],[], [], opts);
iterations(k) = info.iter;
error(k) = norm(z-true);
end
```

MATLAB tests (Basis Pursuit)

```
>> y = 1
for k = 1:10
x = 10^(-k);
A = [1-x,1];
[z,info] = spgl1( A, y, [],[], [], opts);
iterations(k) = info.iter;
error(k) = norm(z-true);
end

iterations =
4    4   1593  133005  7834764   3   4   4   6   4

error =
0    0  1.4e-09 4.3e-08 4.4e-05  0.7 0.7 0.7 0.7 0.7
```

Theorem 4

Fix the dimensions $m, N \in \mathbb{N}$ where $N \geq 4$. Consider the Basis Pursuit problem in (3). Choose any integer $K > 2$. There exists a domain Ω of inputs $(A, y) \in \mathbb{R}^{m \times N} \times \mathbb{R}^m$ such that:

(i) *No algorithm, even randomised, can halt and produce K correct digits for all inputs in Ω.*

(ii) *There does exist an algorithm that will provide $K - 1$ correct digits. However, any algorithm will need arbitrarily long time to reach $K - 1$ correct digits.*

(iii) *The problem of producing $K - 2$ digits is in P (polynomial in n, the number of variables).*

This statement is valid for any model of computation.

- Given the linear system

$$Ux_0 = y.$$

- Solve

$$\min \|z\|_1 \quad \text{subject to } P_\Omega U z = P_\Omega y,$$

where P_Ω is a projection and $\Omega \subset \{1, \ldots, N\}$ is subsampled with $|\Omega| = m$.

If U is unitary, Ω is chosen uniformly at random and

$$m \geq C \cdot N \cdot \mu(U) \cdot s \cdot \log(\epsilon^{-1}) \cdot \log(N),$$

then $\mathbb{P}(z = x_0) \geq 1 - \epsilon$, where

$$\mu(U) = \max_{i,j} |U_{i,j}|^2$$

is referred to as the incoherence parameter.

Pillars of Compressed Sensing

- Sparsity
- Incoherence
- Uniform Random Subsampling

Pillars of Compressed Sensing

- Sparsity
- Incoherence
- Uniform Random Subsampling

Problem: These concepts are absent in virtually all the inverse problems listed above. Moreover, uniform random subsampling gives highly suboptimal results.

Compressed sensing is currently used with great success in these fields, however the current theory does not cover this.

Simplified real world CS problem

- Unknown signal $x_0 \in \mathbb{C}^N$.
- Sampling equipments samples Ux_0 where $U \in \mathbb{C}^{N \times N}$.
- x_0 may not be sparse but $\tilde{x} = V_{\mathrm{dwt}} x_0$ may be.

Subsample $\Omega \subset \{1, \ldots, N\}$ with $m = |\Omega|$ and solve

$$\min \|z\|_1 \quad \text{subject to } P_\Omega U V_{\mathrm{dwt}}^{-1} z = P_\Omega U V_{\mathrm{dwt}}^{-1} \tilde{x}.$$

Problems with standard CS theory

We need to make sure that UV_{dwt}^{-1} satisfies the requirements for compressed sensing. What about the coherence?

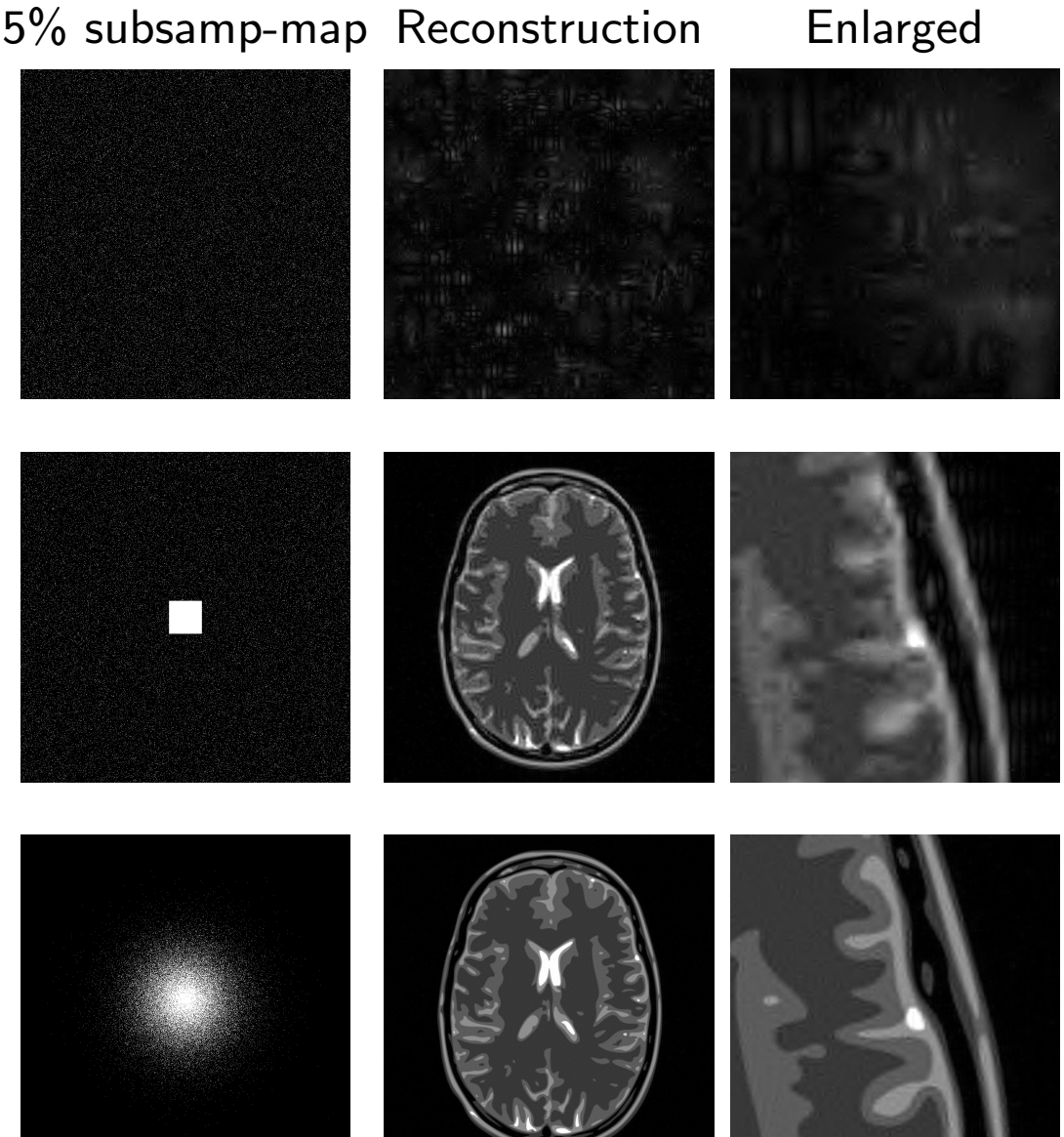
- Let $U_N = U_{\mathrm{dft}} V_{\mathrm{dwt}}^{-1} \in \mathbb{C}^{N \times N}$ where U_{dft} is the discrete Fourier transform. Then $\mu(U_N) = 1$ for all Daubechies wavelets and all N.
- Let $U_N = U_{\mathrm{Had}} V_{\mathrm{dwt}}^{-1} \in \mathbb{C}^{N \times N}$ where U_{Had} is a Hadamard matrix. Then $\mu(U_N) = 1$ for all Daubechies wavelets and all N.

This means that

$$m \geq C \cdot N \cdot \mu(U) \cdot s \cdot \log(\epsilon^{-1}) \cdot \log(N) > N.$$

Uniform Random Subsampling

$$U = U_{\mathrm{dft}} V_{\mathrm{dwt}}^{-1}.$$

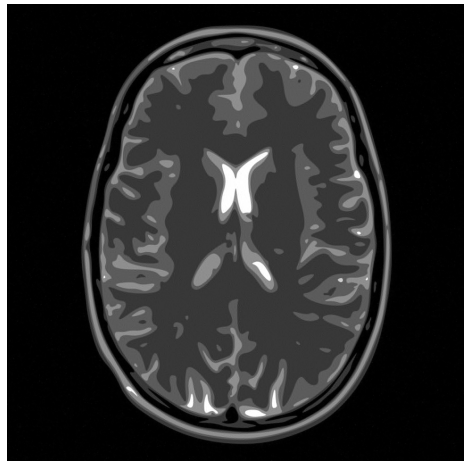


Sparsity

- The classical idea of sparsity in compressed sensing is that there are s important coefficients in the vector x_0 that we want to recover.
- The location of these coefficients is arbitrary.

Sparsity and the Flip Test

Let

$x =$ 

and

$$y = U_{\mathrm{df}} x, \qquad A = P_\Omega U_{\mathrm{df}} V_{\mathrm{dw}}^{-1},$$

where P_Ω is a projection and $\Omega \subset \{1, \ldots, N\}$ is subsampled with $|\Omega| = m$. Solve

$$\min \|z\|_1 \quad \text{subject to } Az = P_\Omega y.$$

Sparsity - The Flip Test

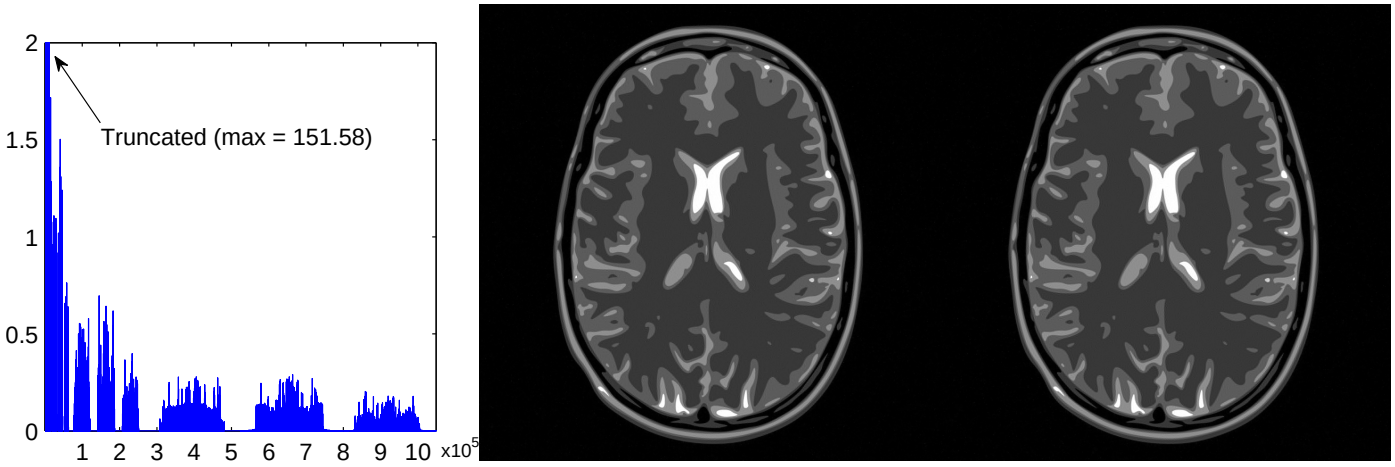
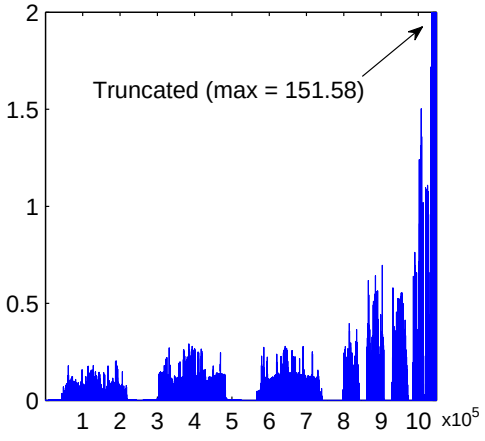


Figure : Wavelet coefficients and subsampling reconstructions from 10% of Fourier coefficients with distributions $(1+\omega_1^2+\omega_2^2)^{-1}$ and $(1+\omega_1^2+\omega_2^2)^{-3/2}$.

If sparsity is the right model we should be able to flip the coefficients. Let

$z_f =$



Sparsity - The Flip Test

- Let
$$\tilde{y} = U_{\mathrm{df}} V_{\mathrm{dw}}^{-1} z_f$$
- Solve
$$\min \|z\|_1 \quad \text{subject to } Az = P_\Omega \tilde{y}$$
to get $\hat{z}_f$.
- Flip the coefficients of $\hat{z}_f$ back to get $\hat{z}$, and let $\hat{x} = V_{\mathrm{dw}}^{-1} \hat{z}$.
- If the ordering of the wavelet coefficients did not matter i.e. sparsity is the right model, then $\hat{x}$ should be close to x.

Sparsity- The Flip Test: Results

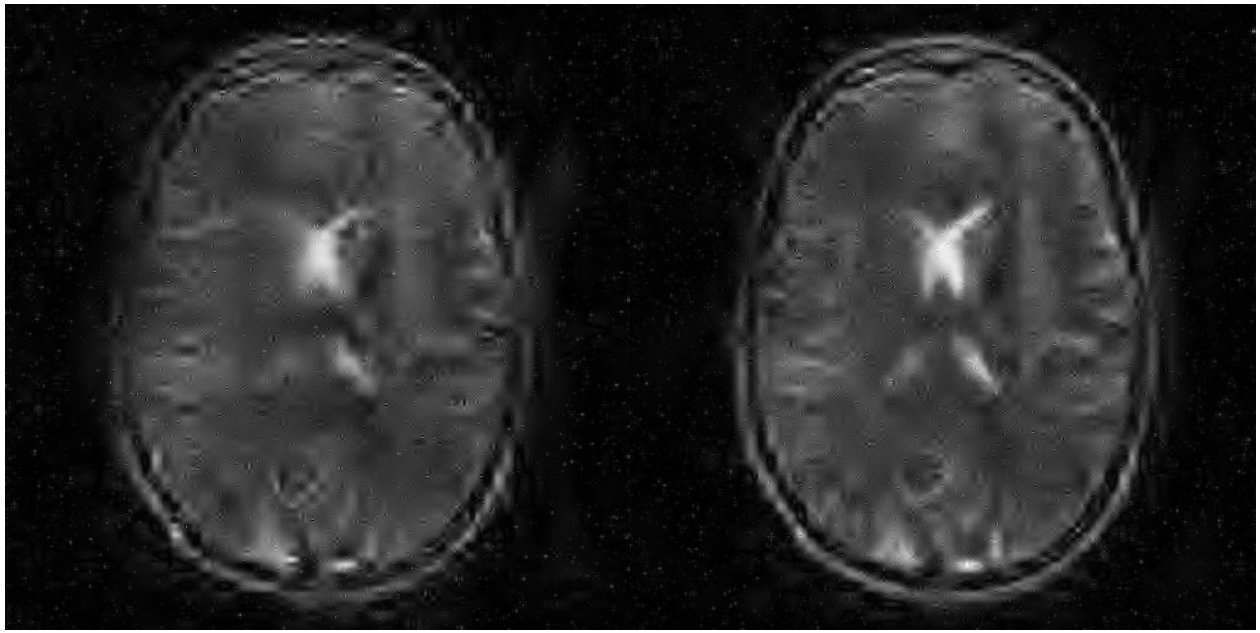


Figure : The reconstructions from the reversed coefficients.

Conclusion: The ordering of the coefficients did matter. Moreover, this phenomenon happens with all wavelets, curvelets, contourlets and shearlets and any reasonable subsampling scheme.

Question: Is sparsity really the right model?

Compressed sensing 2.0

- Compressed sensing 2.0 must incorporates structure.
- It must explain the two intriguing phenomena observed in practice:
  - The optimal sampling strategy is signal structure dependent
  - The success of compressed sensing is resolution dependent

New Pillars of Compressed Sensing

- Asymptotic Sparsity
- Asymptotic Incoherence
- Multi-level Subsampling

New Pillars of Compressed Sensing

From *SIAM News* Oct. 2017, A. Bastounis, B. Adcock, A. Hansen

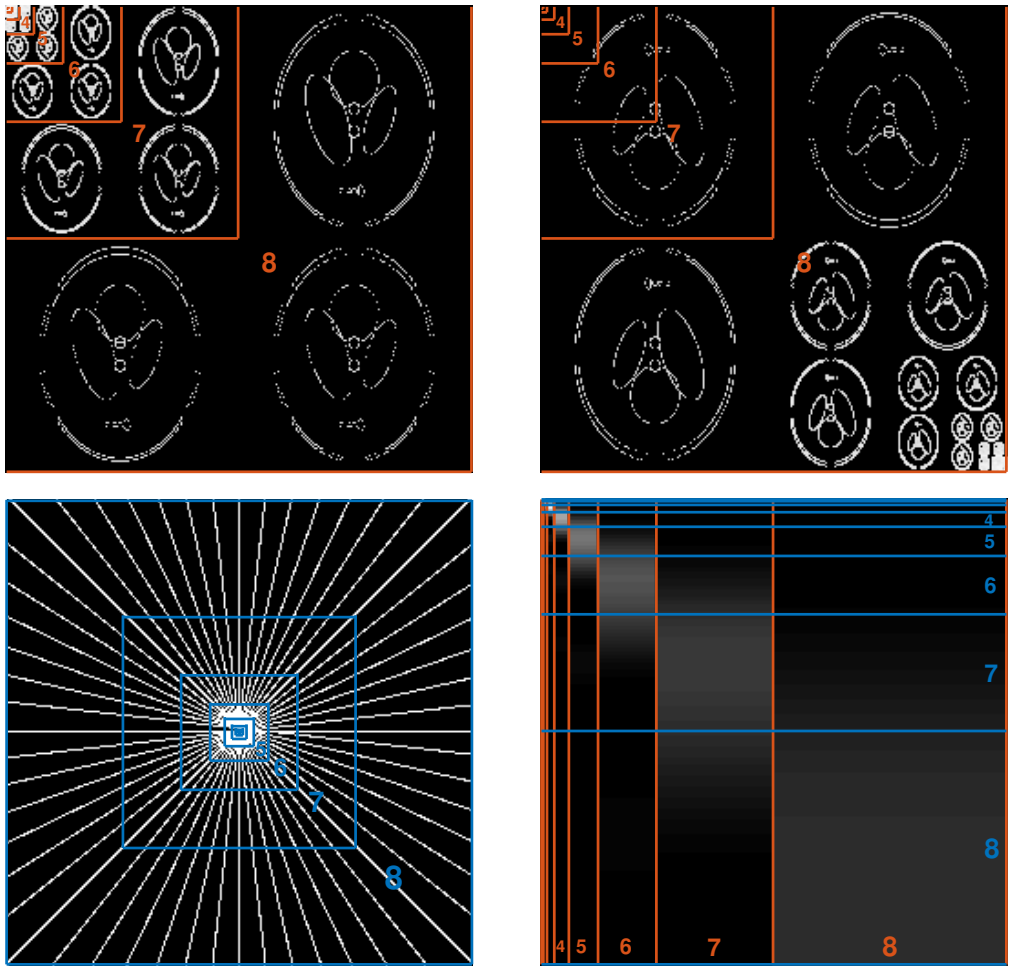


Figure : (a) Wavelet coefficients arranged according to increasing scale. (b) flipped coefficients. (c) Radial sampling map in k-space. (d) Absolute values of the matrix $U = U_{\mathrm{df}} V_{\mathrm{dw}}^{-1}$, with larger and smaller values correspond to lighter or darker colours respectively.

Definition 5

For $r \in \mathbb{N}$ let $\mathbf{M} = (M_1, \dots, M_r) \in \mathbb{N}^r$ with $1 \leq M_1 < \dots < M_r$ and $\mathbf{s} = (s_1, \dots, s_r) \in \mathbb{N}^r$, with $s_k \leq M_k - M_{k-1}$, $k = 1, \dots, r$, where $M_0 = 0$. We say that $\beta \in l^2(\mathbb{N})$ is $(\mathbf{s}, \mathbf{M})$-sparse if, for each $k = 1, \dots, r$,

$$\Delta_k := \operatorname{supp}(\beta) \cap \{M_{k-1} + 1, \dots, M_k\},$$

satisfies $|\Delta_k| \leq s_k$. We denote the set of $(\mathbf{s}, \mathbf{M})$-sparse vectors by $\Sigma_{\mathbf{s},\mathbf{M}}$.

Definition 6

Let $f = \sum_{j\in\mathbb{N}} \beta_j \varphi_j \in \mathcal{H}$, where $\beta = (\beta_j)_{j\in\mathbb{N}} \in l^1(\mathbb{N})$. Let

$$\sigma_{\mathbf{s},\mathbf{M}}(f) := \min_{\eta\in\Sigma_{\mathbf{s},\mathbf{M}}} \|\beta - \eta\|_{l^1}. \tag{4}$$

Definition 7

Let $r \in \mathbb{N}$, $\mathbf{N} = (N_1, \ldots, N_r) \in \mathbb{N}^r$ with $1 \leq N_1 < \ldots < N_r$, $\mathbf{m} = (m_1, \ldots, m_r) \in \mathbb{N}^r$, with $m_k \leq N_k - N_{k-1}$, $k = 1, \ldots, r$, and suppose that

$$\Omega_k \subseteq \{N_{k-1} + 1, \ldots, N_k\}, \quad |\Omega_k| = m_k, \quad k = 1, \ldots, r,$$

are chosen uniformly at random, where $N_0 = 0$. We refer to the set

$$\Omega = \Omega_{\mathbf{N},\mathbf{m}} := \Omega_1 \cup \ldots \cup \Omega_r.$$

as an $(\mathbf{N}, \mathbf{m})$-multilevel sampling scheme.

r-level Sampling Scheme

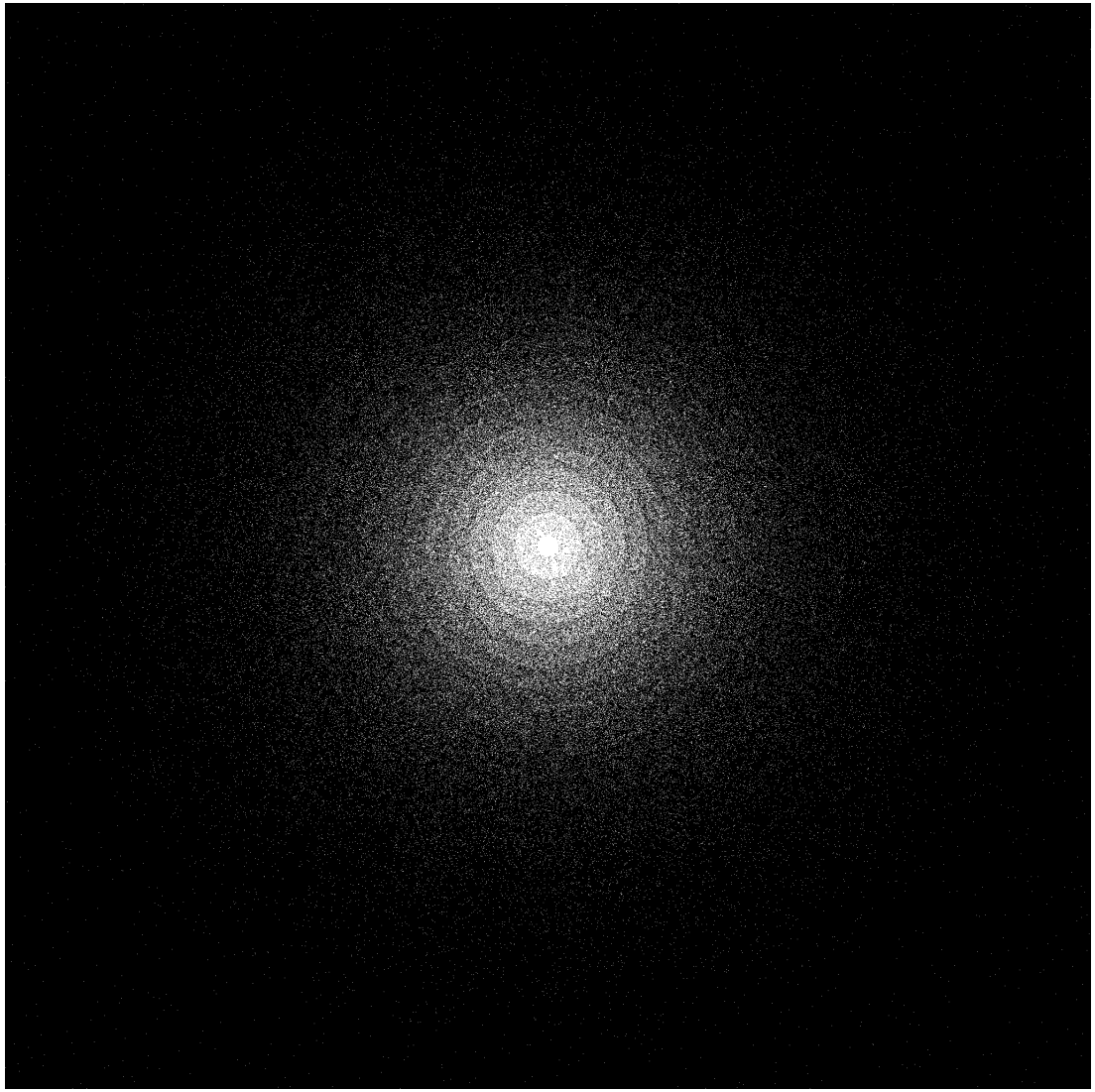
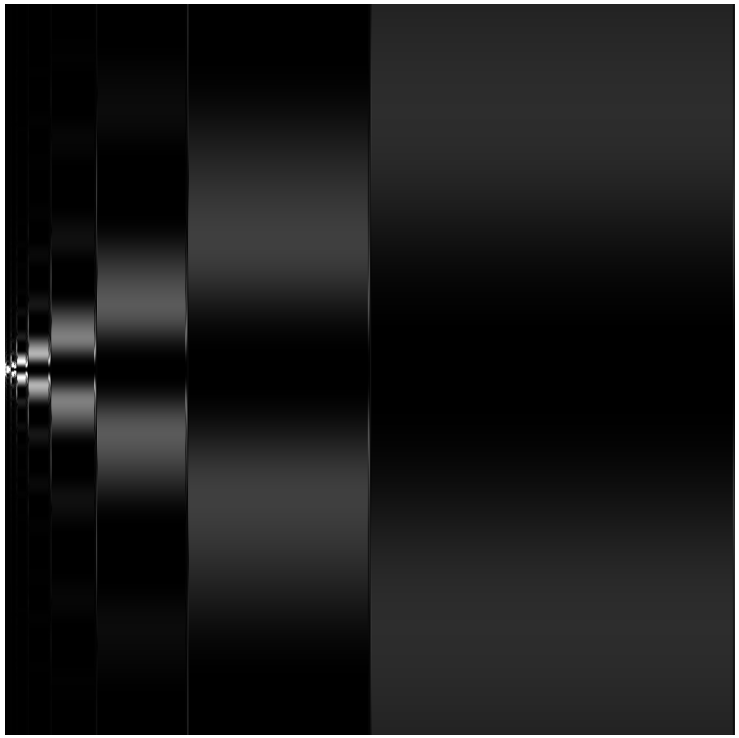


Figure : The typical sampling pattern that will be used.

Analog inverse problems are asymptotically incoherent

Fourier to DB4



Fourier to Legendre Polynomials

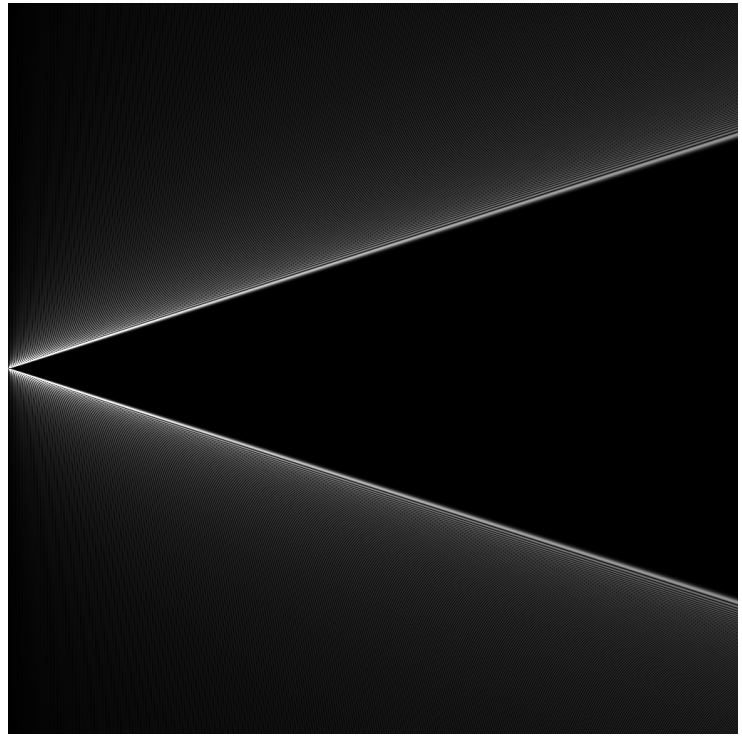


Figure : Plots of the absolute values of the entries of the matrix U

Definition 8

Let $U \in \mathbb{C}^{N\times N}$. If $\mathbf{N} = (N_1, \ldots, N_r) \in \mathbb{N}^r$ and $\mathbf{M} = (M_1, \ldots, M_r) \in \mathbb{N}^r$ with $1 \leq N_1 < \ldots N_r$ and $1 \leq M_1 < \ldots < M_r$ we define the $(k, l)^{\text{th}}$ local coherence of U with respect to $\mathbf{N}$ and $\mathbf{M}$ by

$$\mu_{\mathbf{N},\mathbf{M}}(k, l) = \sqrt{\mu(P_{N_k}^{N_{k-1}} U P_{M_l}^{M_{l-1}}) \cdot \mu(P_{N_k}^{N_{k-1}} U)}, \quad k, l = 1, \ldots, r,$$

where $N_0 = M_0 = 0$.

The optimization problem

$$\inf_{\eta \in \ell^1(\mathbb{N})} \|\eta\|_{\ell^1} \text{ subject to } \|P_\Omega U \eta - y\| \leq \delta. \tag{5}$$

Theoretical Results (recovery and stability)

Let $U \in \mathbb{C}^{N\times N}$ be an isometry and $\beta \in \mathbb{C}^N$. Suppose that $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ is a multilevel sampling scheme, where $\mathbf{N} = (N_1, \dots, N_r) \in \mathbb{N}^r$ and $\mathbf{m} = (m_1, \dots, m_r) \in \mathbb{N}^r$. Let $(\mathbf{s}, \mathbf{M})$, where $\mathbf{M} = (M_1, \dots, M_r) \in \mathbb{N}^r$, $M_1 < \dots < M_r$, and $\mathbf{s} = (s_1, \dots, s_r) \in \mathbb{N}^r$, be any pair such that the following holds: for $\epsilon > 0$ and $1 \leq k \leq r$,

$$1 \gtrsim \frac{N_k - N_{k-1}}{m_k} \cdot \log(\epsilon^{-1}) \cdot \left(\sum_{l=1}^{r} \mu_{\mathbf{N},\mathbf{M}}(k,l) \cdot s_l \right) \cdot \log(N) . \tag{6}$$

Suppose that $\xi \in \mathbb{C}^N$ is a minimizer of (5) with $\delta = \tilde{\delta}\sqrt{K^{-1}}$ and $K = \max_{1\leq k\leq r}\{(N_k - N_{k-1})/m_k\}$. Then, with probability exceeding $1 - s\epsilon$, where $s = s_1 + \dots + s_r$, we have that

$$\|\xi - \beta\| \leq C \cdot \left(\tilde{\delta} \cdot \left(1 + L \cdot \sqrt{s}\right) + \sigma_{\mathbf{s},\mathbf{M}}(f) \right),$$

for some constant C, where $\sigma_{\mathbf{s},\mathbf{M}}(f)$ is as in (4), $L = 1 + \frac{\sqrt{\log_2\left(6\epsilon^{-1}\right)}}{\log_2(4KM\sqrt{s})}$ and $K = \max_{k=1,\dots,r}\left\{\frac{N_k - N_{k-1}}{m_k}\right\}$.

Number of samples in each level:

$$m_k \gtrsim \log(\epsilon^{-1}) \cdot \log(N) \cdot \frac{N_k - N_{k-1}}{N_{k-1}} \cdot \left(\hat{s}_k + \sum_{l=1}^{k-2} s_l \cdot 2^{-\alpha(k-l)} + \sum_{l=k+2}^{r} s_l \cdot 2^{-v(l-k)} \right),$$

where $\hat{s}_k = \max\{s_{k-1}, s_k, s_{k+1}\}$. Note that

$$m \gtrsim s \cdot \log(N), \qquad m = m_1 + \ldots + m_r, \quad s = s_1 + \ldots + s_r.$$

Note that this shows how the success of CS will be resolution dependent.

Instabilities of deep learning in inverse problems

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

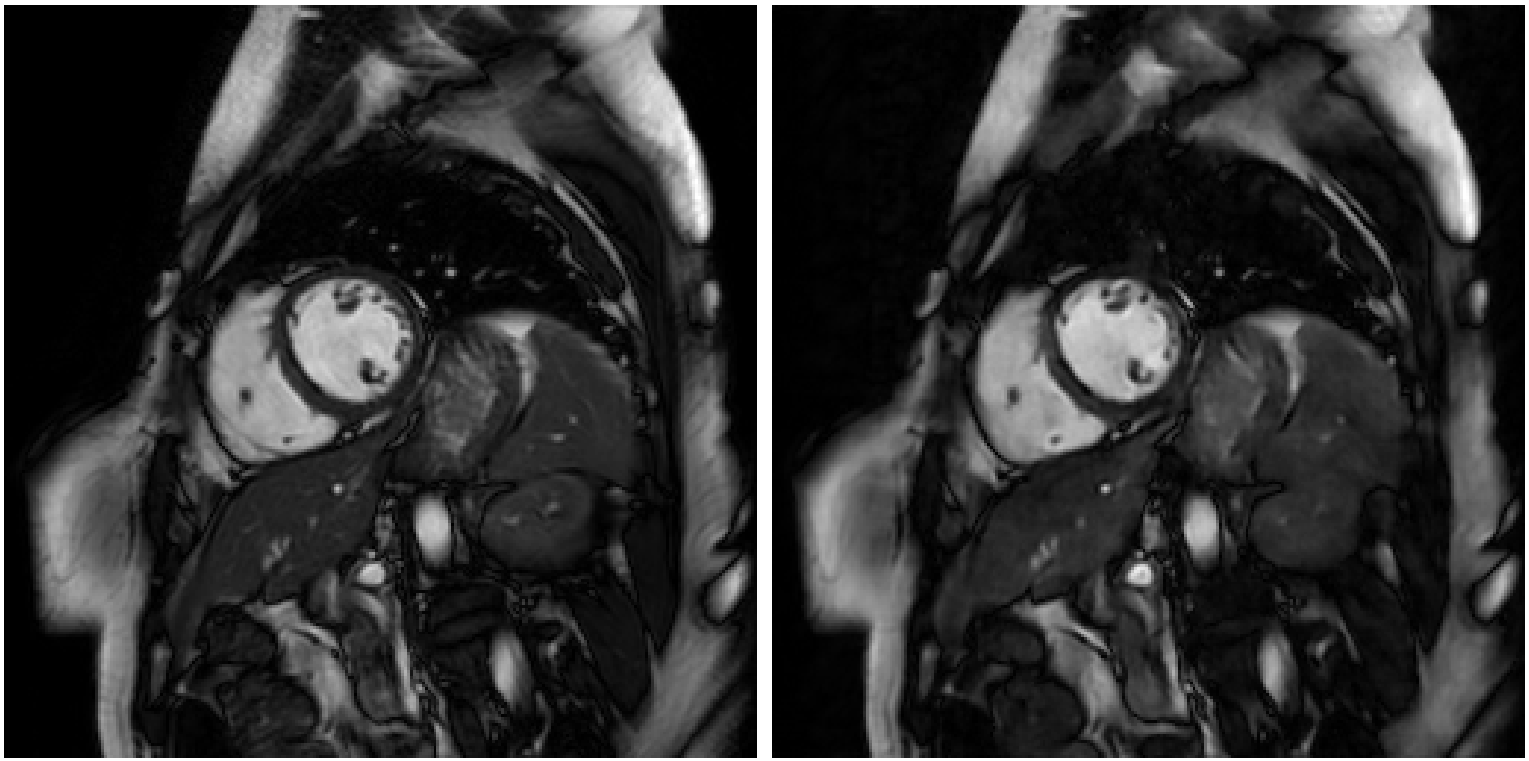


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

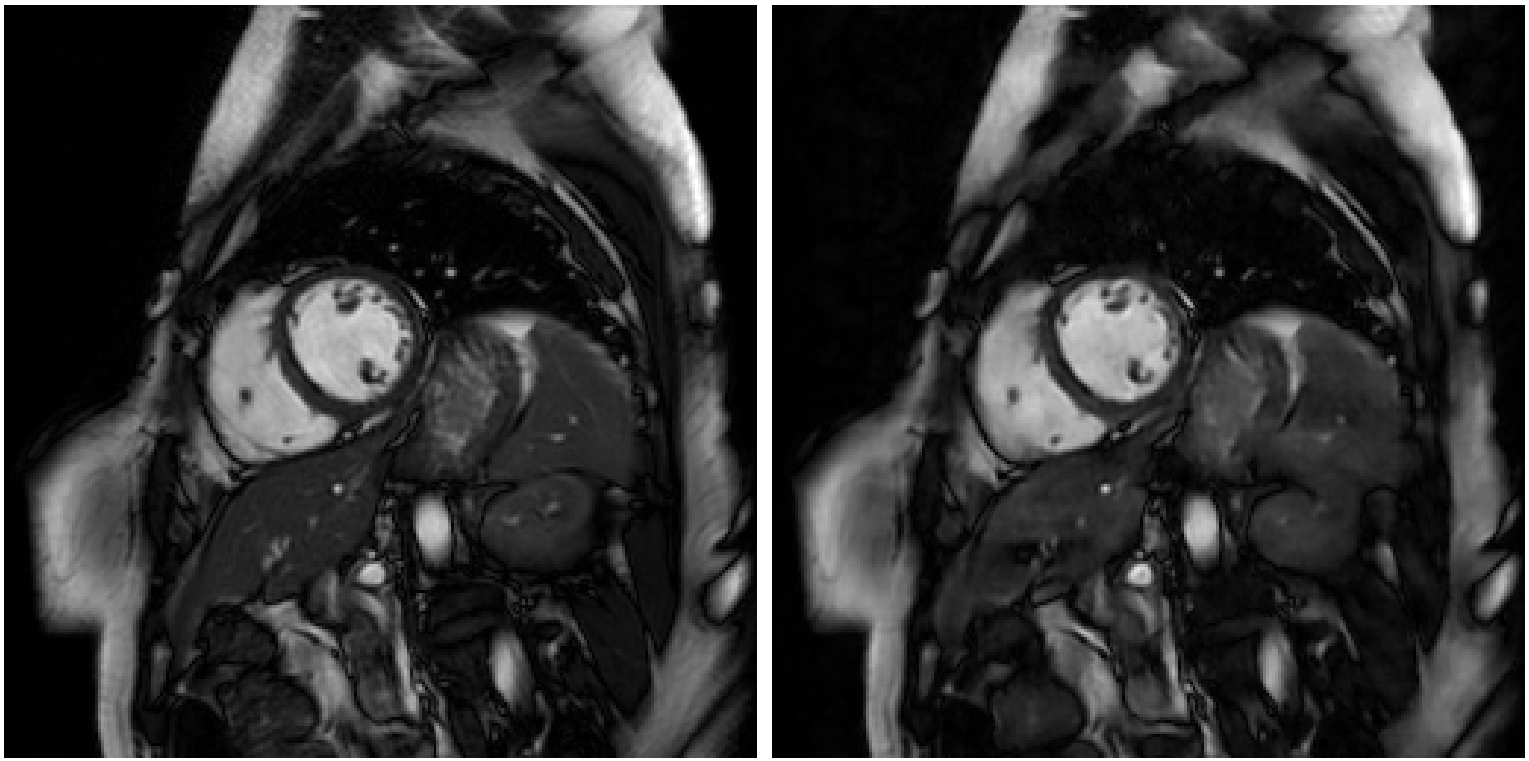


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

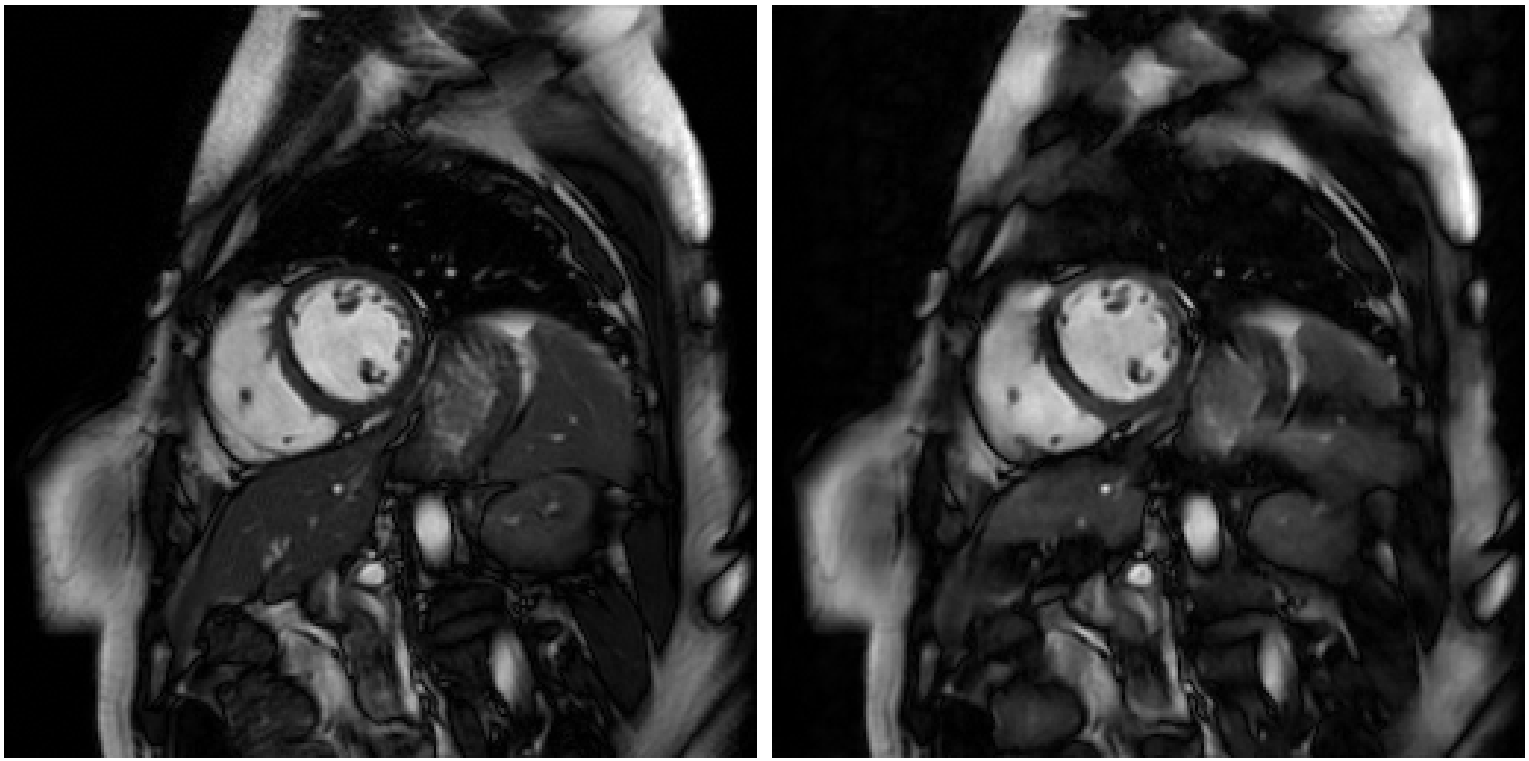


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

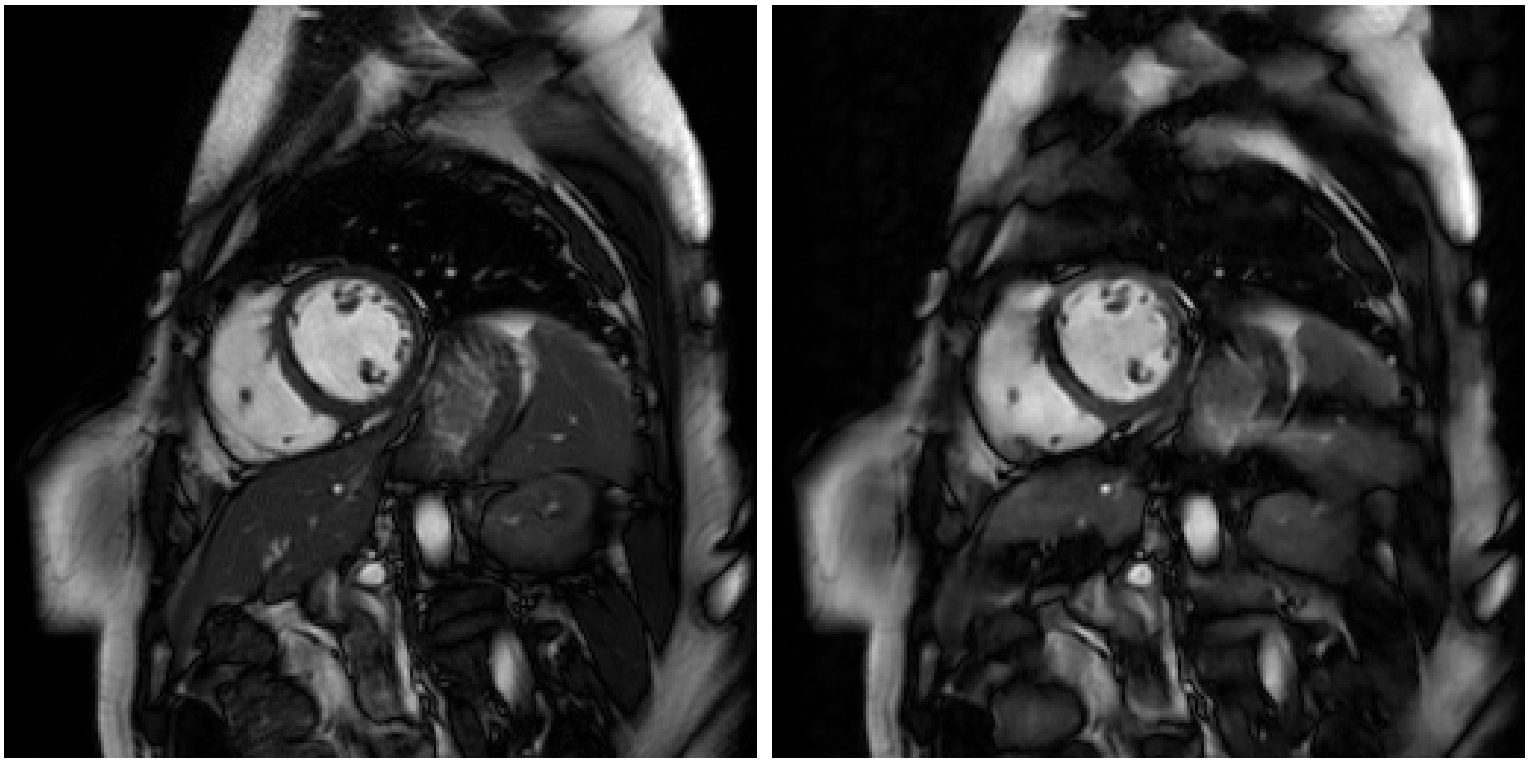


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

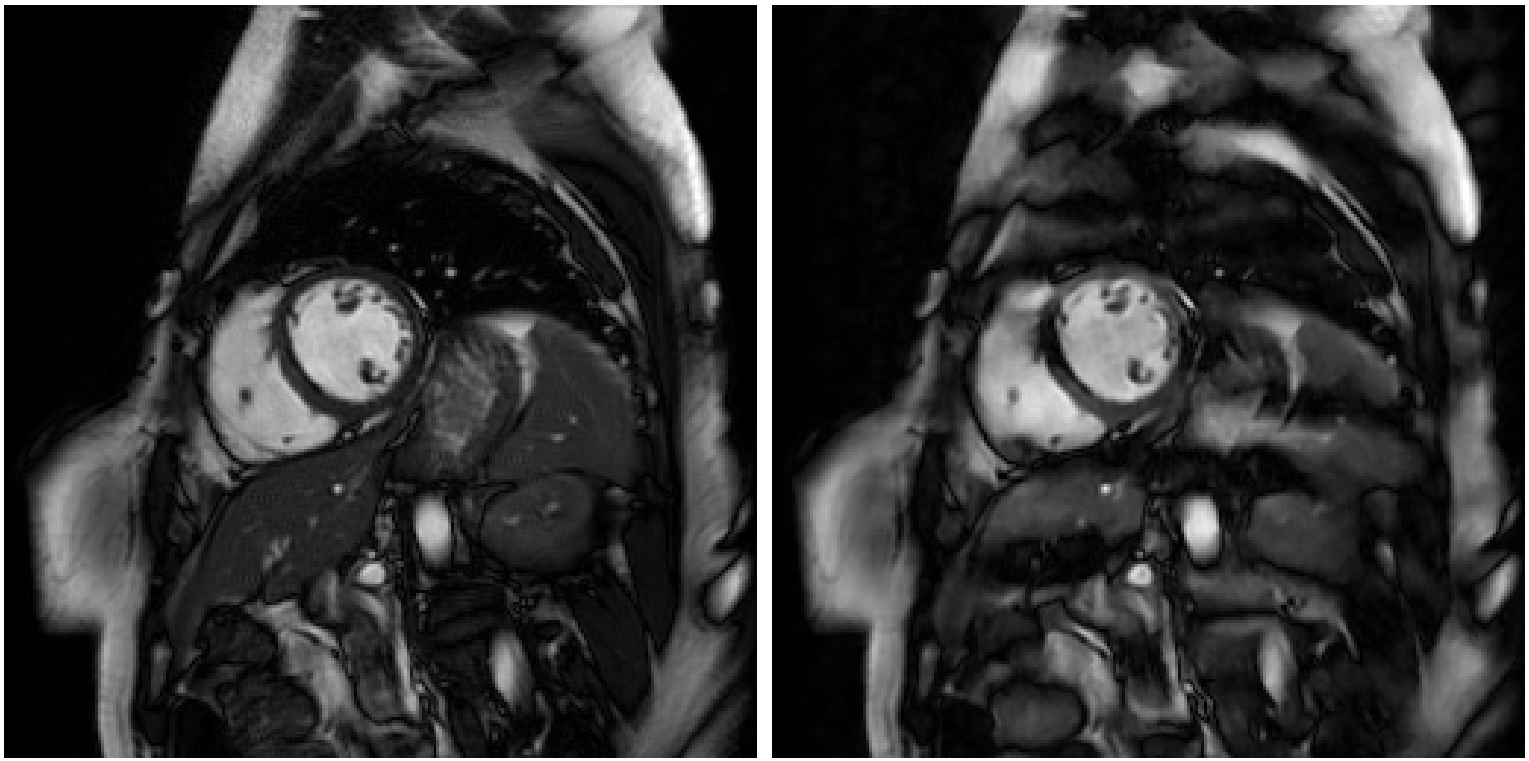


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

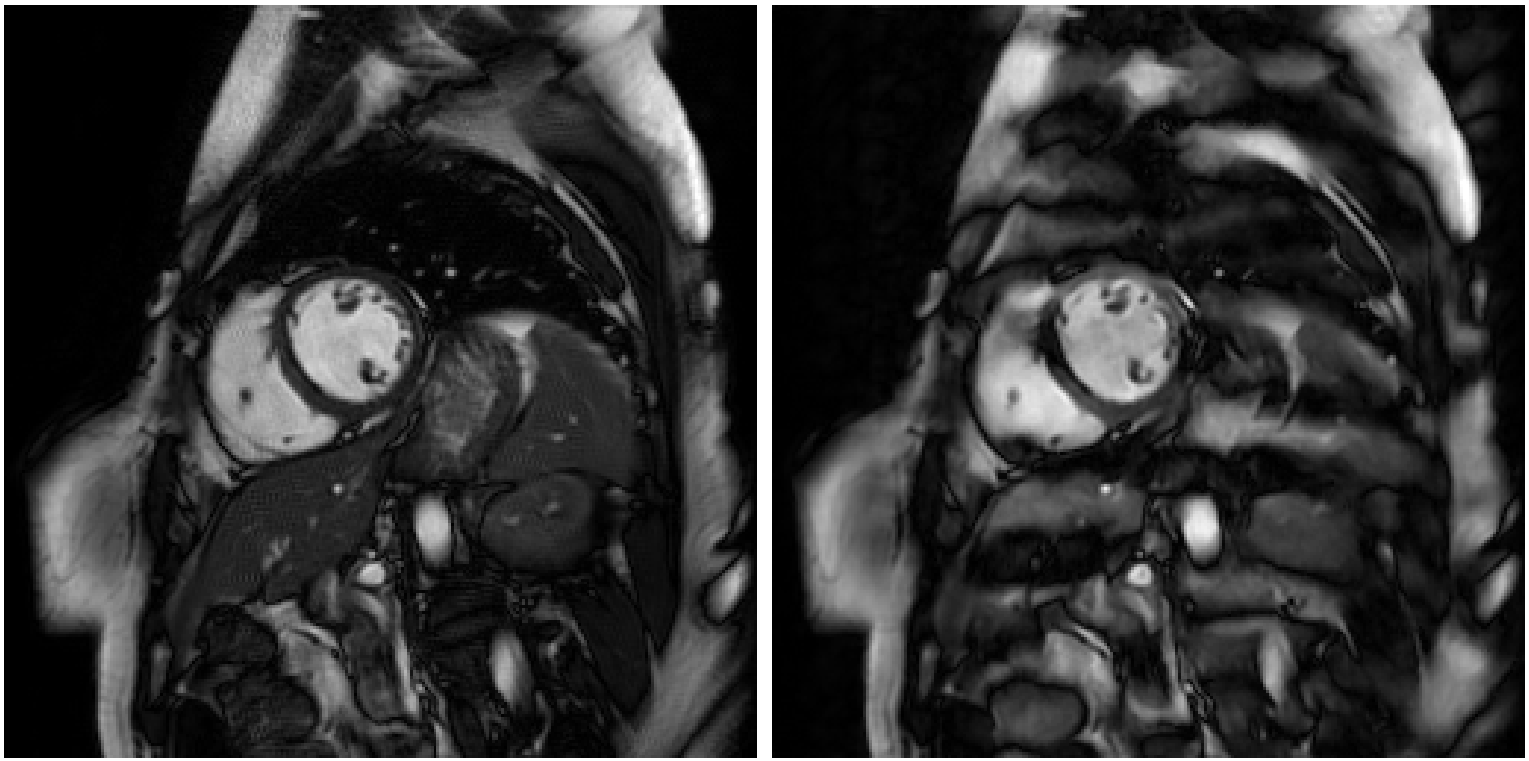


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

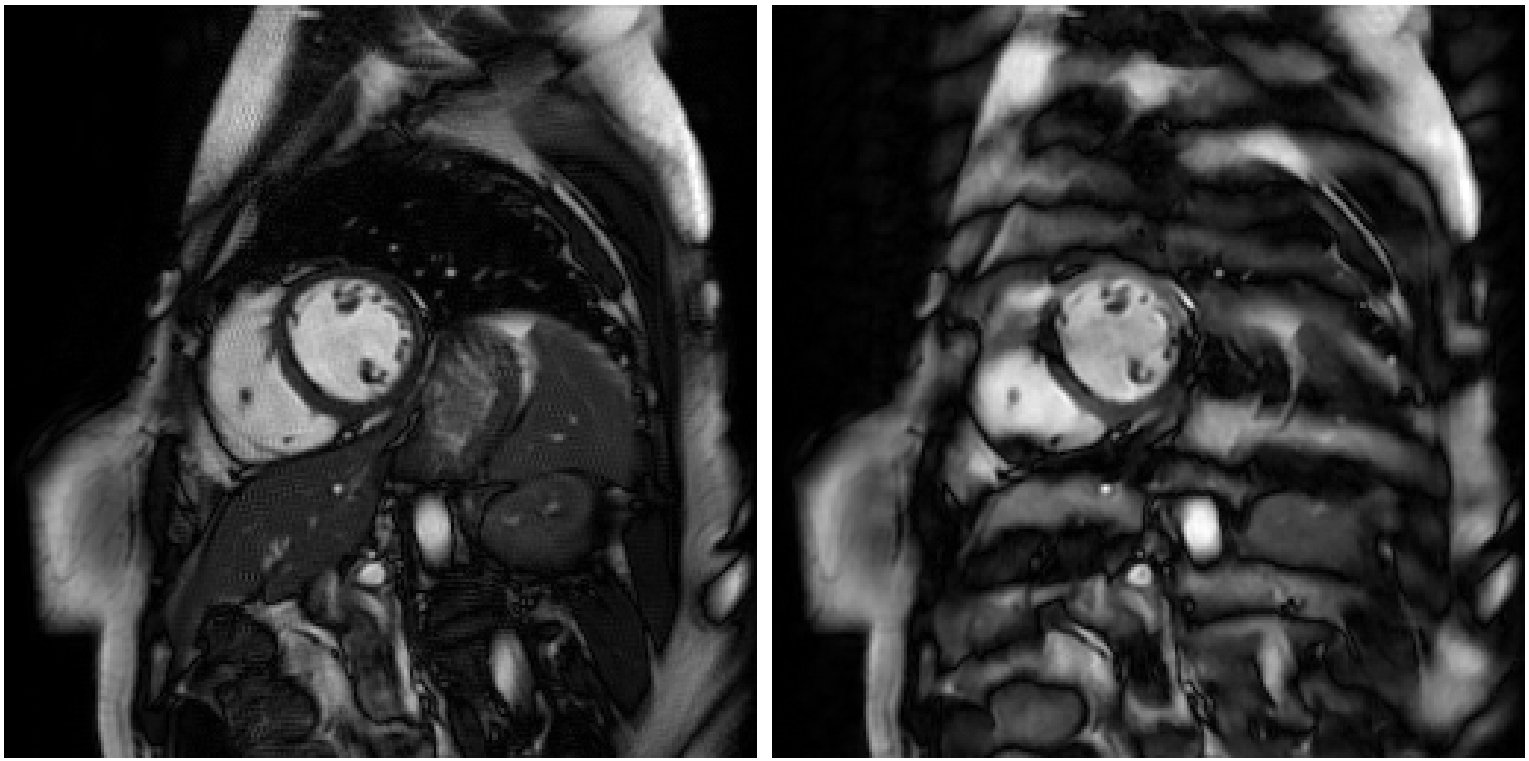


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities of deep learning in inverse problems

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

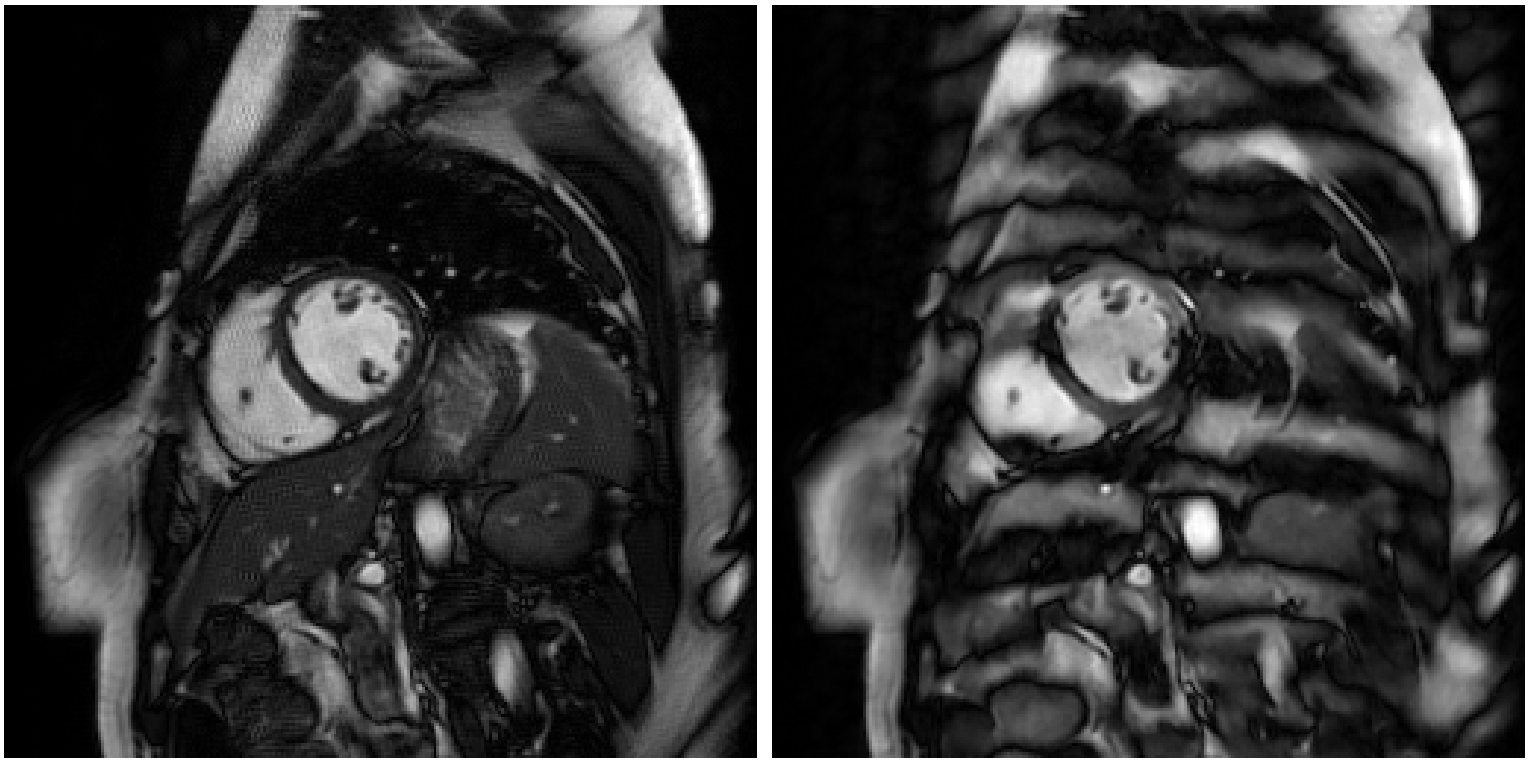


Figure : Left: Original image. Right: Deep learning reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

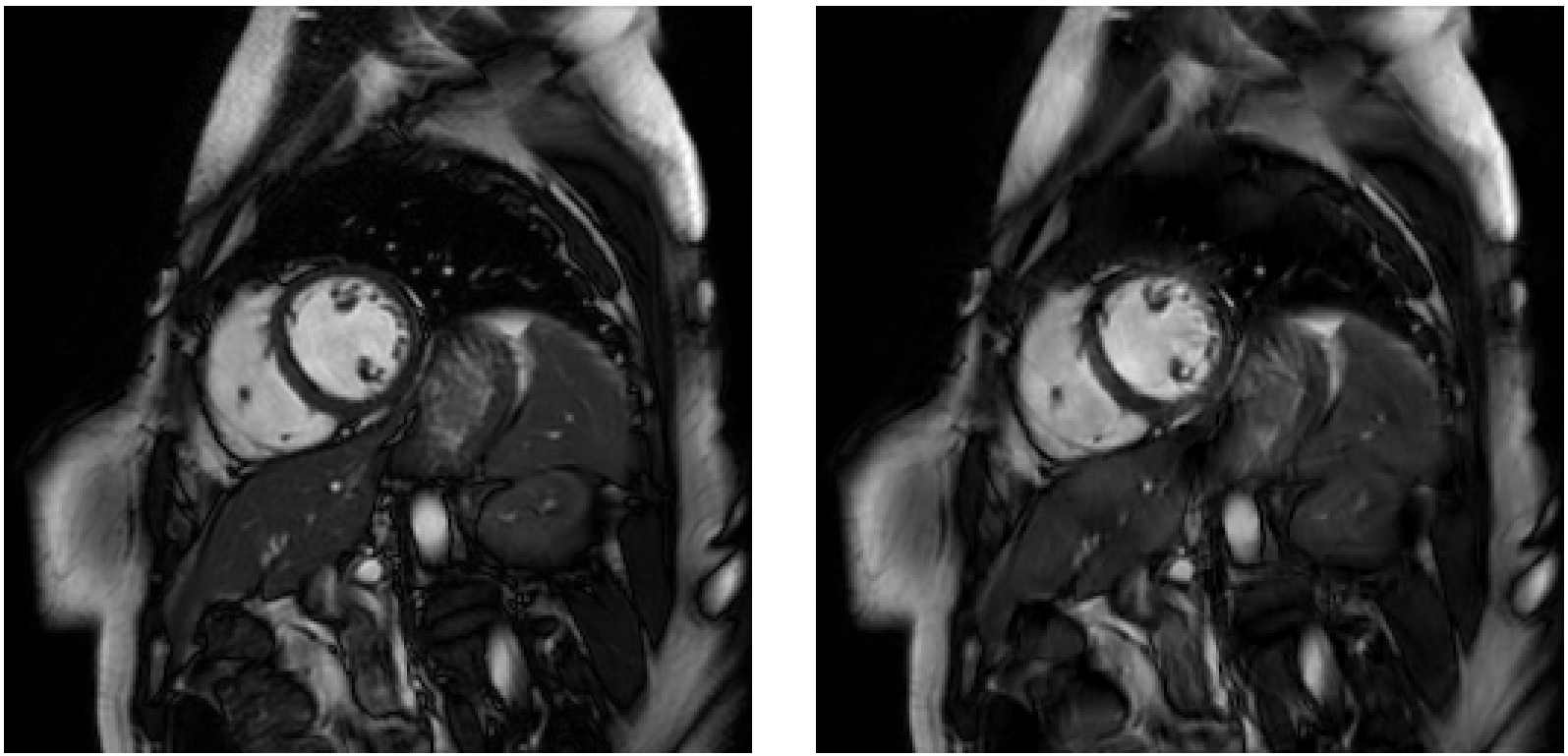


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

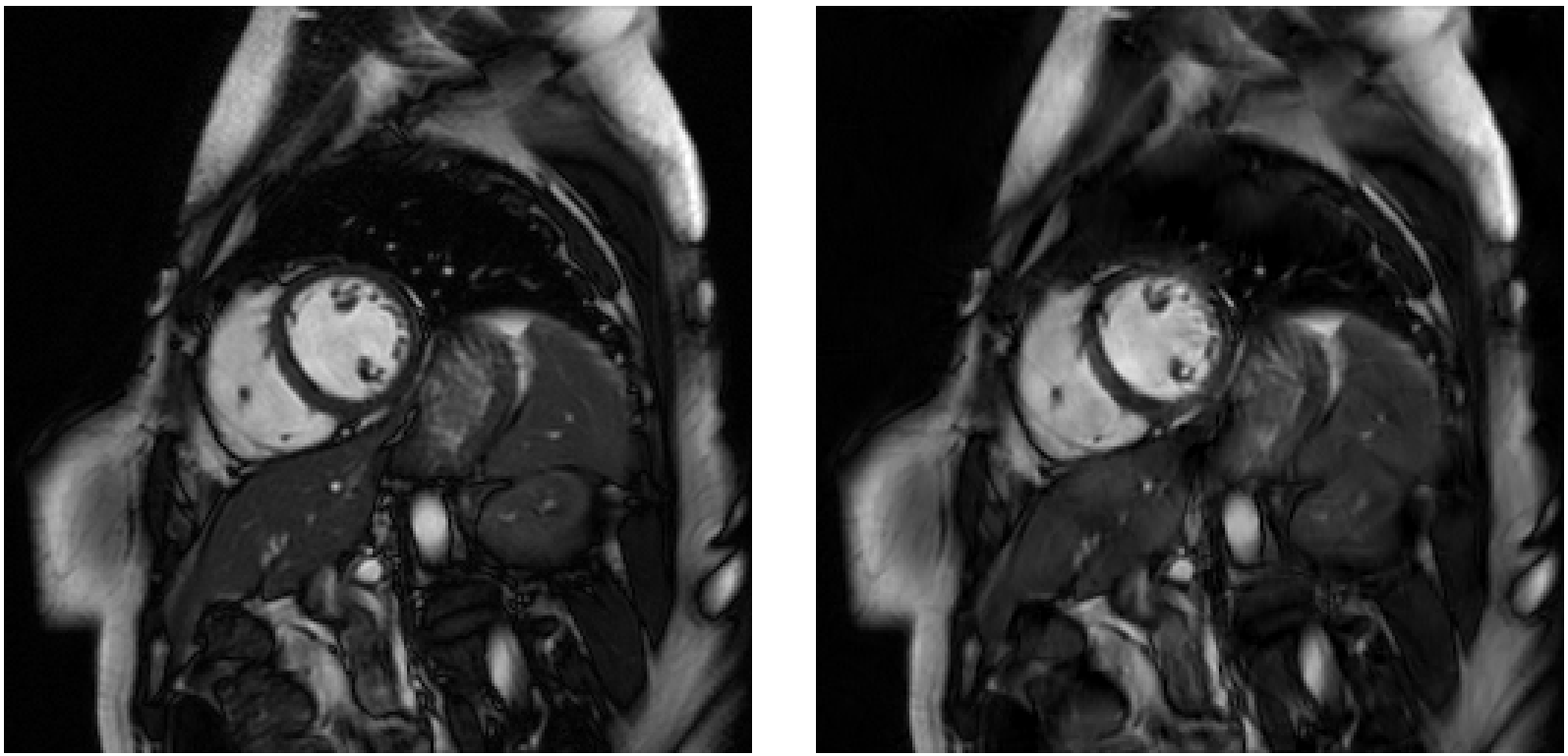


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

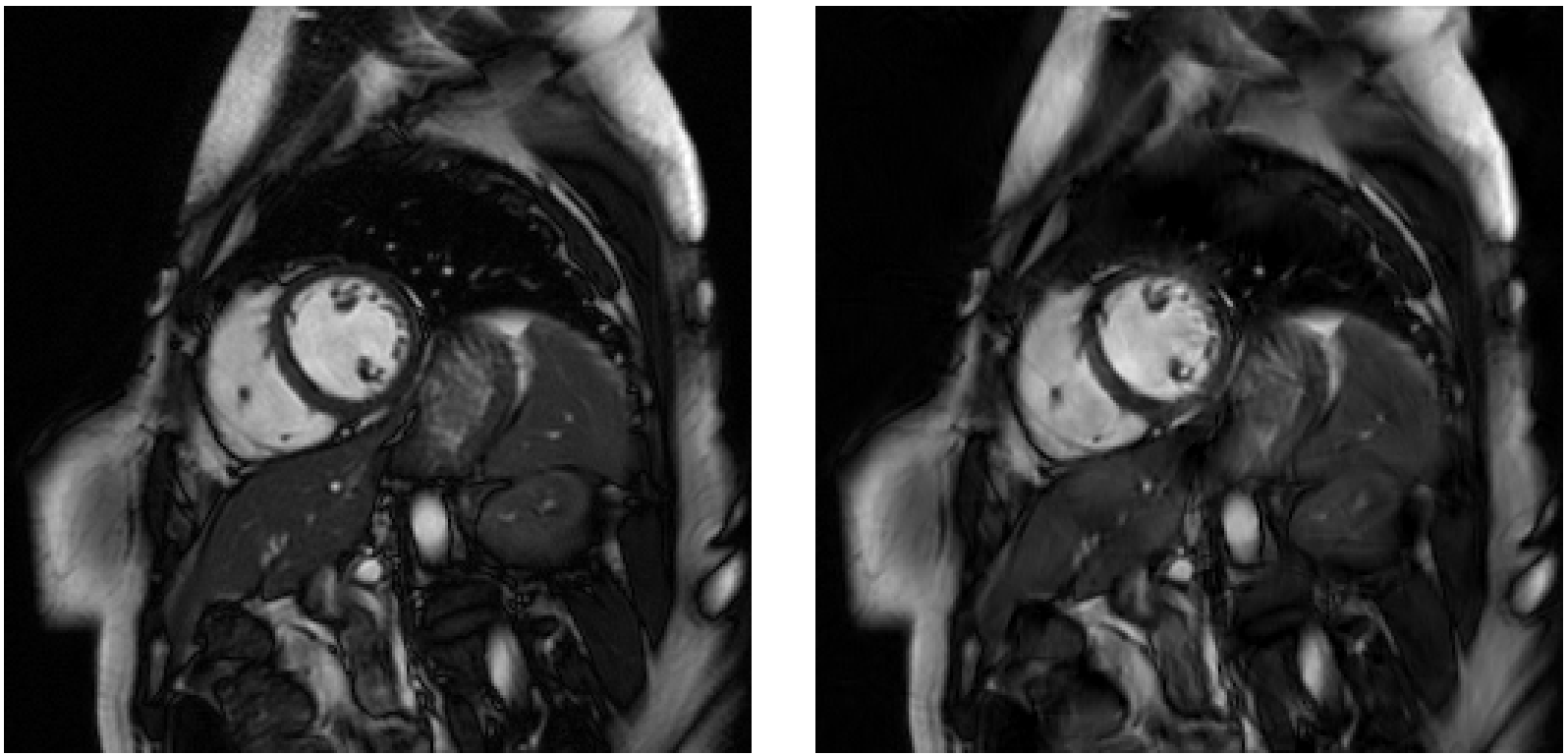


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

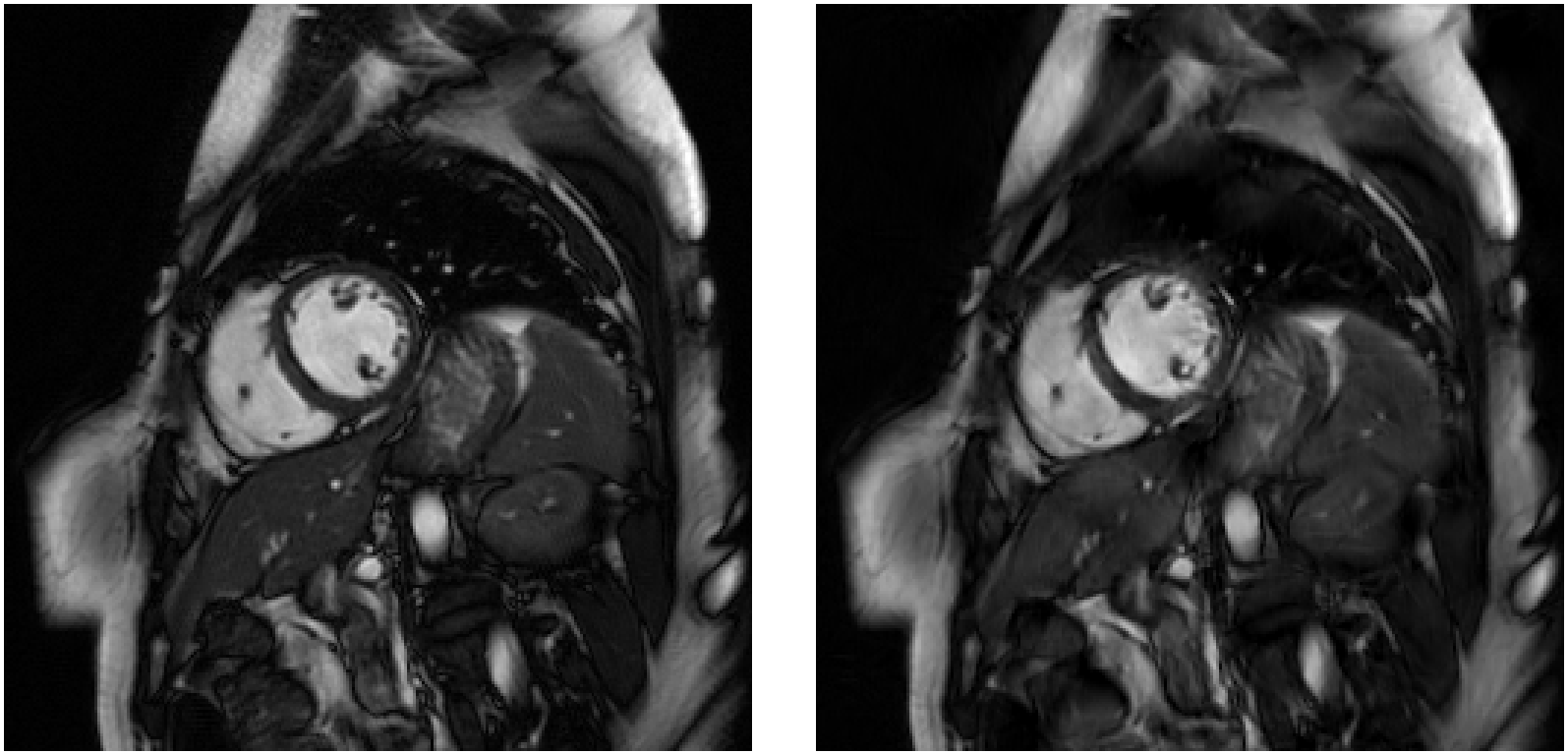


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

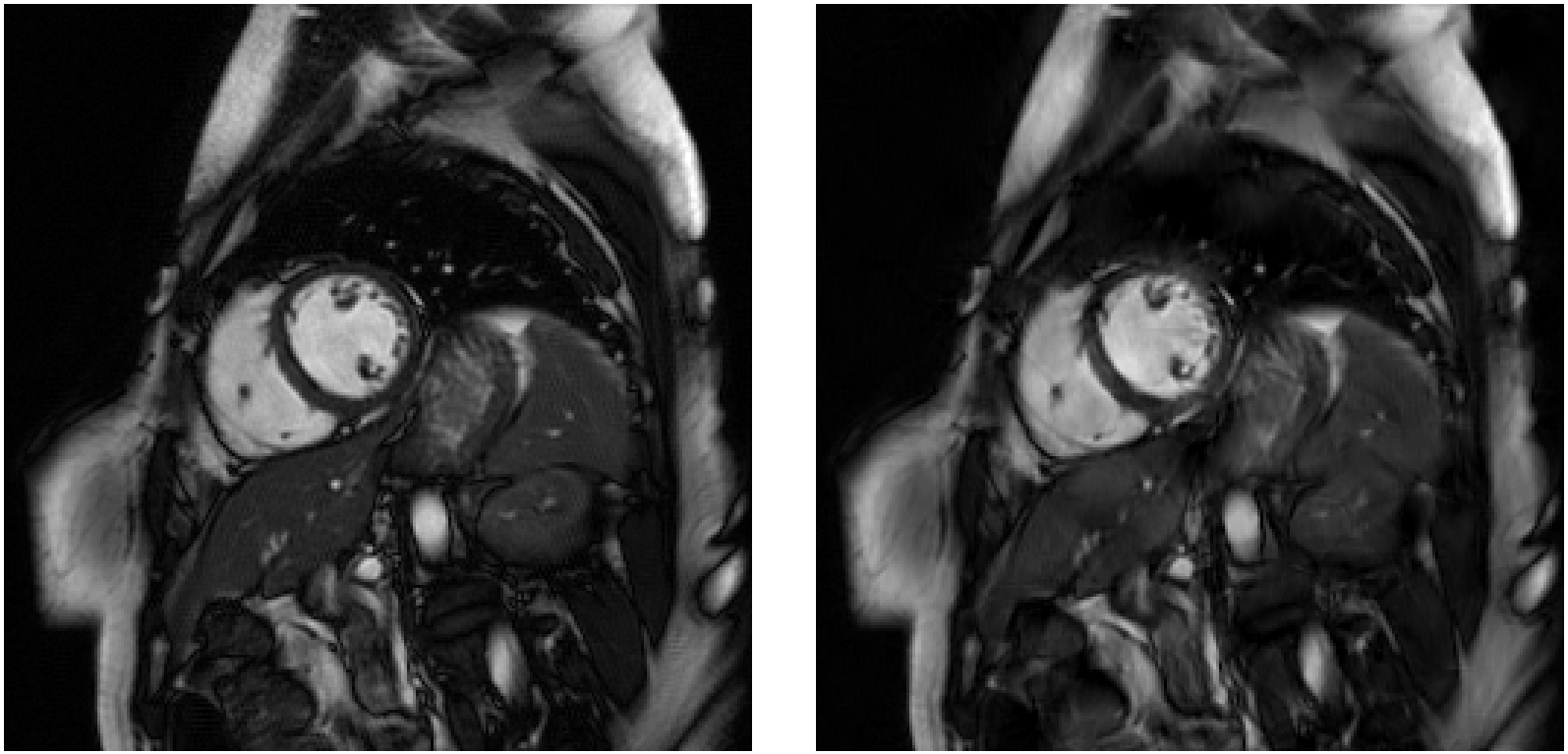


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

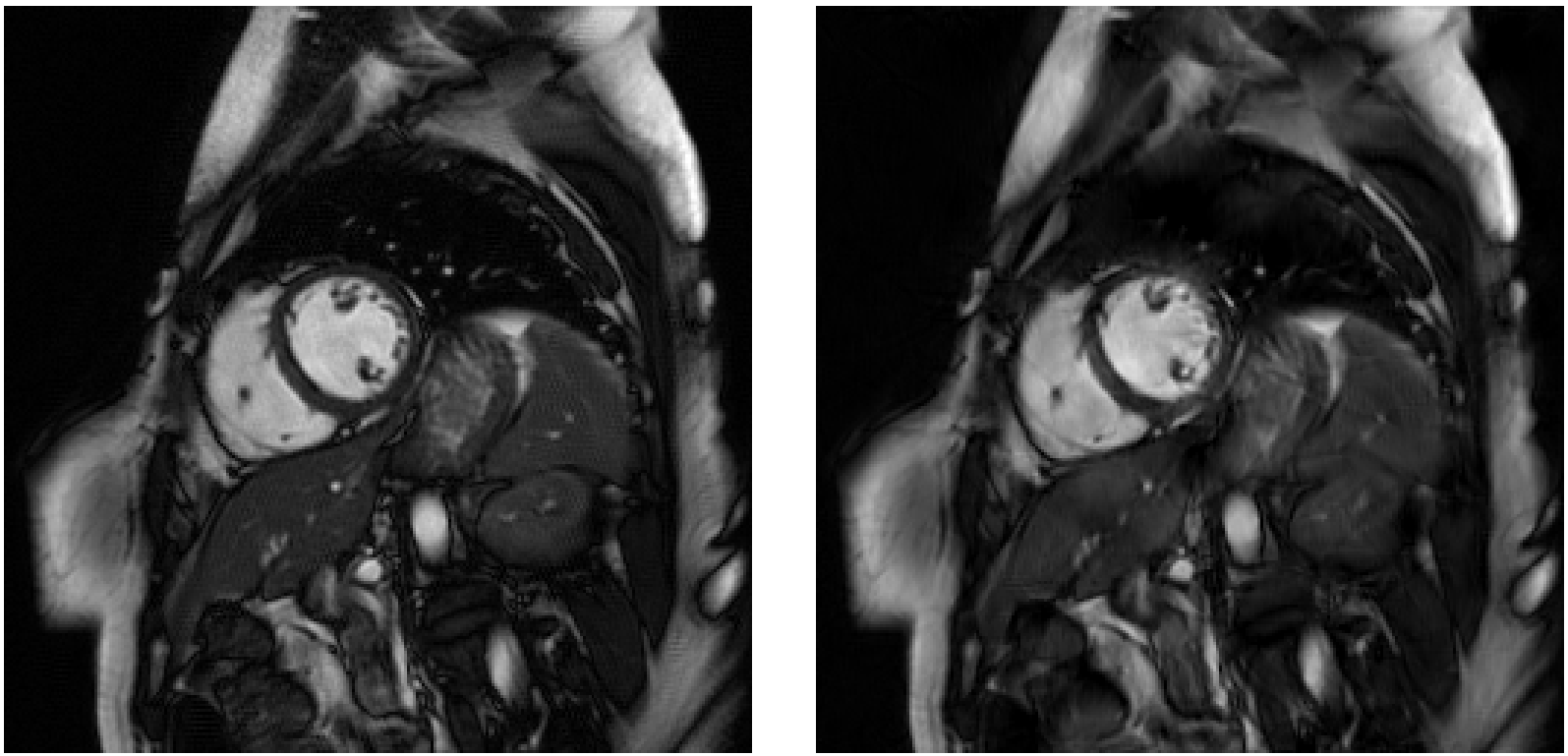


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

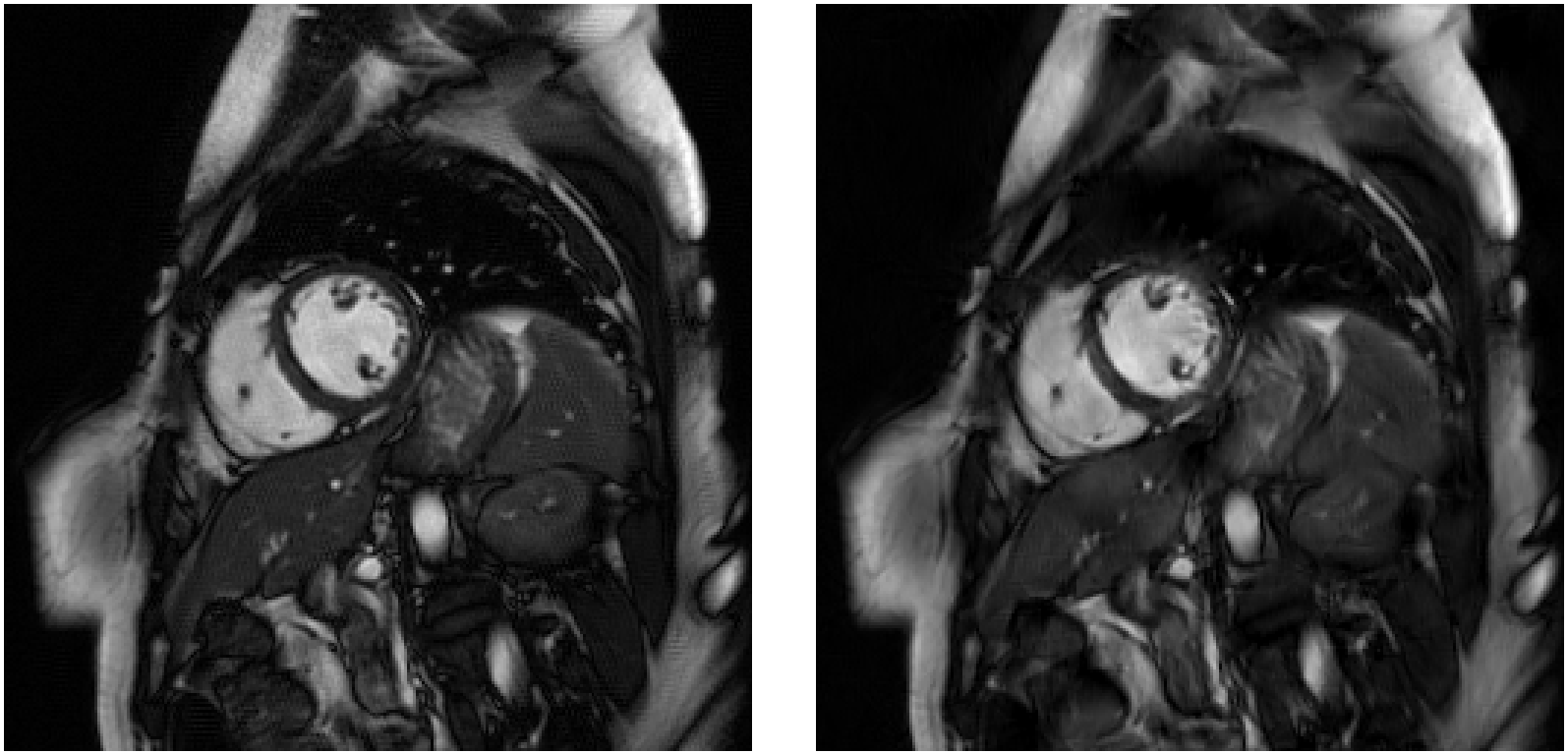


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

Repeating the experiment with compressed sensing

Experiment from "On stability and instability of deep learning in inverse problems", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen (coming)

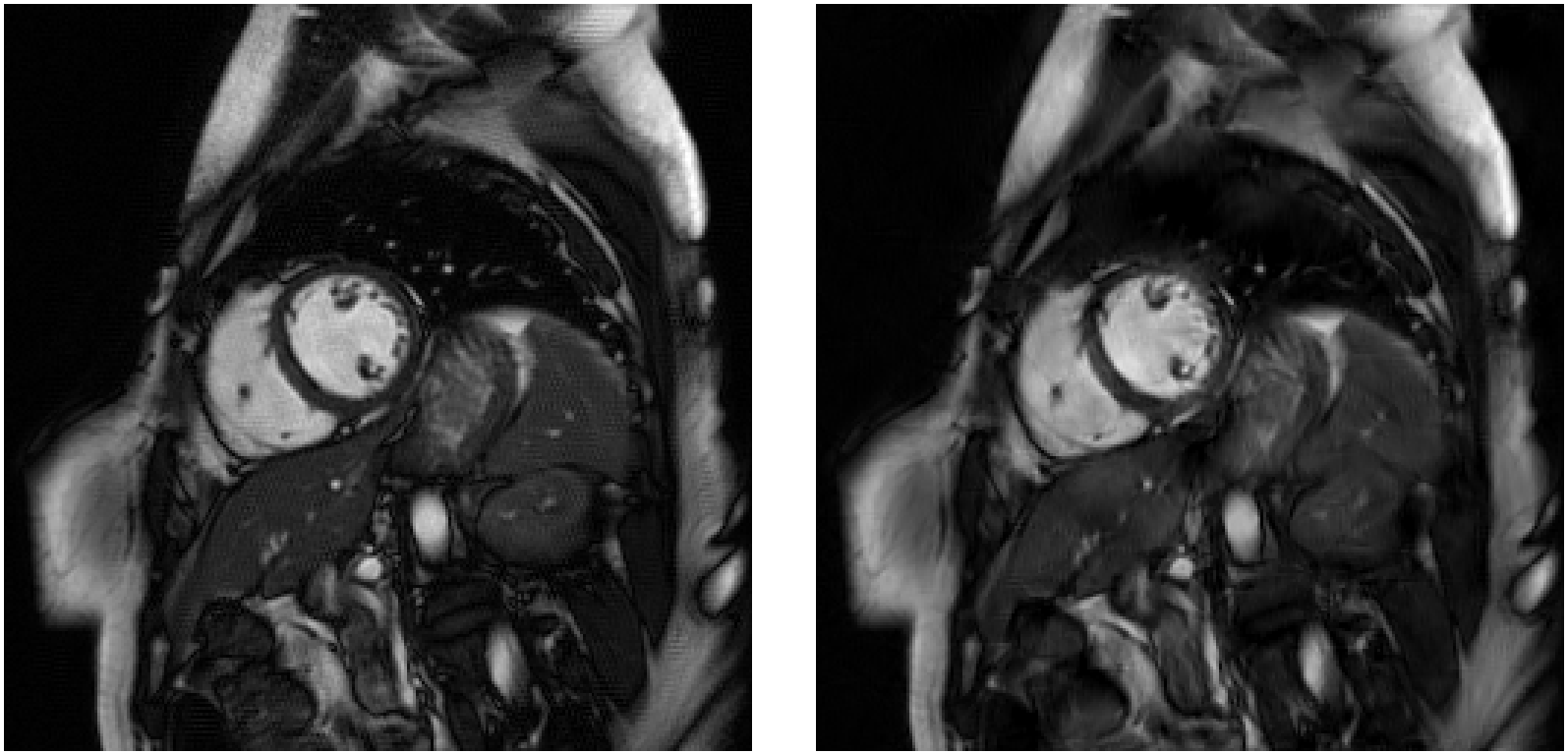


Figure : Left: Original image. Right: Compressed sensing reconstruction (25 % subsampling).

DL in Inverse Problems: 1st Step (Experiments)

CT experiment with deep learning - capturing details

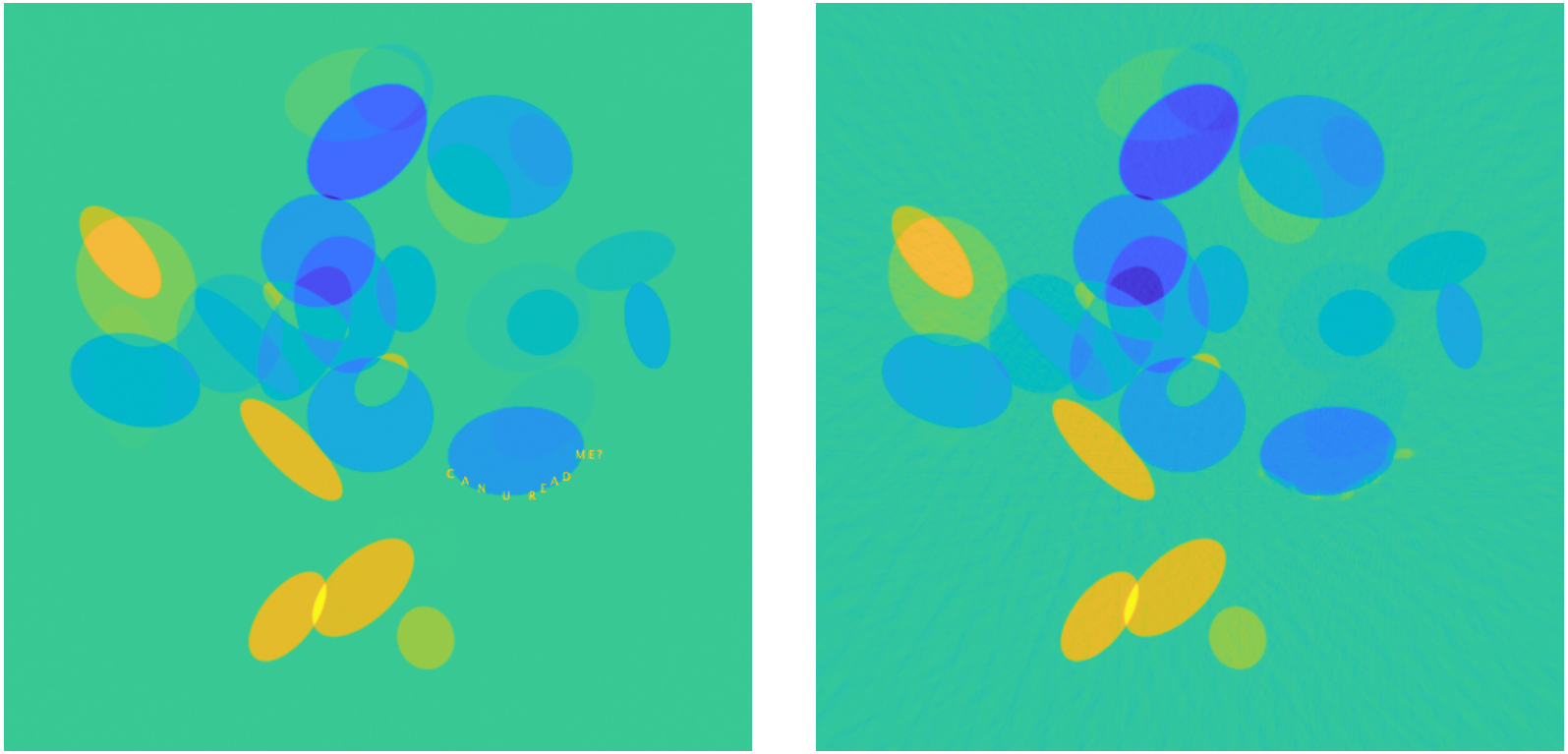


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.

DL in Inverse Problems: 1st Step (Experiments)

CT experiment with deep learning - capturing details

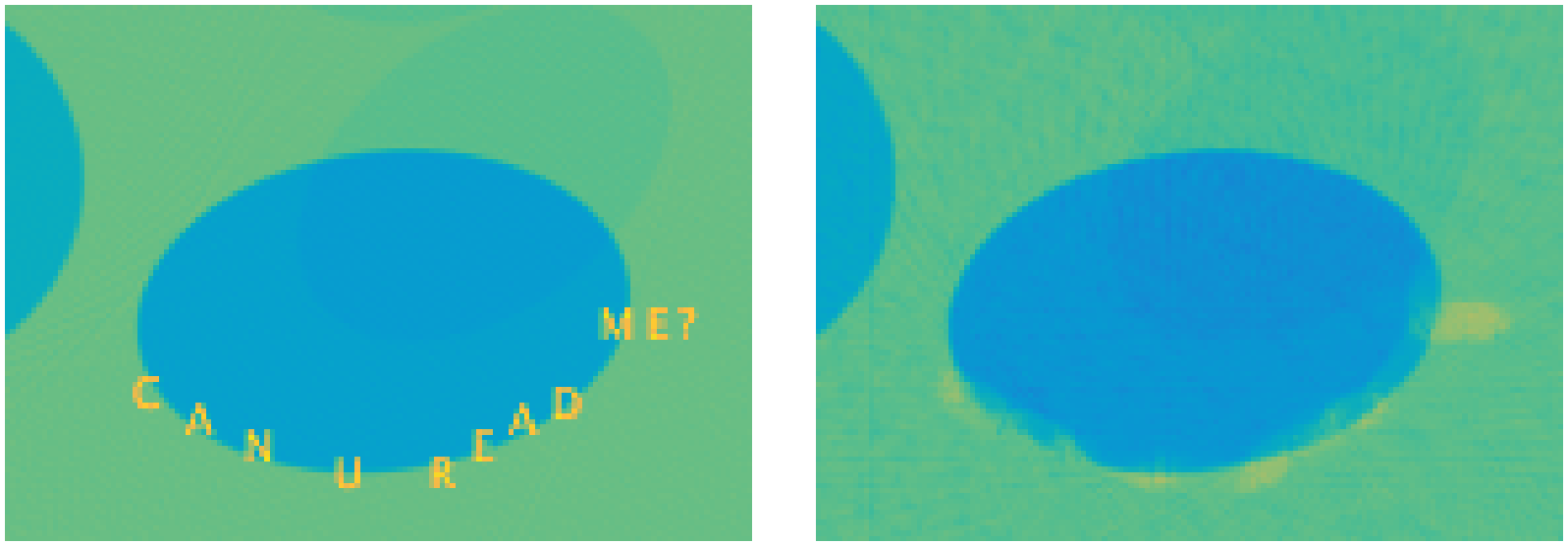


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.

Repeating the experiment with compressed sensing

CT experiment with compressed sensing - capturing details

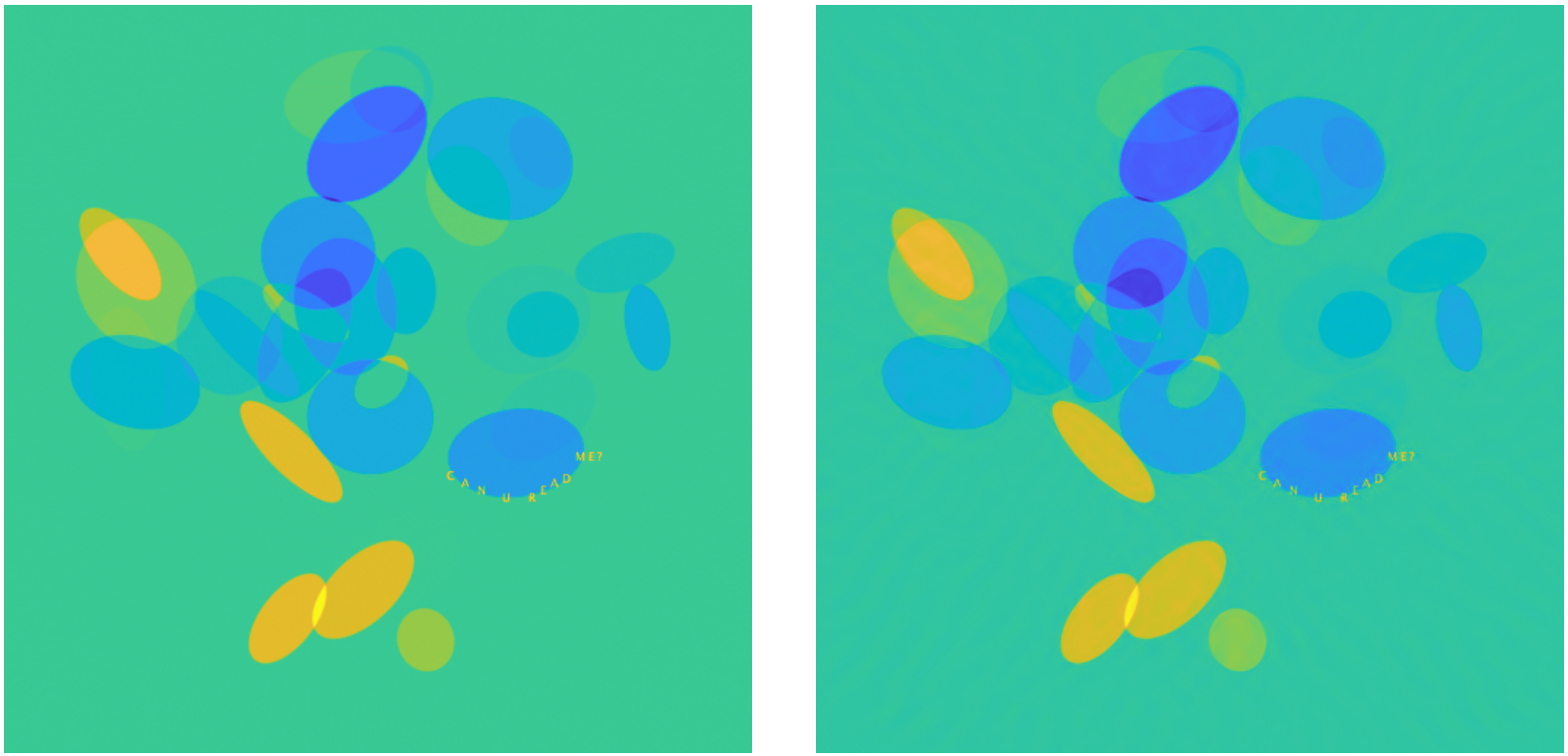


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.

Repeating the experiment with compressed sensing

CT experiment with compressed sensing - capturing details

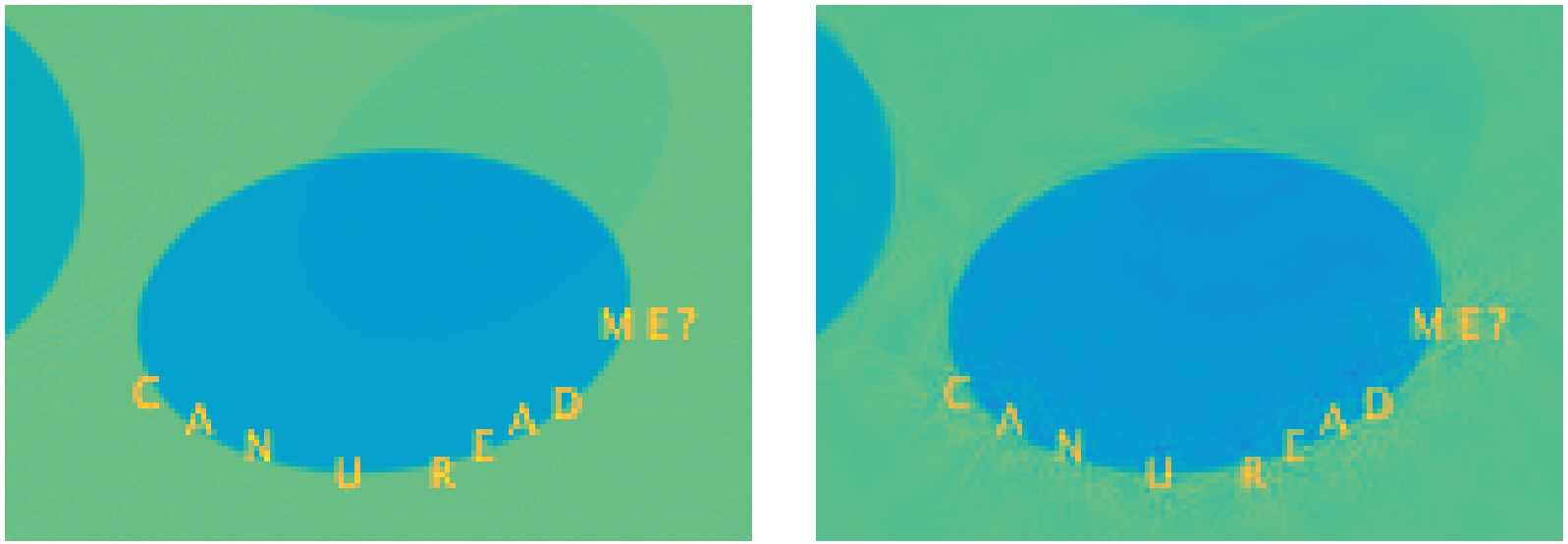


Figure : Left: Original image (with small detail added). Right: Reconstruction from Radon transform with 50 lines using neural network.