# Image Denoising

## Learning Sparse Representations For Image and Signal Modeling

Giacomo Boracchi

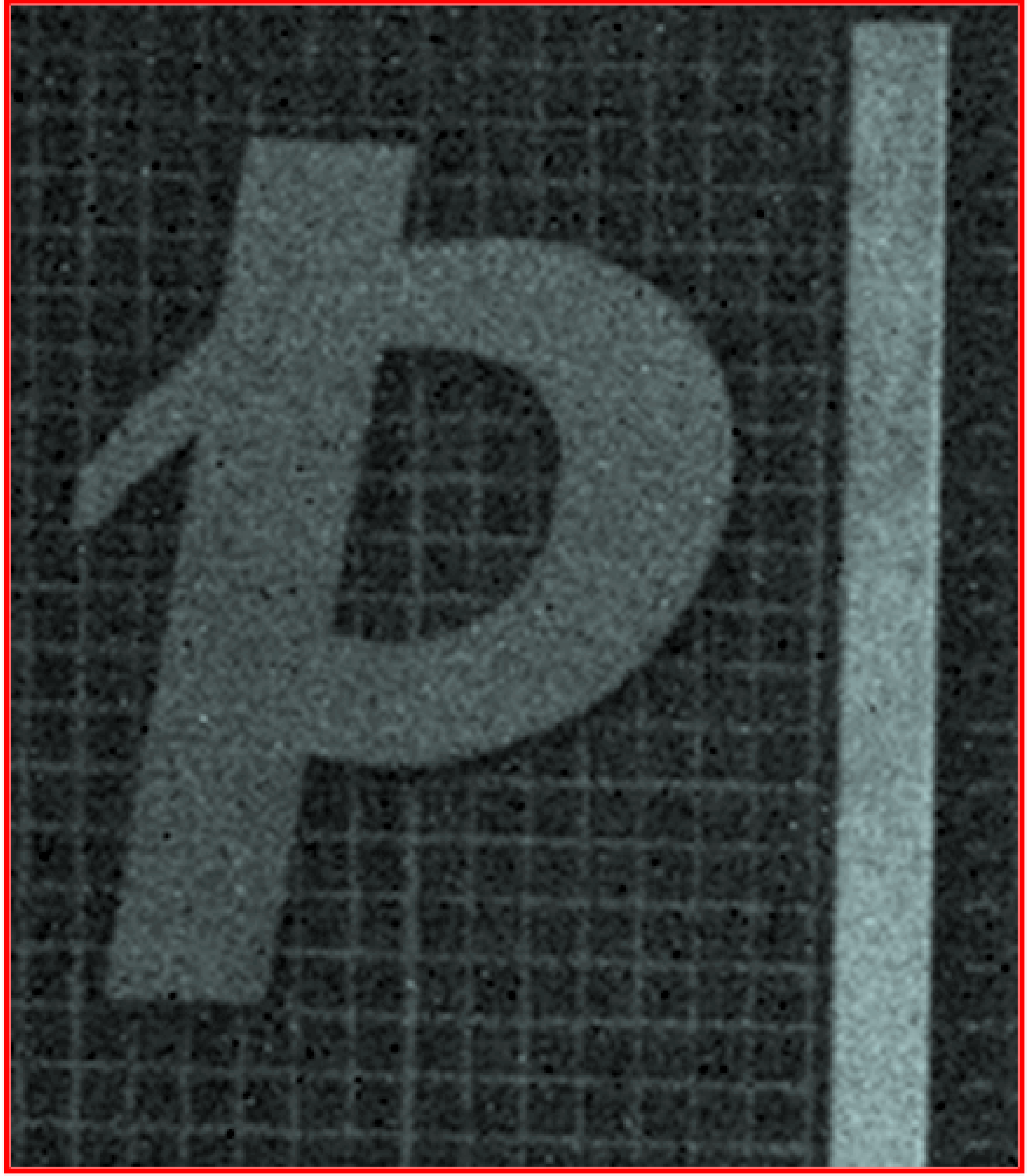
https://boracchi.faculty.polimi.it/

May 17th 2023

# Problem Formulation

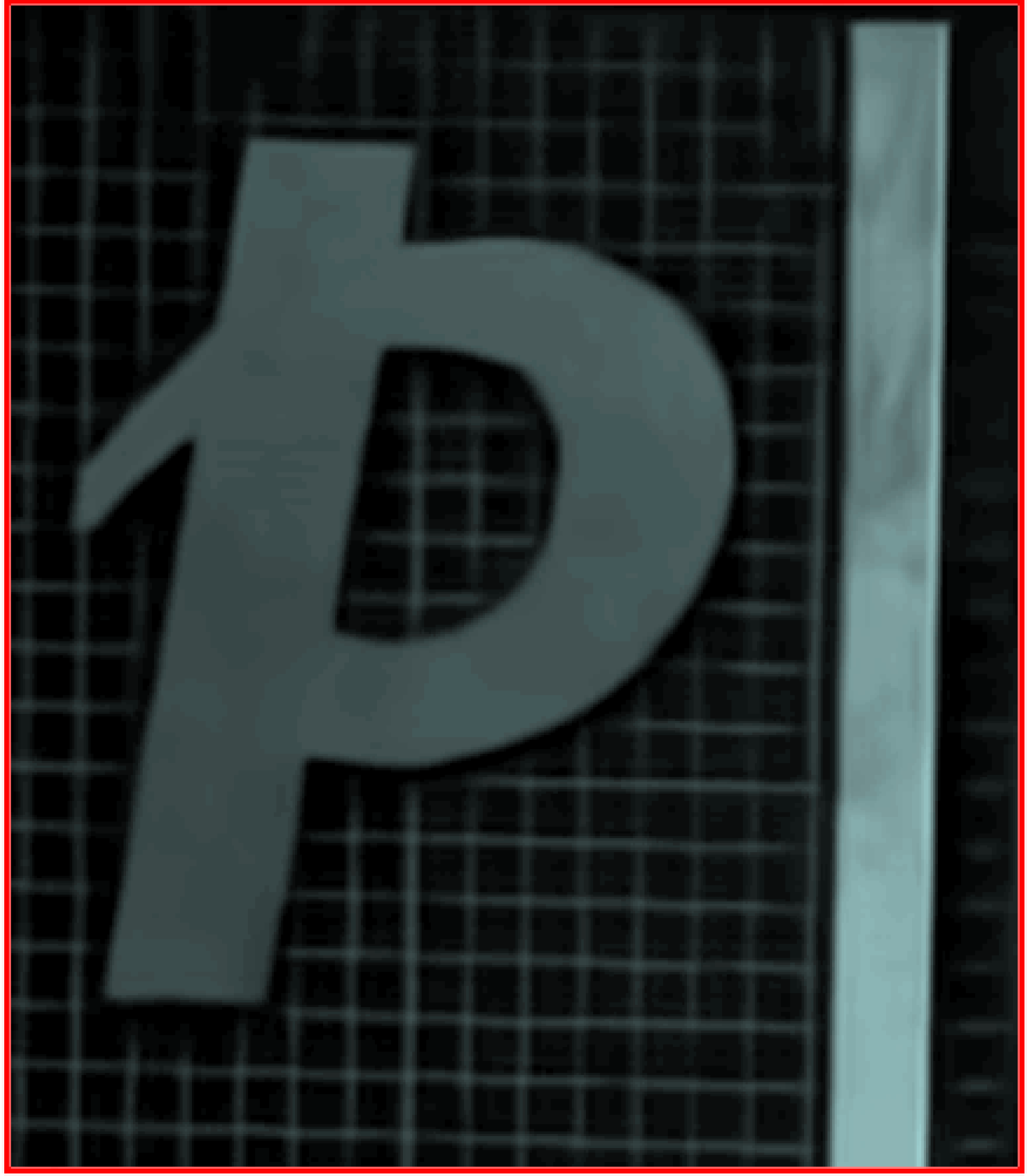
# Denoising: The Issue

A Detail in Camera Raw Image $z$



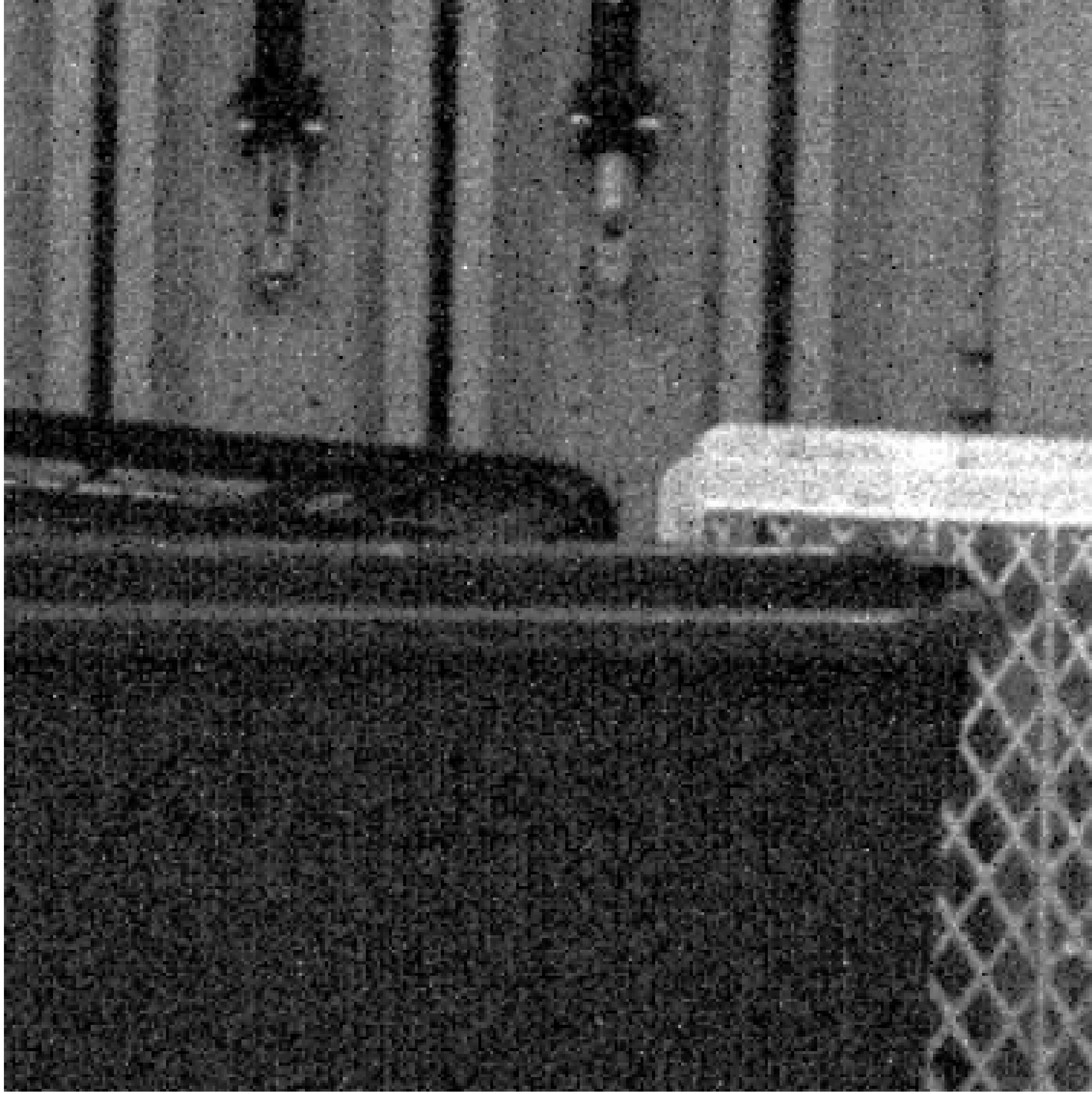
# Denoising: The Issue

Denoised $\hat{y}$



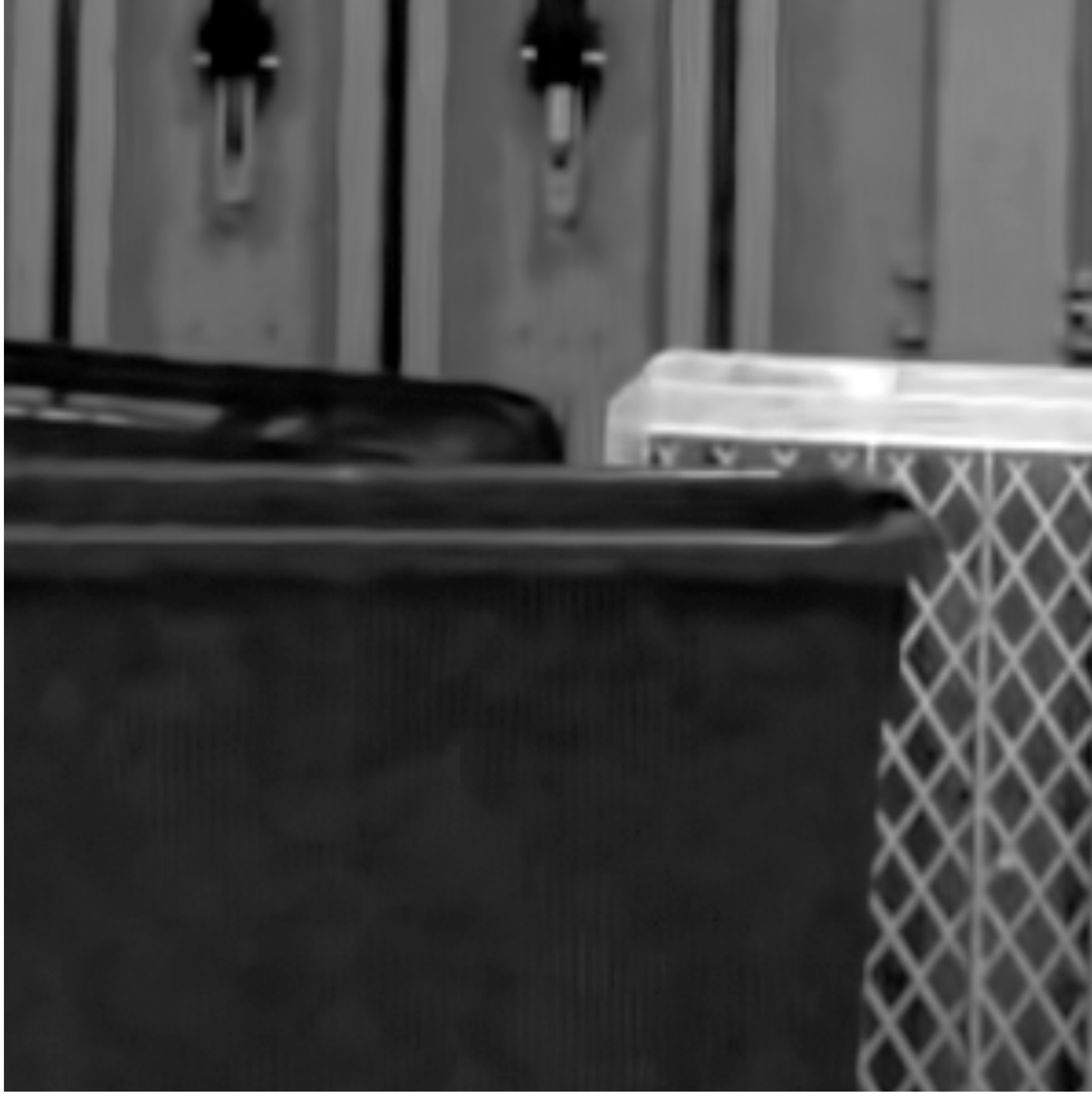
# Denoising: The Issue

A Detail in Camera Raw Image $z$



# Denoising: The Issue

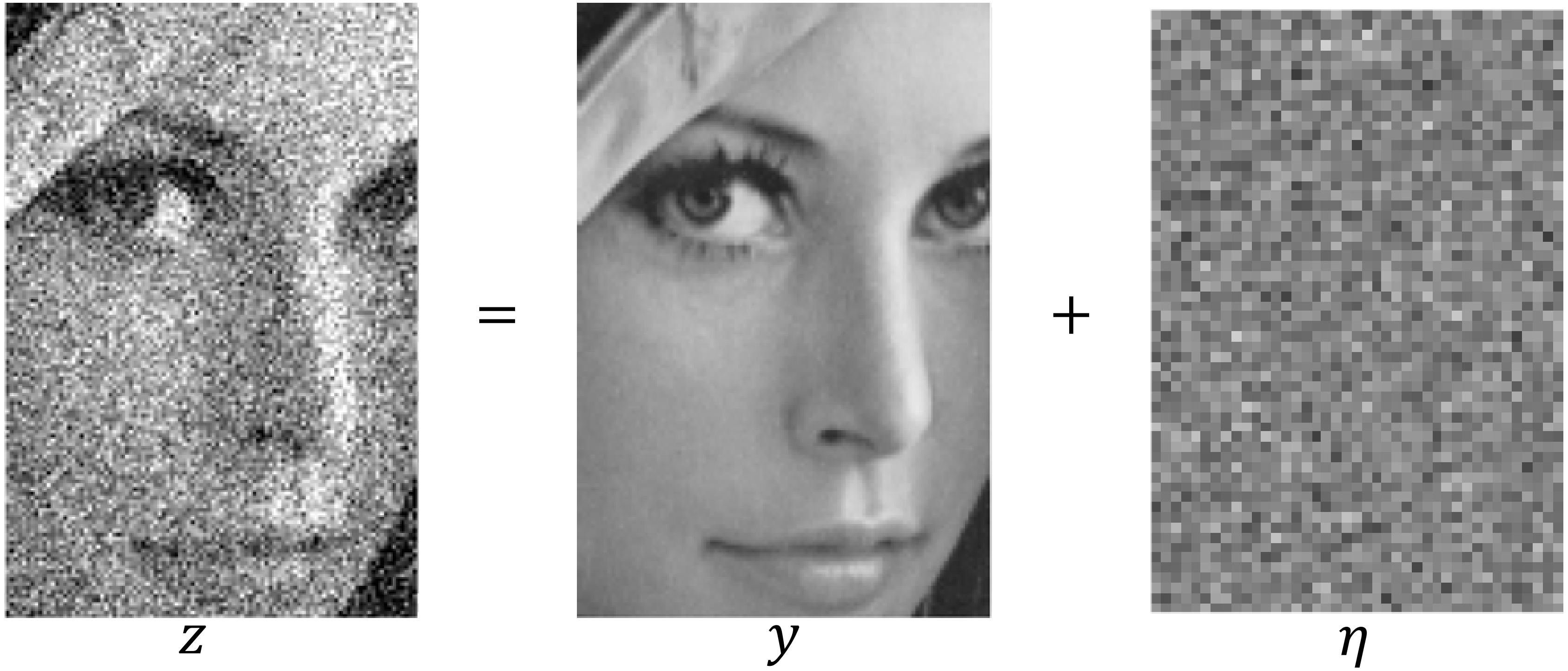
Denoised $\hat{y}$



# Image Formation Model

Observation model is

$$z(x) = y(x) + \eta(x), \qquad x \in \mathcal{X}$$



# Image Formation Model

Observation model is

$$z(x) = y(x) + \eta(x), \qquad x \in \mathcal{X}$$

Where

- $x$ denotes the pixel coordinates in the domain $\mathcal{X} \subset \mathbb{Z}^2$
- $y$ is the original (noise-free and unknown) image, $y \in [0,1]$
- $z$ is the noisy observation, $\mathrm{z} \in [0,1]$ (clipping)
- $\eta$ is the noise realization

**For the sake of simplicity** we assume Additive White Gaussian Noise (AWGN):

$\eta \sim N(0, \sigma^2)$ and $\eta(x)$ are all independent realizations.

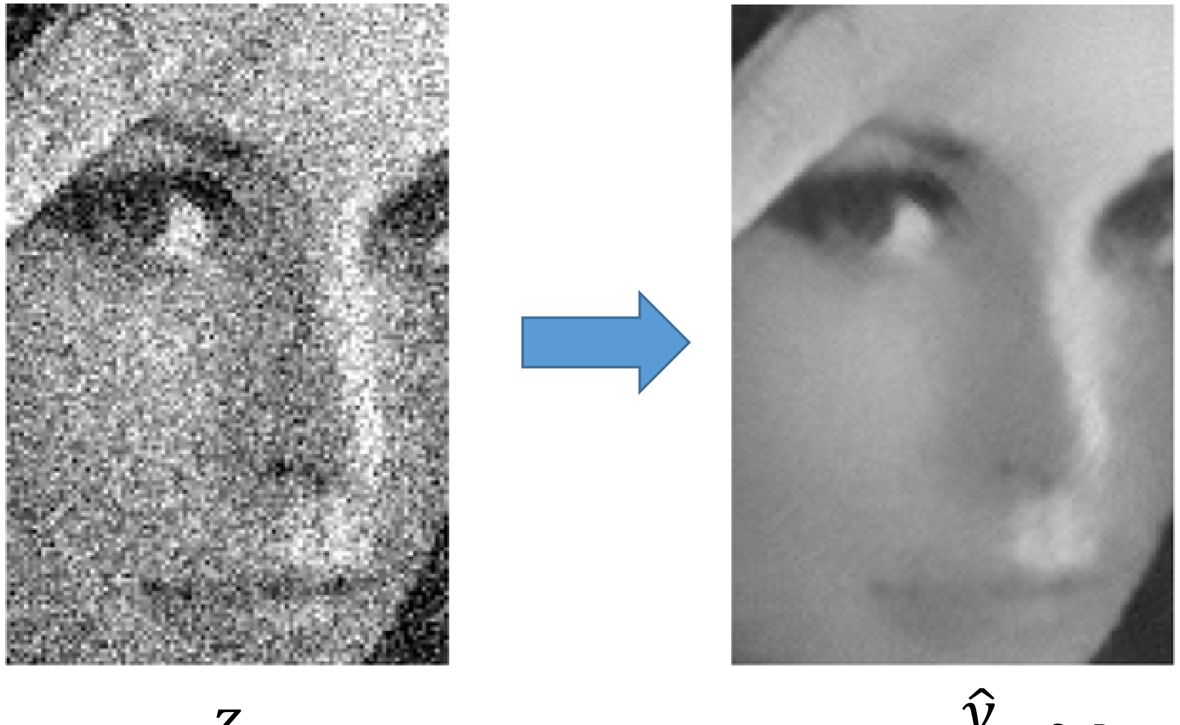
The noise standard deviation $\sigma$ is also assumed as known.

# Goal of Image Denoising

The goal of **image denoising** is to compute $\hat{y}$ *realistic* estimate of the original image $y$, given the noisy observation $z$

Denoising is an **ill posed problem** and requires some form of **regularization** to promote outputs that are close to natural images.

Our Prior: **Sparsity w.r.t. DCT basis!**

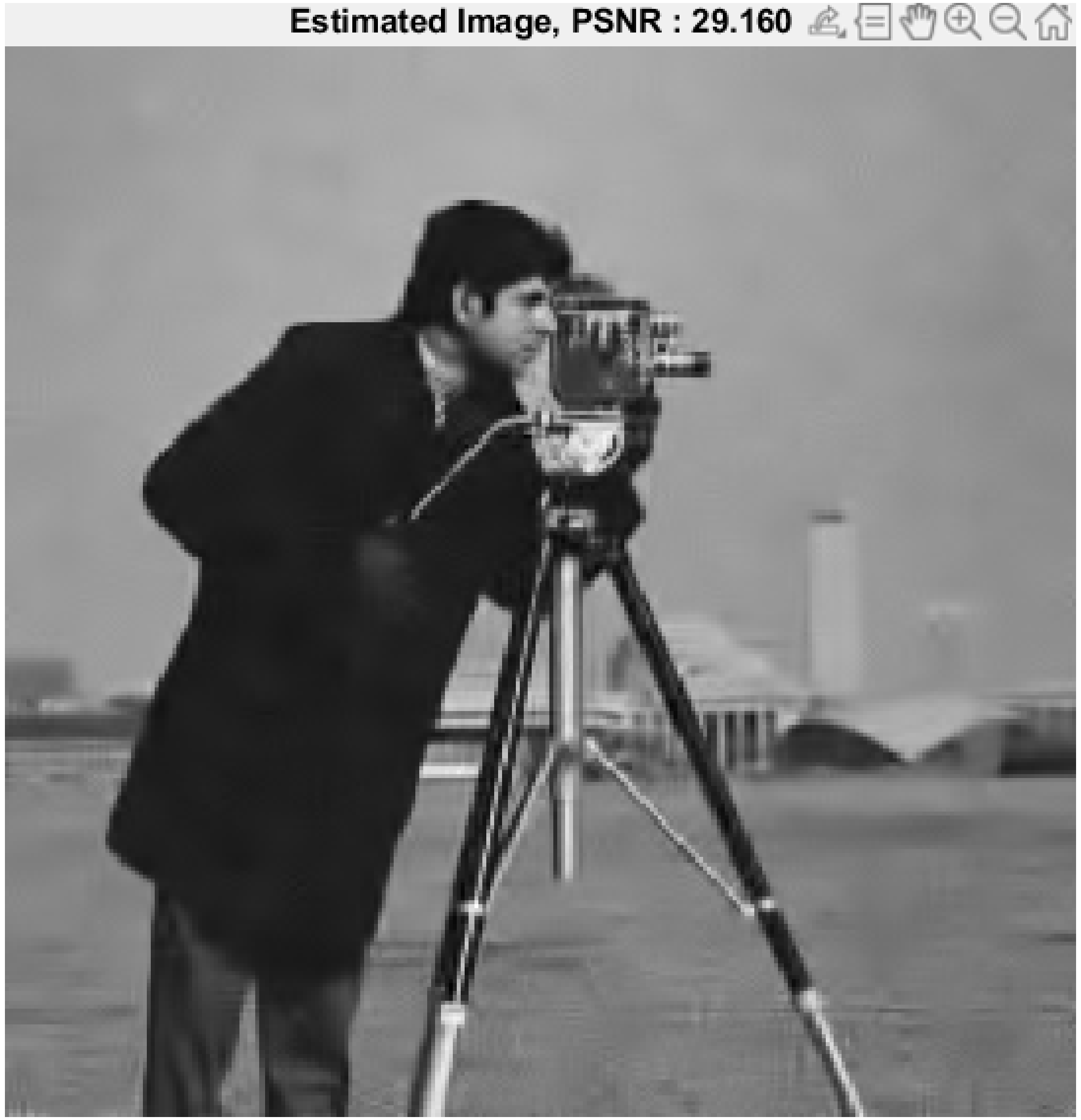


# Image Denoising

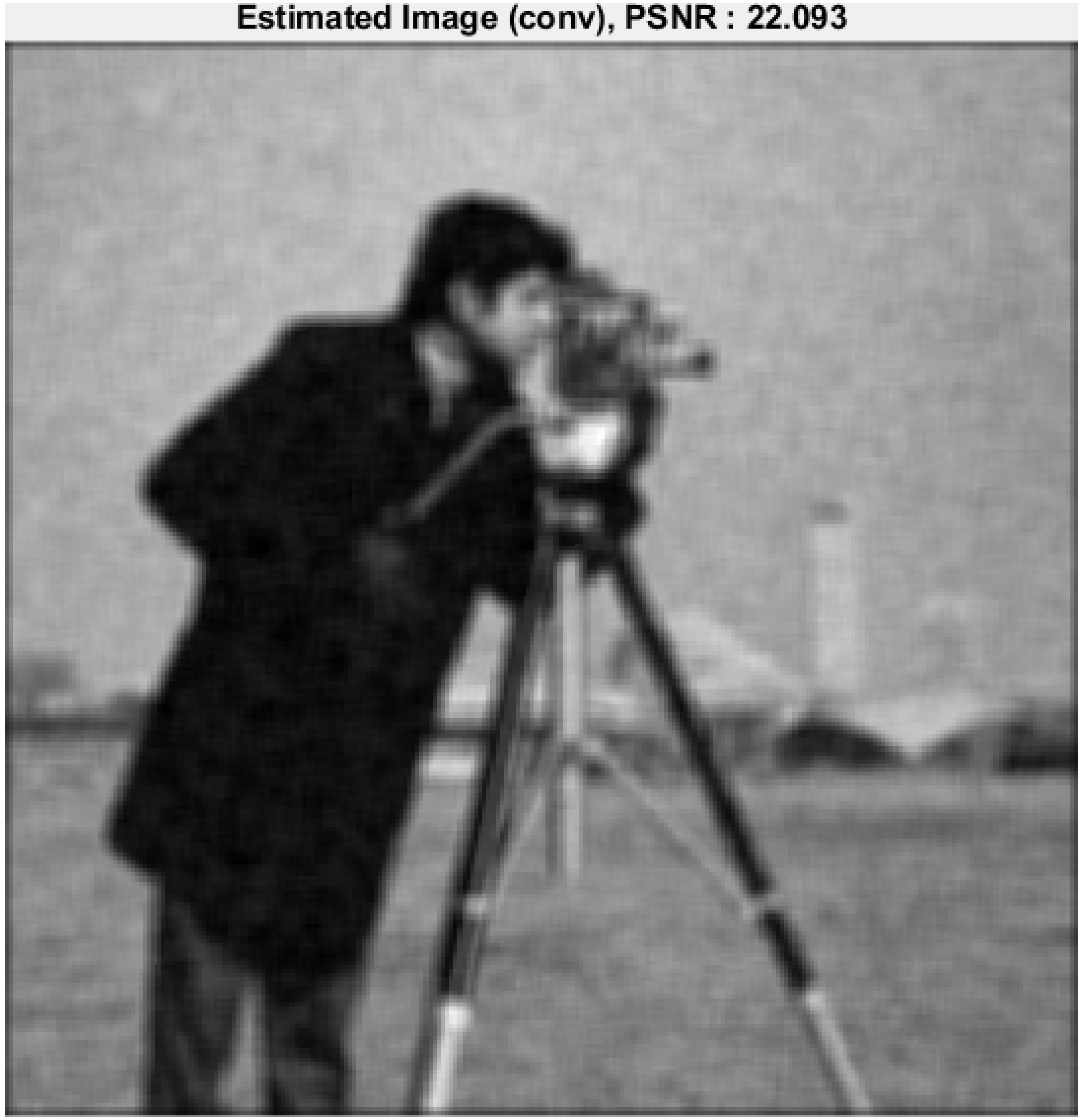
Deniosing is a fundamental step in image processing pipelines

- Improves the quality of digital images to the standard we are used to
- Eases the following algorithms in imaging pipelines from those solving low-level (e.g., edge detection), till high-level (recognition) problems
- It is also a tool to quantitativelly assess the performance of a descriptive model for images.

# DCT Denoising



# Denoising by Convolution



G. Boracchi

# Image Denoising By Sparsity Priors

# Sliding DCT Denoising

A very powerful, yet simple denoising algorithm that can pair much more sophisticated alternatives

A description of the algorithm steps can be found here

**Yu, Guoshen, and Guillermo Sapiro. "DCT image denoising: a simple and effective image denoising algorithm."** ***Image Processing On Line*** **1 (2011): 292-296.**

https://www.ipol.im/pub/art/2011/ys-dct/article.pdf

# Assignment

Sliding DCT

# Implement DCT denoising on a natural image

- Load the cameraman image
- Add additive white Gaussian noise having standard deviation $\sigma$
- For each patch over a tile, perform denoising in the DCT domain use $\tau = 3\sigma$ or $\tau = \sigma\sqrt{2\,\ln p^2}$ as in **[Donoho & Johnstone]**
- Remember **not to threshold the DC coefficient, which contains the average patch intensity**

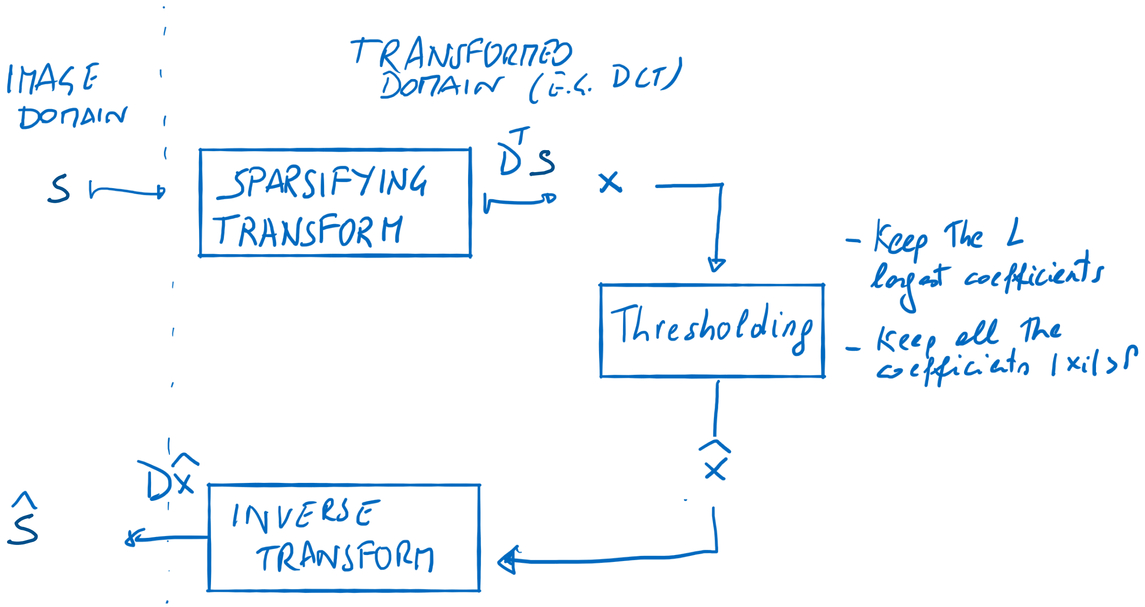


IMAGE DOMAIN
TRANSFORMED DOMAIN (E.G. DCT)
$s$
SPARSIFYING TRANSFORM
$D^T s$
$x$
Thresholding
- Keep The L largest coefficients
- Keep all The coefficients $|x_i| > \Gamma$
$\hat{x}$
INVERSE TRANSFORM
$D\hat{x}$
$\hat{s}$

- Reconstruct the denoised patch $\hat{s}$

D.L. Donoho, I.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage. Biometrika, vol. 83, pp. 425-455, 1994. http://dx.doi.org/10.1093/biomet/81.3.425

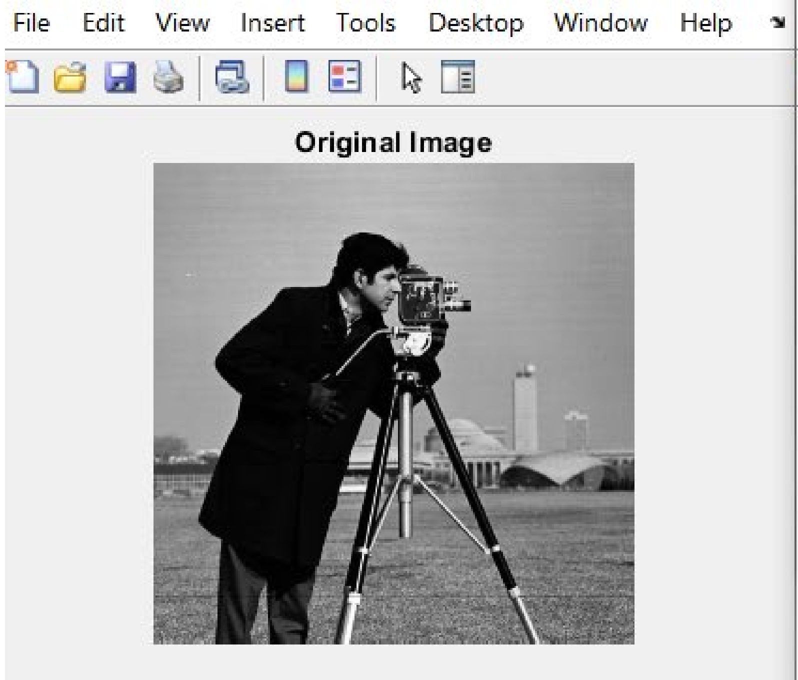
# Assess Denoising Performance

Measure the PSNR of the denoised image

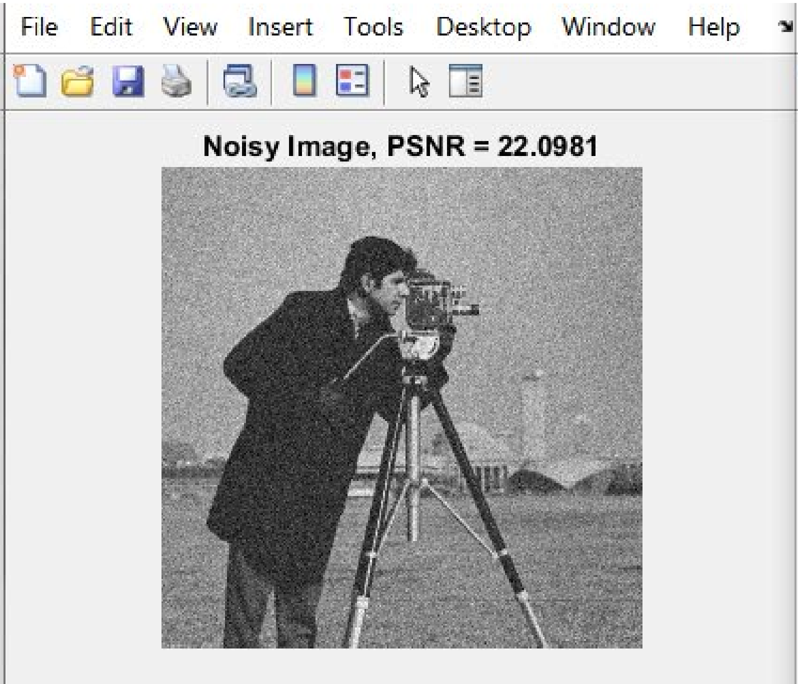
$$PSNR(\hat{y}, y) = 10\ \log_{10} \frac{1}{MSE(\hat{y}, y)}$$

Where 1  stands for the signal peak (image is assumed to be in [0,1])

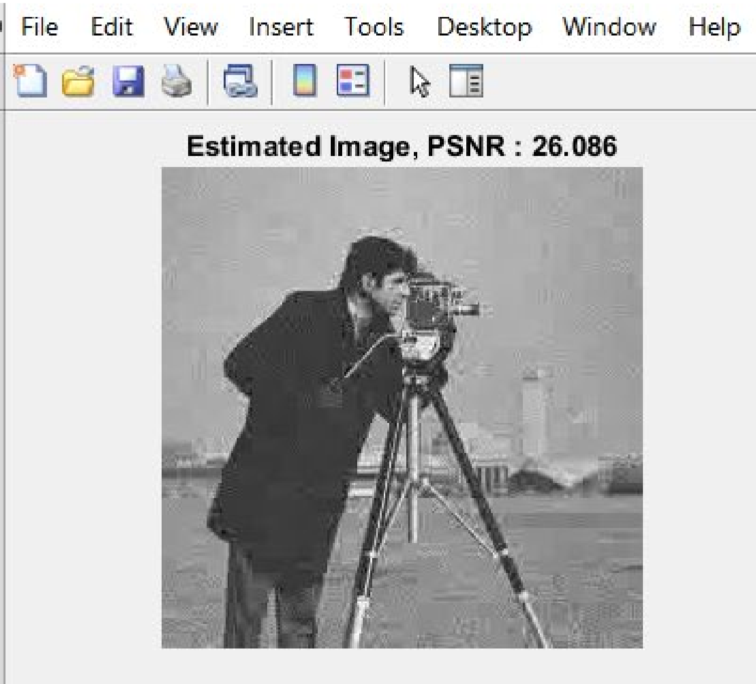
```
sigma_noise = 20/255; img = im2double(imread('cameraman.tif'));
```



Original Image



Noisy Image, PSNR = 22.0981



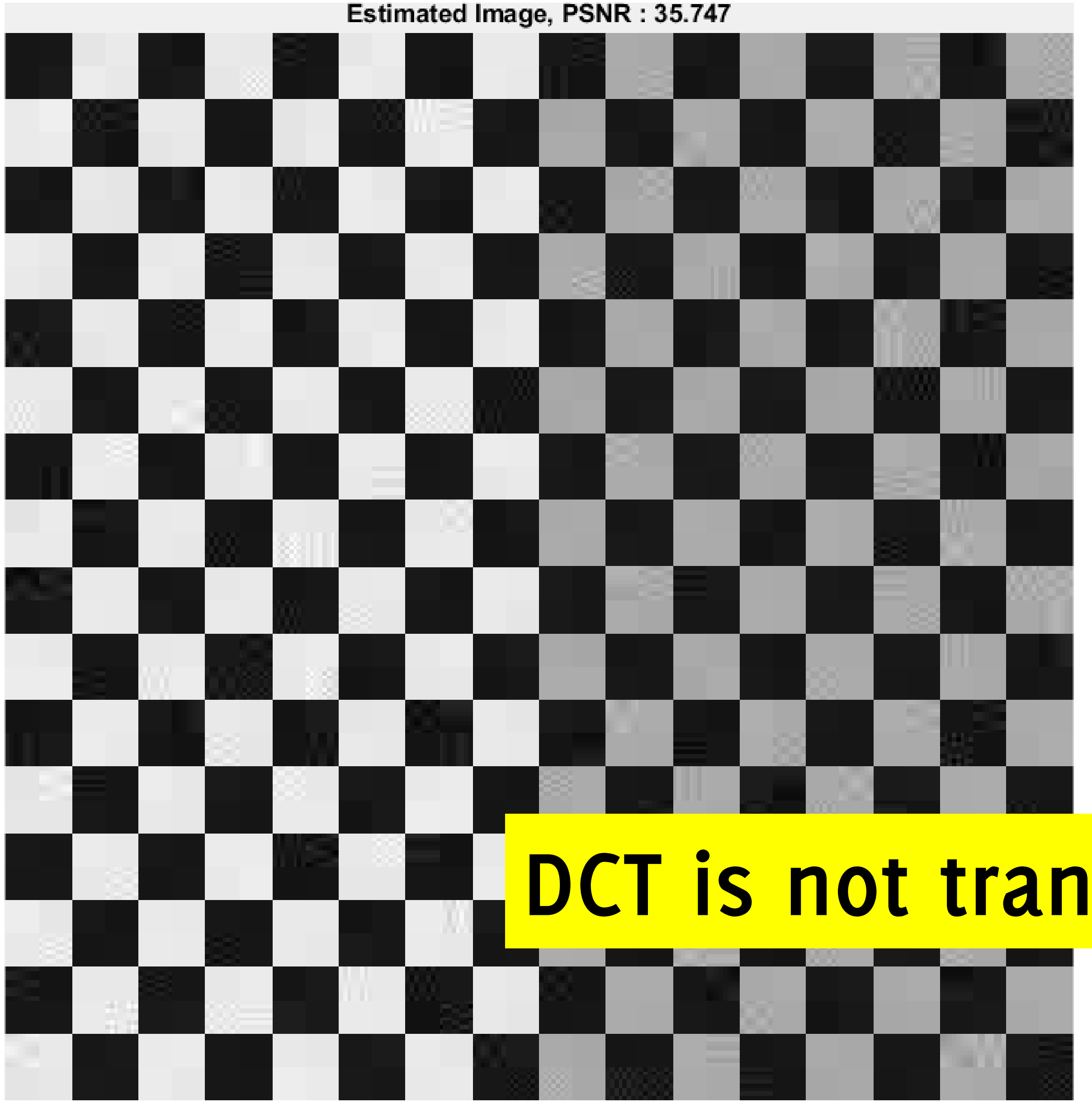
Estimated Image, PSNR : 26.086

# Assignment

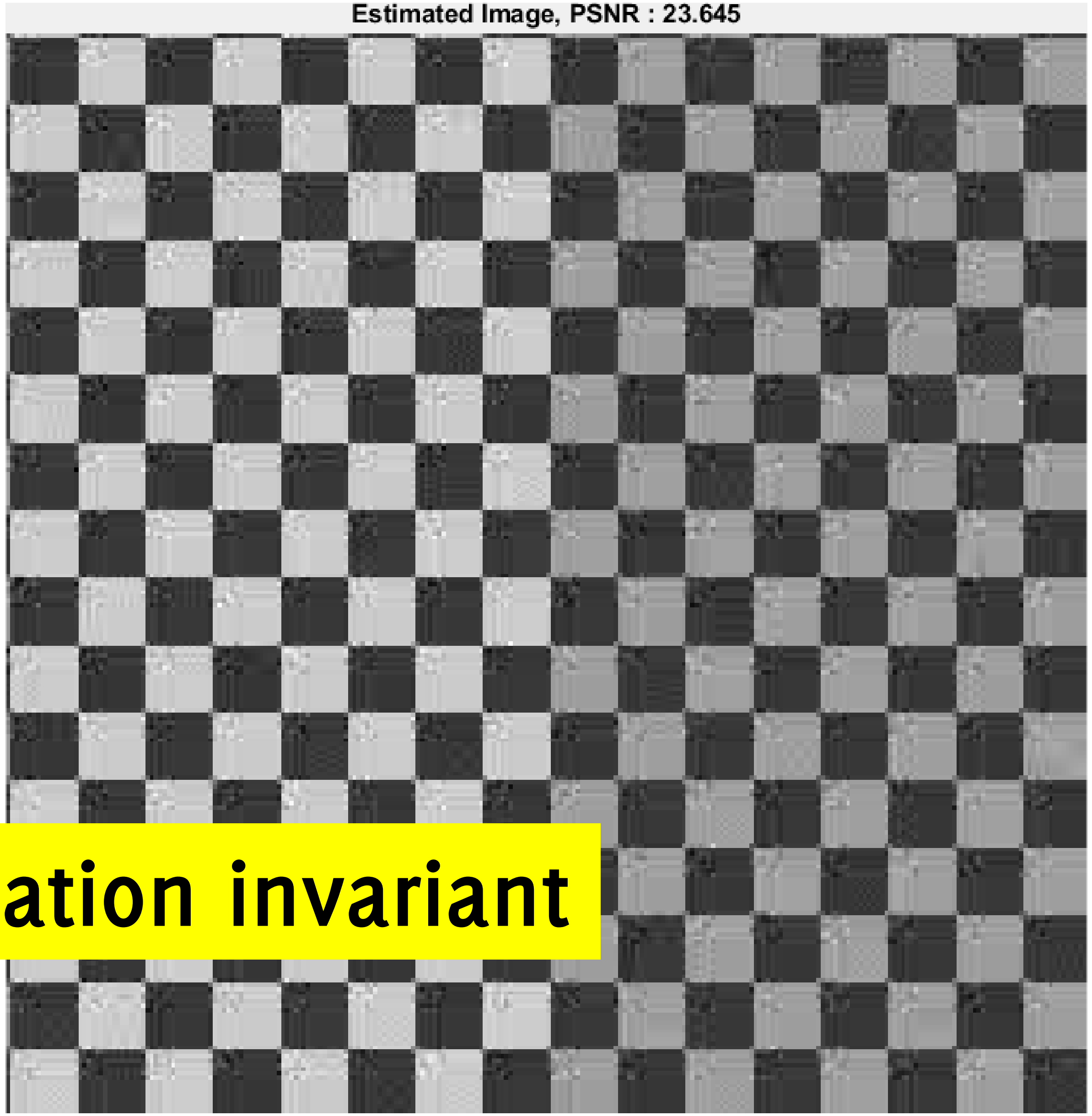
Try the following

- Adopt no aggregation (take non-overlapping patches)
- Denoise a $16 \times 16$ checkerboard image
- Measure the PSNR
- Repeat the operation after shifting 1 right and 1 pixel down the checkerboard

## No shift



## Shift [1 row, 1 col]



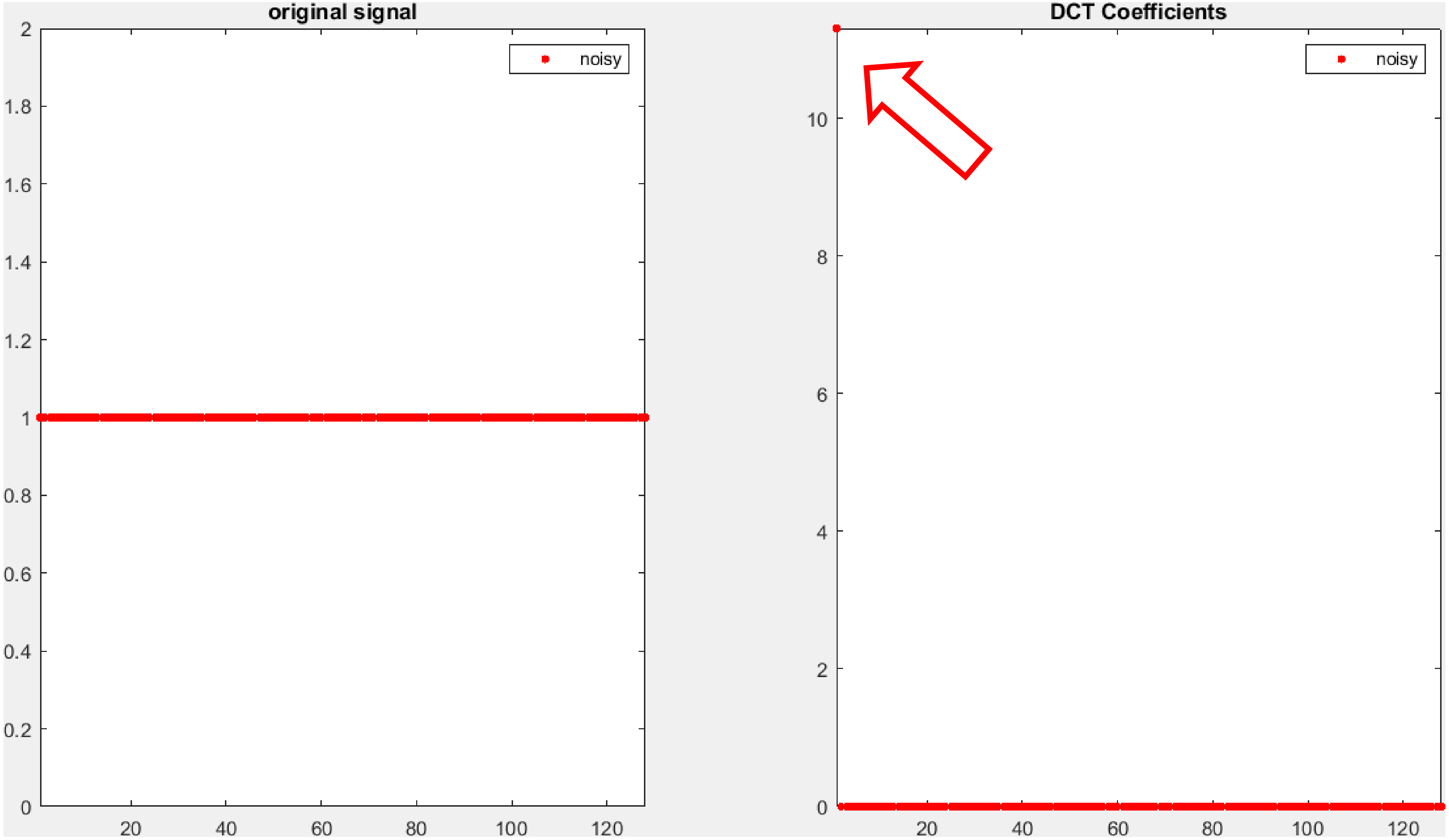
**DCT is not translation invariant**

G. Boracchi

# Let's investigate this further…

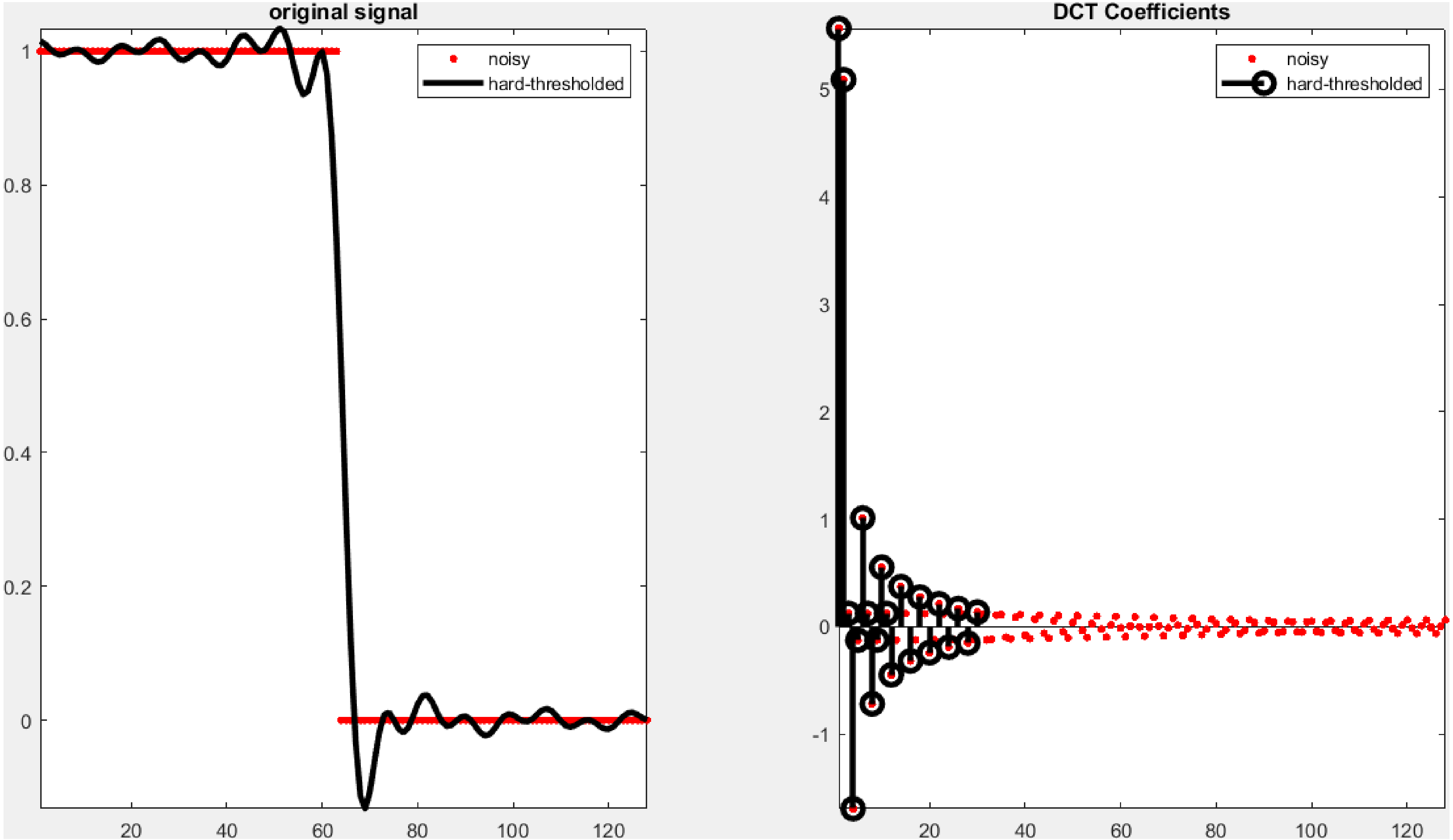
You might want to go back to the 1D signal and check what happens when transforming in DCT domain a constant singal or a shifted version of it (thus including two different levels)

# A Very Sparse Signal



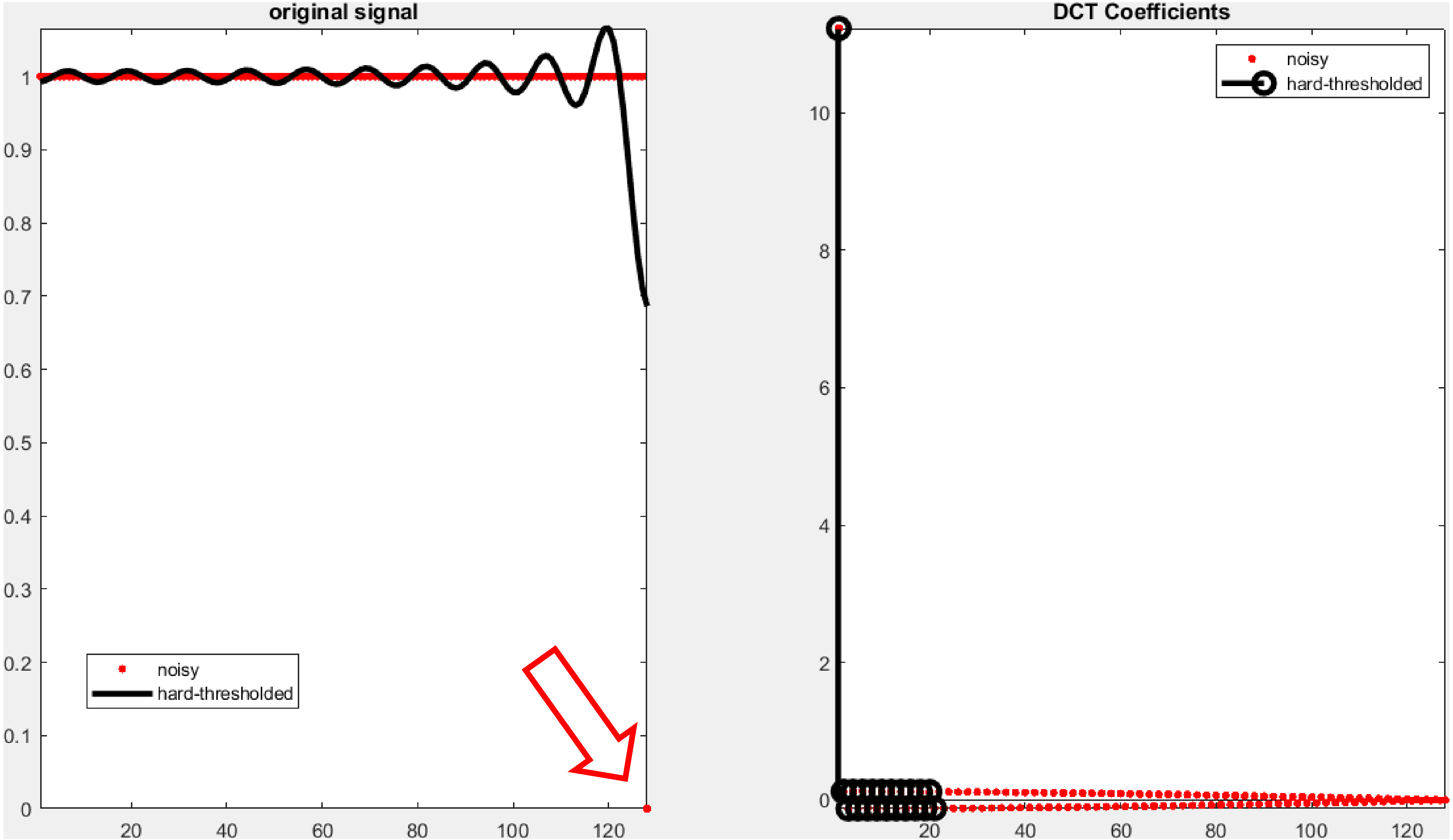
G. Boracchi

# A Shift breaks sparsity!



G. Boracchi

# A Shift breaks sparsity!



G. Boracchi

# Assignment: Move to Sliding DCT

Provide an estimate for each block centered in a pixel.

-> each pixel receives and aggregates $p^2$ estimates

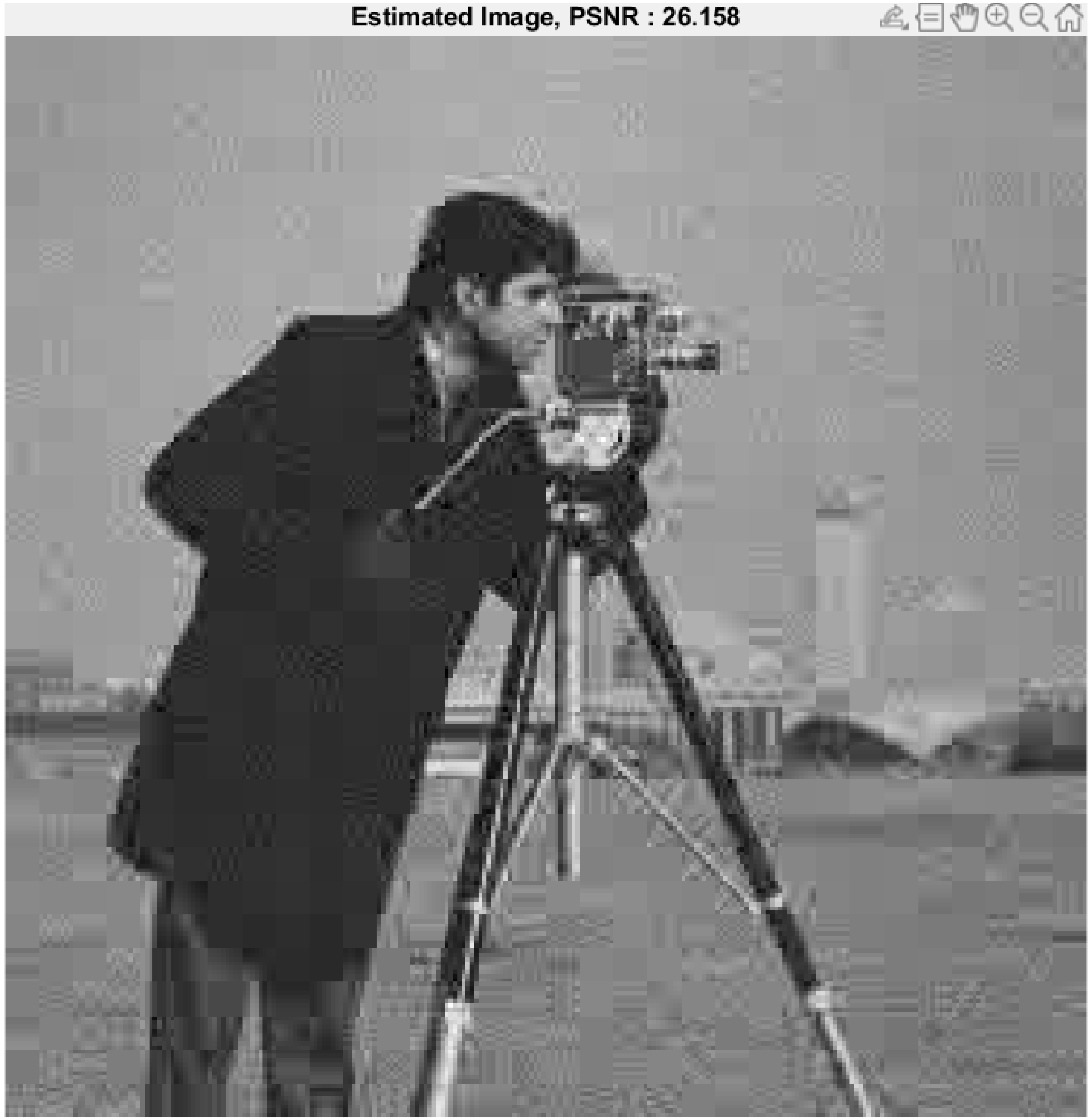
Adopt simple averaging for aggregation

# Aggregation

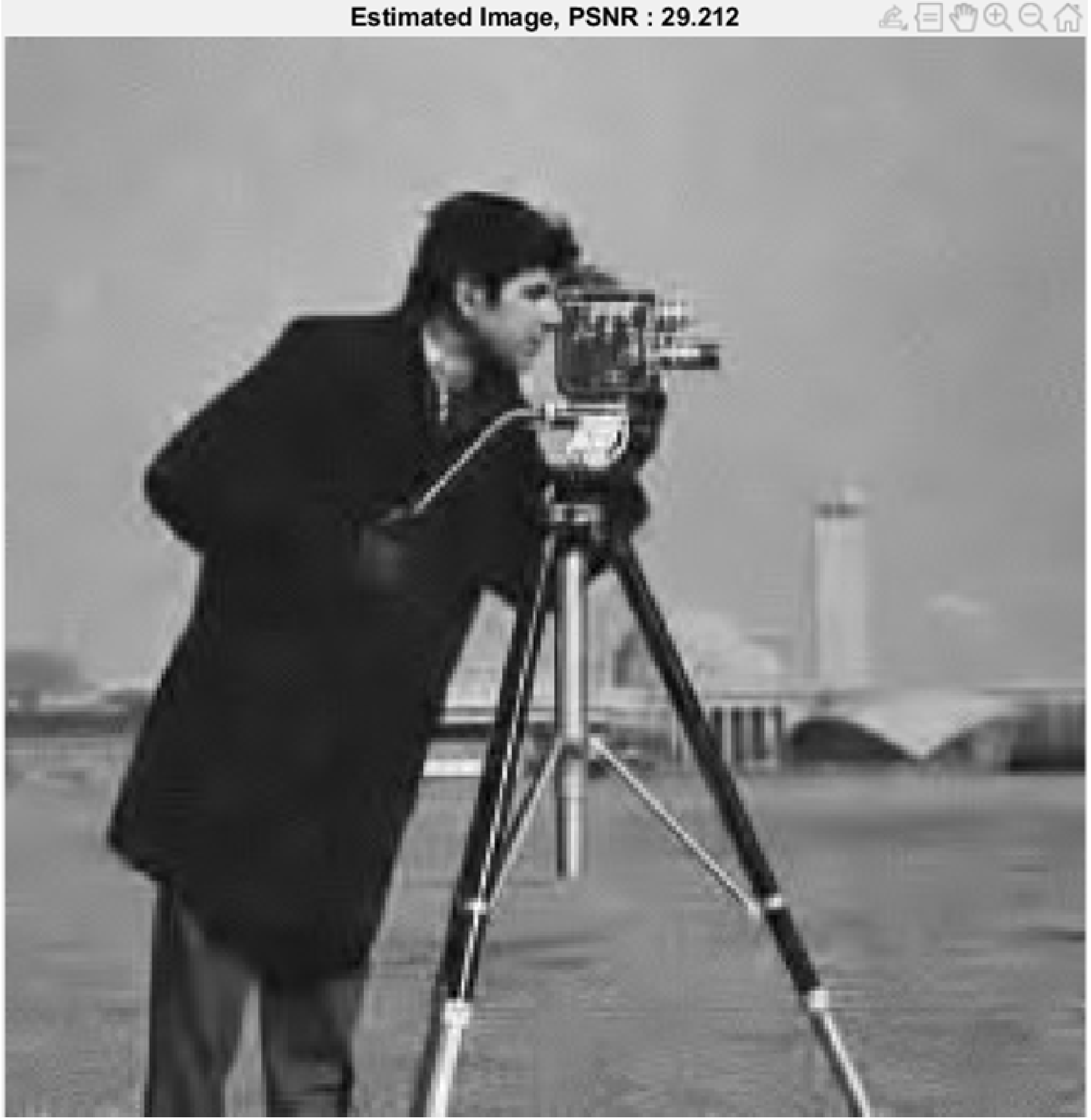
Aggregation considers all the possible shifts, thus **make the DCT translation invariant**

However, not all the shifted versions of the input are good at the same!

# The Benefit of Aggregation



Estimated Image, PSNR : 26.158



Estimated Image, PSNR : 29.212

G. Boracchi

# Aggregation helps

Use Sliding DCT with 2 different types of aggregation weights

- Uniform

$$w(x) = 1$$

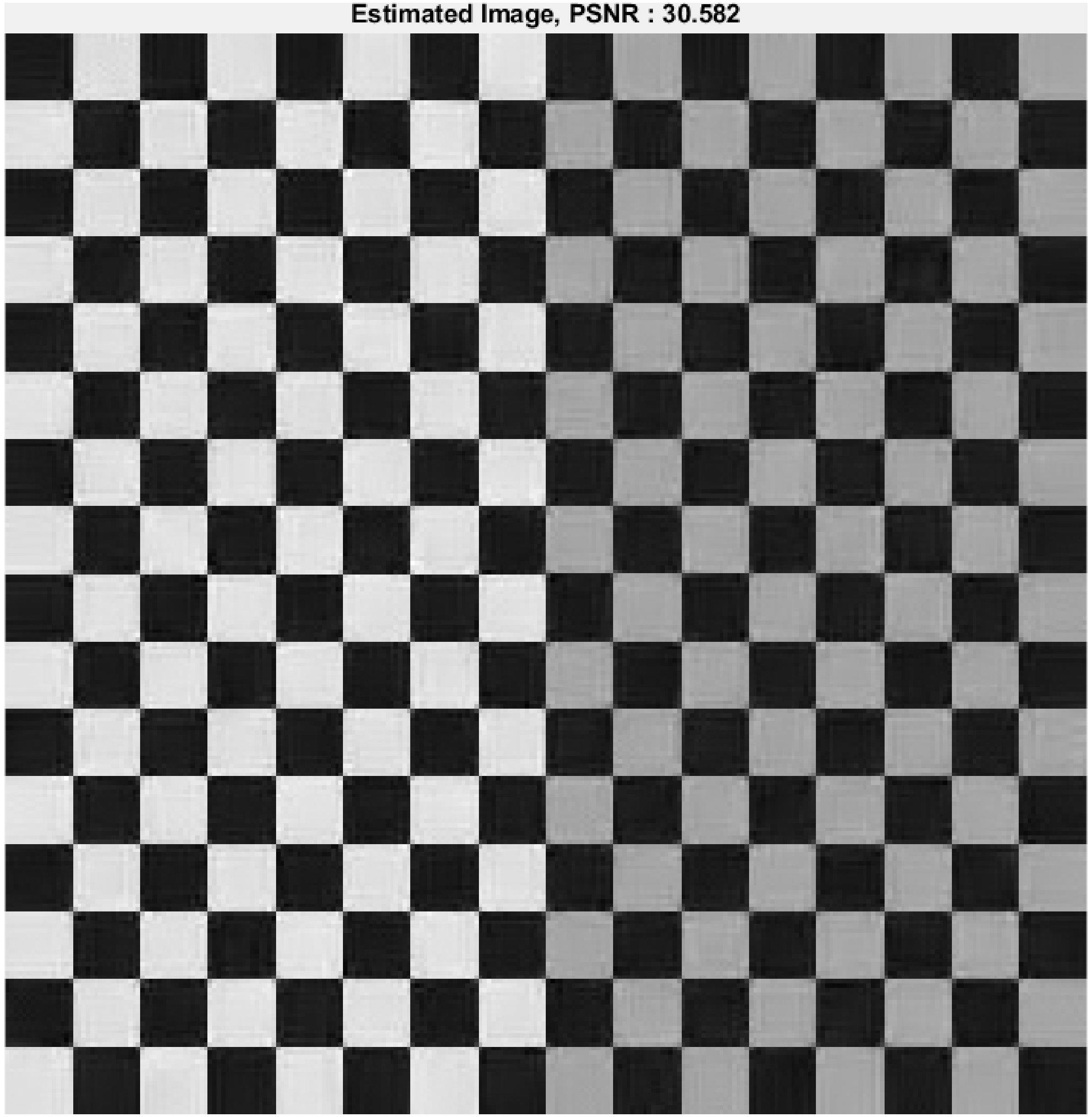
- Sparsity-aware

$$w(x) = \frac{1}{||\hat{x}||_0}$$

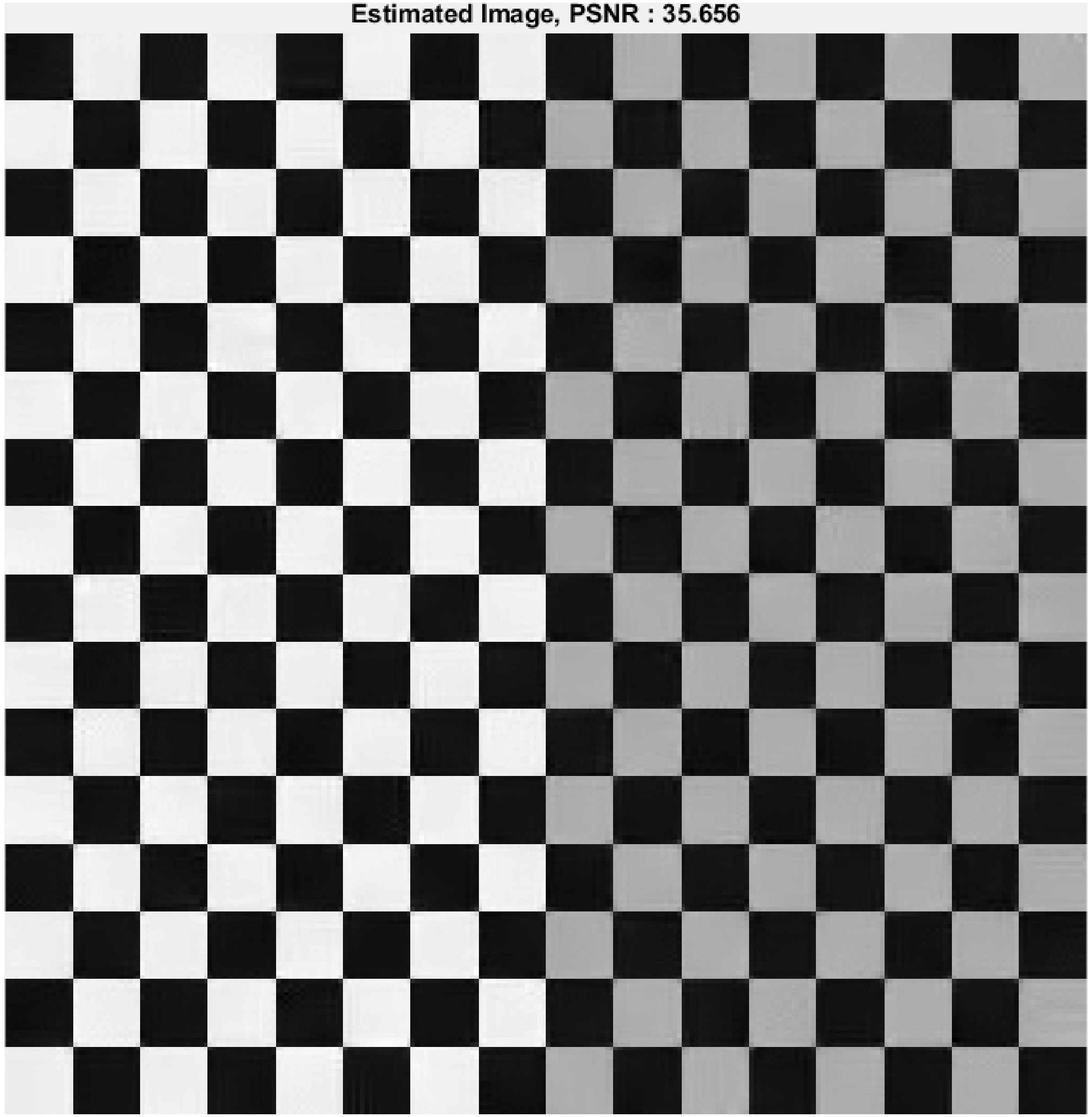
Make sure that when the DC coefficient is zero, $||\hat{x}||_0$ is set to 1

Sparsity-aware weights are larger to those blocks that are sparser. As these achieve superior performance

# Uniform Weights



# Sparsity-aware



G. Boracchi

# Assignment

Implement both forms of aggregation and

- Test both on natural images
- Test both on checkerborard

Finally, test how much the choice of the threshold $\tau$ influences the denoising performance. Observe the resulting image when:

- $\tau \ll 3\sigma$
- $\tau \gg 3\sigma$

This is very important to understand how important is the choice of the threshold

# Noise Estimation

# Estimating $\boldsymbol{\sigma}$

The value of $\sigma$ plays a crucial role in Sliding DCT denoising (and in sparsity promoting algorithms in general)

You can notice this when changing the threshold $\tau$

… but how to estimate the noise standard deviation, provided only a noisy image?

# Noise estimation by filtering

Idea: bring all the flat areas of an image «around zero», and then estimate the sample standard deviation.

$$(z \circledast [-1\,,1]) = ((y+\eta) \circledast [-1\,,1]) =$$
$$= y \circledast [-1\,,1] + \eta \circledast [-1\,,1]$$

Now, the first term should be close to zero except at image boundaries. The second term corresponds to a random variable having distribution

$$\eta \circledast [-1\,,1] \sim \mathcal{N}(0, 2\sigma^2)$$

Therefore

$$\hat{\sigma} = \frac{\text{std}\{z \circledast [-1\,,1]\}}{\sqrt{2}}$$

# Noise Estimation by Filtering + Robust Statistics

Using the sample variance std{} might be heavily affected by outliers, which can result from the term $y \circledast [-1, 1]$

$$\hat{\sigma} = \frac{\mathrm{std}\{z \circledast [-1, 1]\}}{\sqrt{2}}$$

A better estimate is provided by a robust estimator of the sample variance, namely the Median of Absolute Deviation

$$\hat{\sigma} = \frac{\mathrm{MAD}\{z \circledast [-1, 1]\}}{0.67449 \ * \sqrt{2}}$$

Being $\mathrm{MAD(X)} = \mathrm{median}\{|X - \mathrm{median}\{X\}|\}$

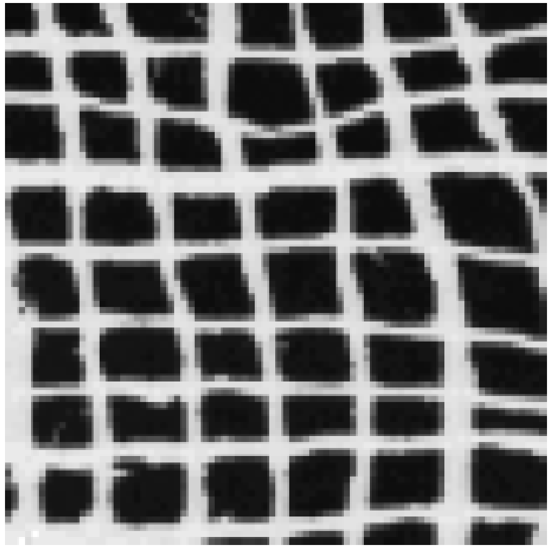
# Assignment

Implement the noise estimation formula and use this in the denoising framework

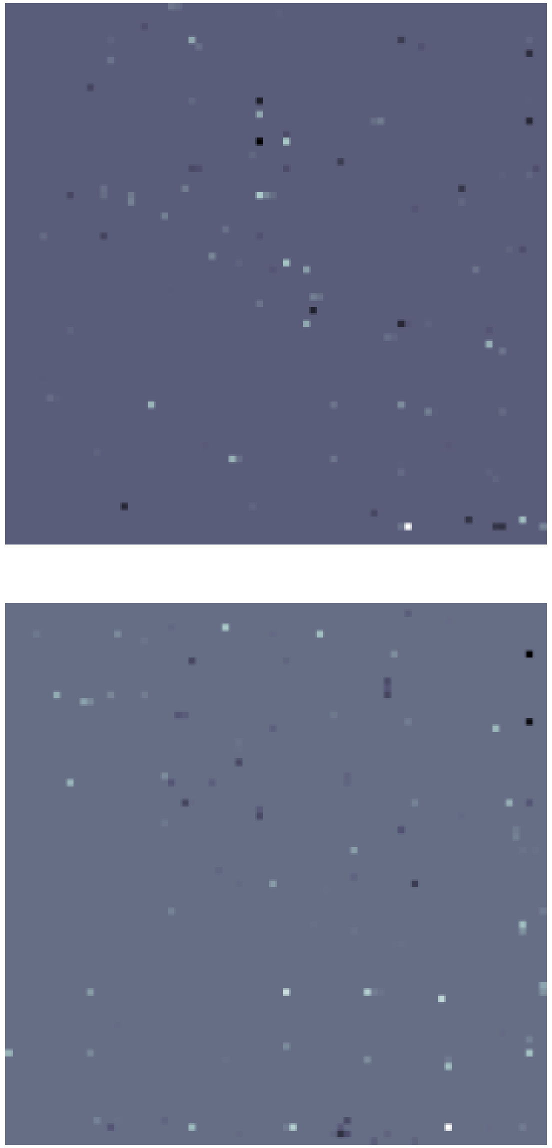
# Convolutional Sparse Coding

Gobal Optimization vs Aggregation of Partial Estimates

## Test Image

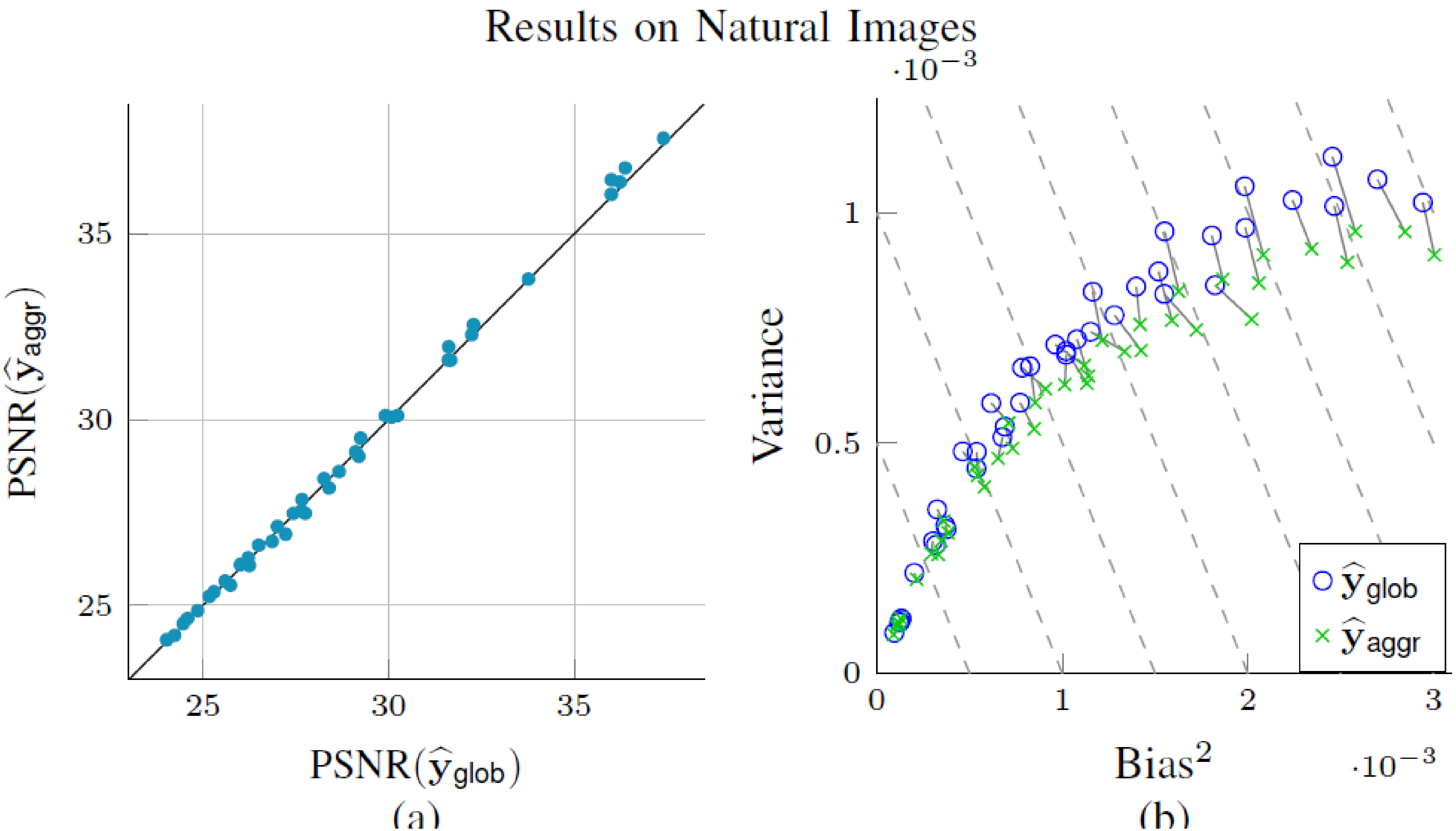


## Feature maps



## Filters

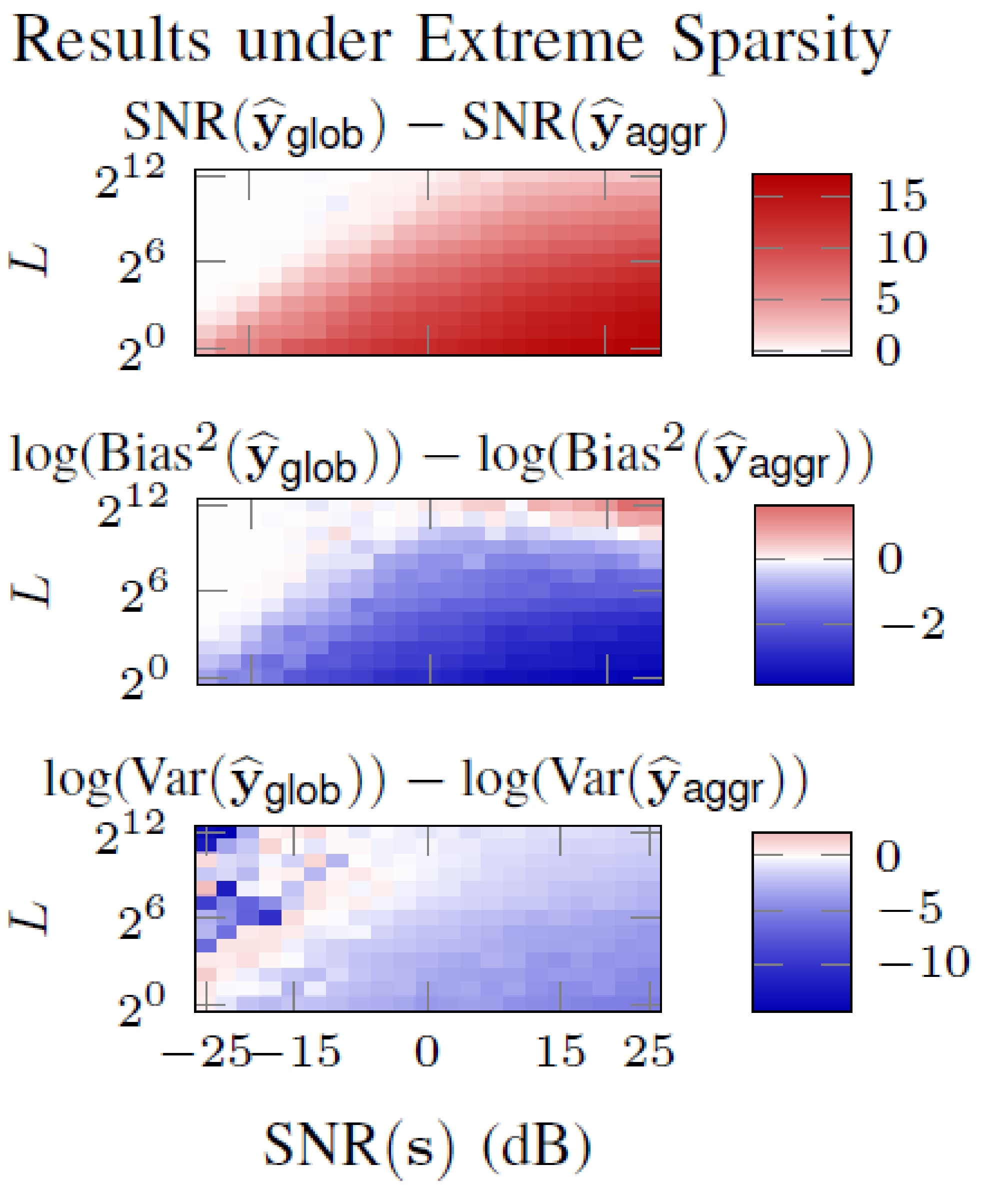
# Global Optimization vs Partial Aggregation, ell1 regularization, natural images



**[SPARS 2017]** D. Carrera, G. Boracchi, A. Foi and B. Wohlberg , *“Sparse denoising: aggregation versus global optimization“* SPARS 2017

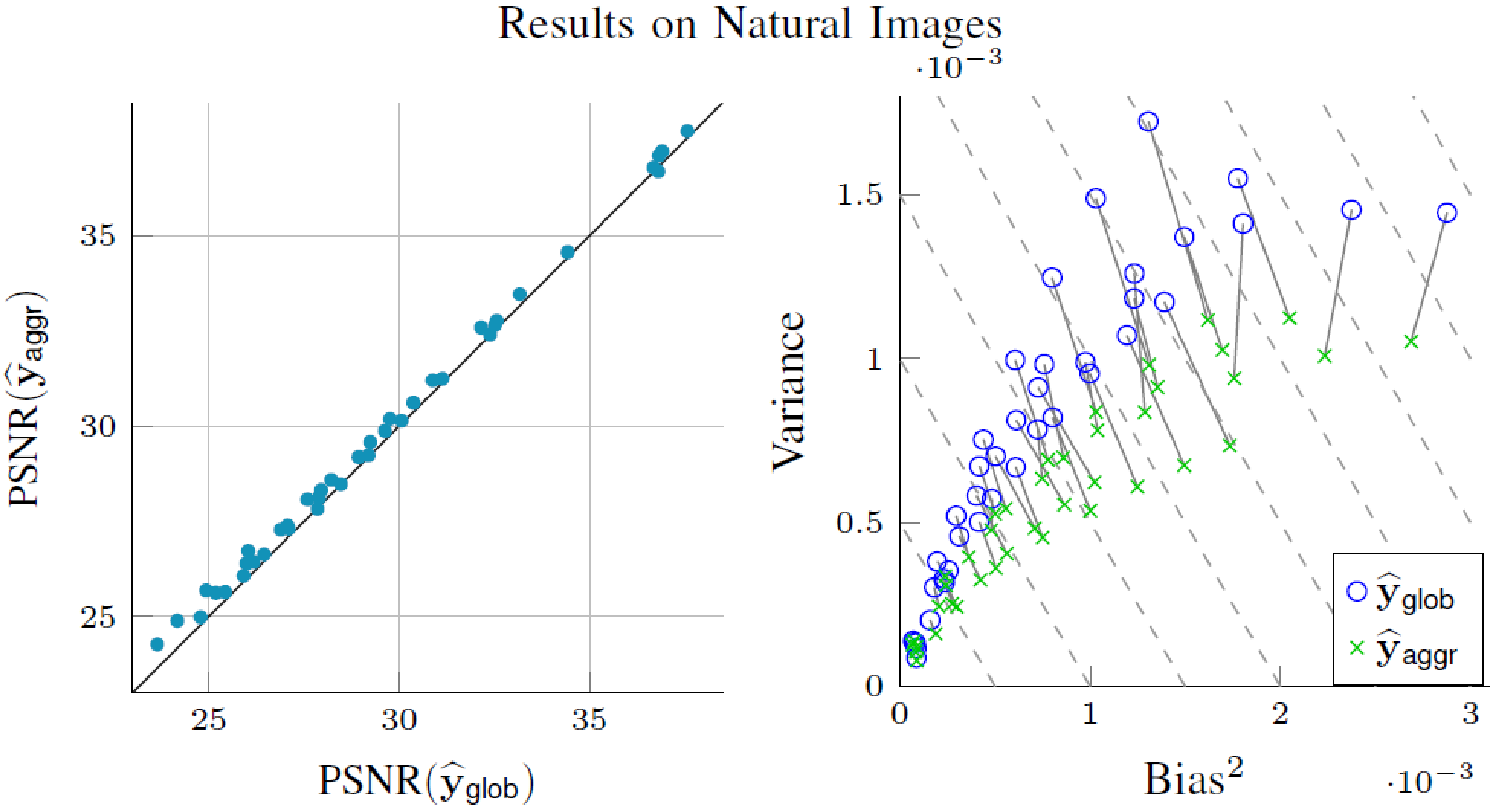
G. Boracchi

# Global Optimization vs Partial Aggregation, ell1 regularization, synthetic and very sparse images



G. Boracchi

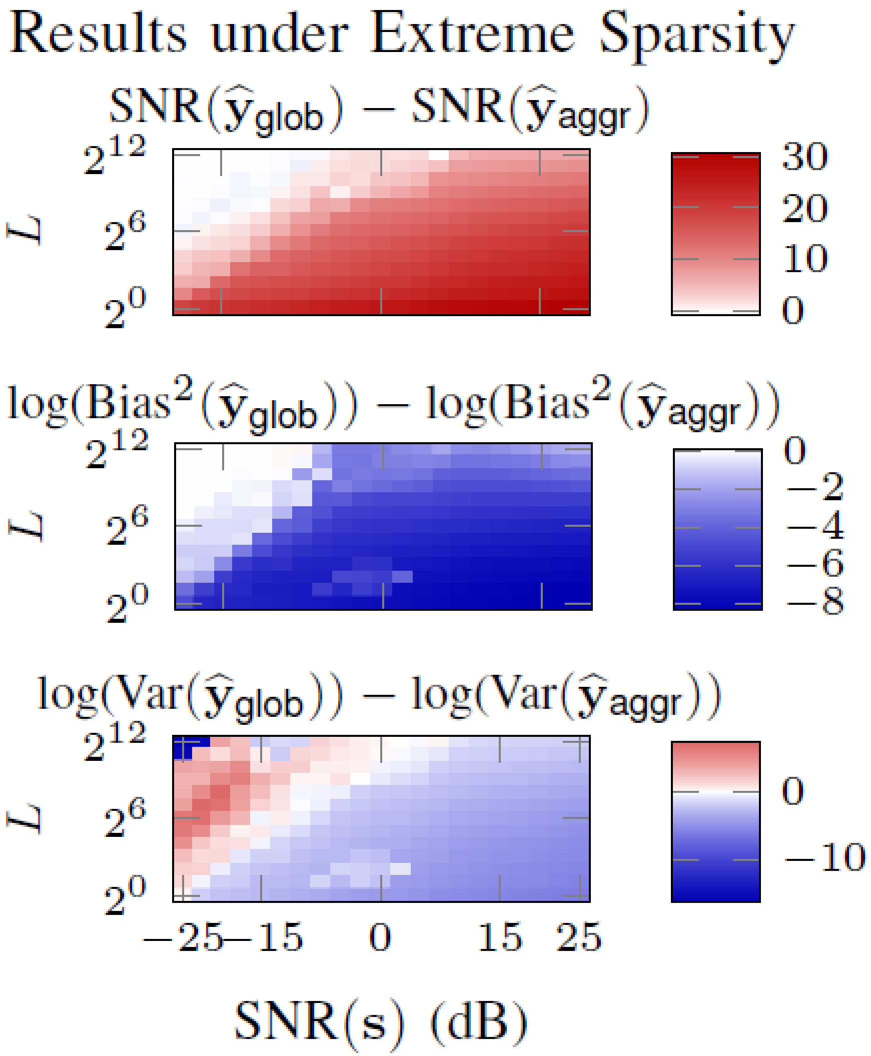
# Global Optimization vs Partial Aggregation, ell o regularization, natural images


Results on Natural Images

(a)

(b)

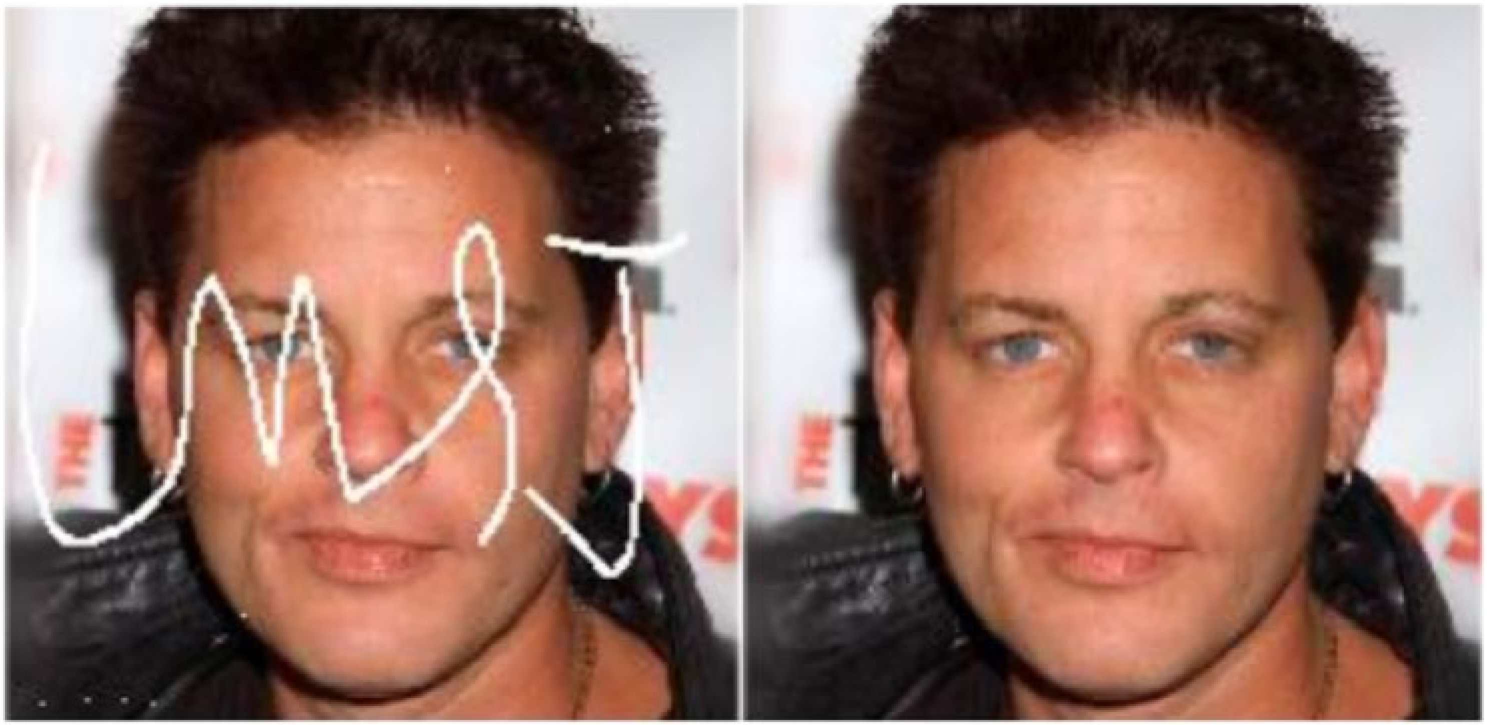
# Global Optimization vs Partial Aggregation, ello regularization, synthetic and very sparse images



G. Boracchi

# Image Inpainting

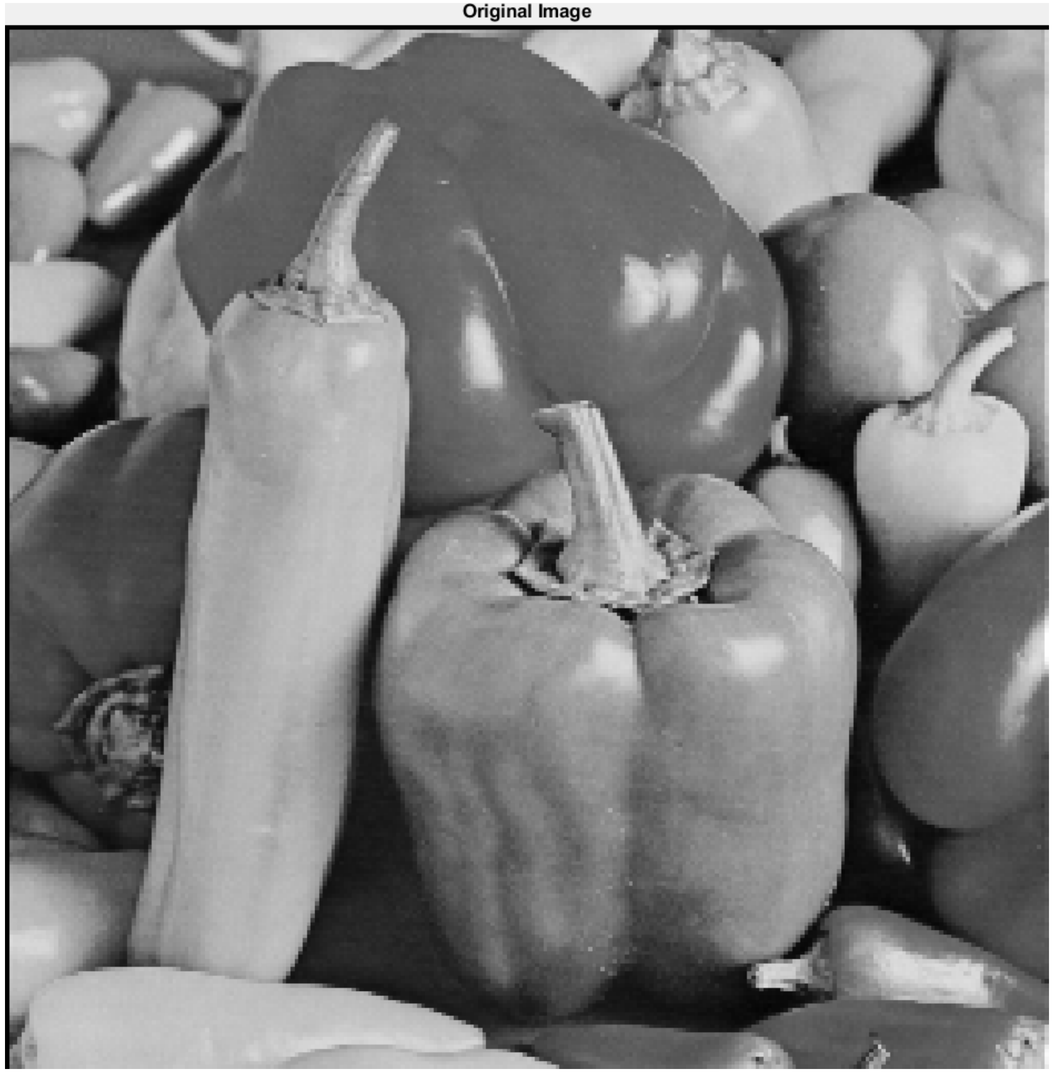
# Image Inpainting



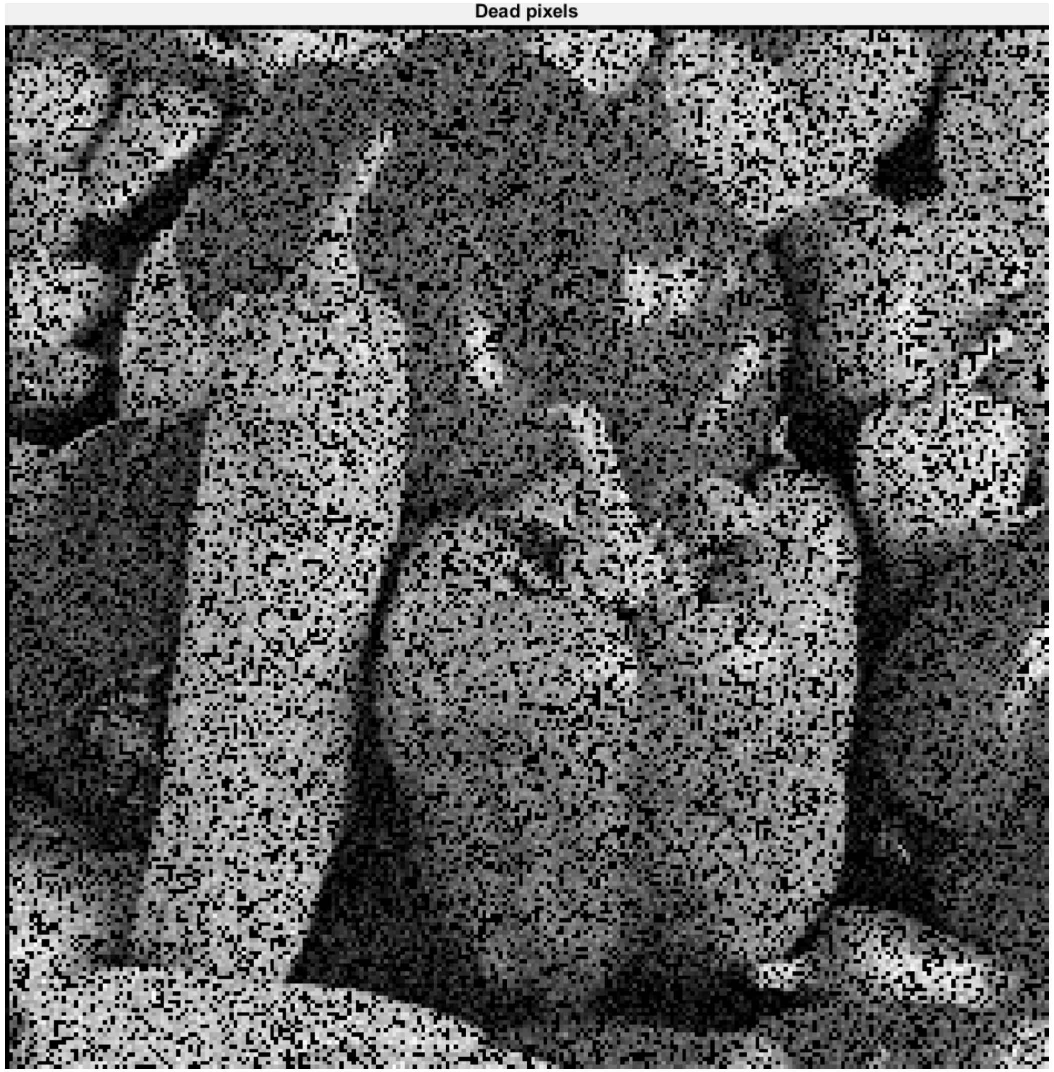
(a) **Masked-Image** (b) **Inpainted-Image**

Jam, Jireh, et al. "A comprehensive review of past and present image inpainting methods." *Computer vision and image understanding* 203 (2021): 103147.

# Image Formation Model

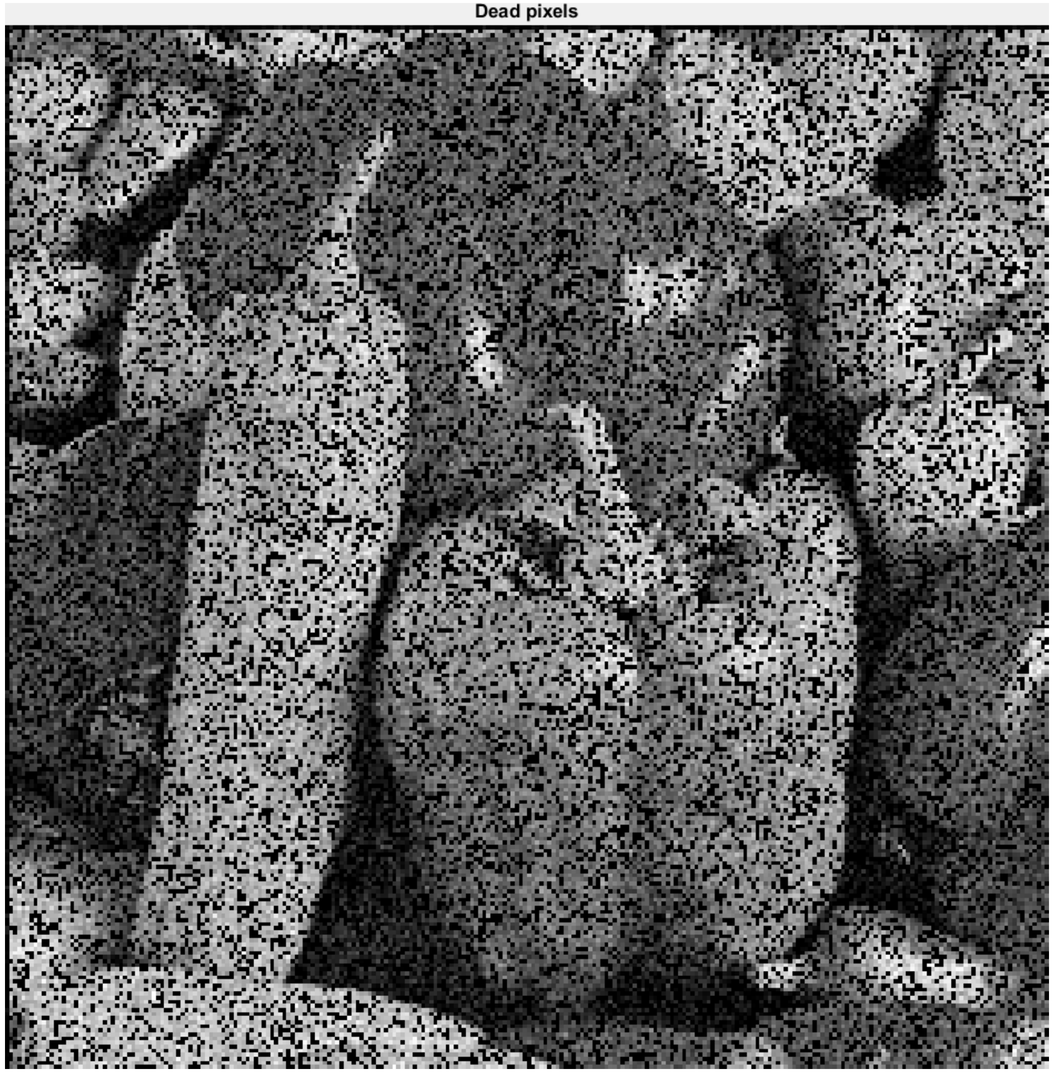


Original Image

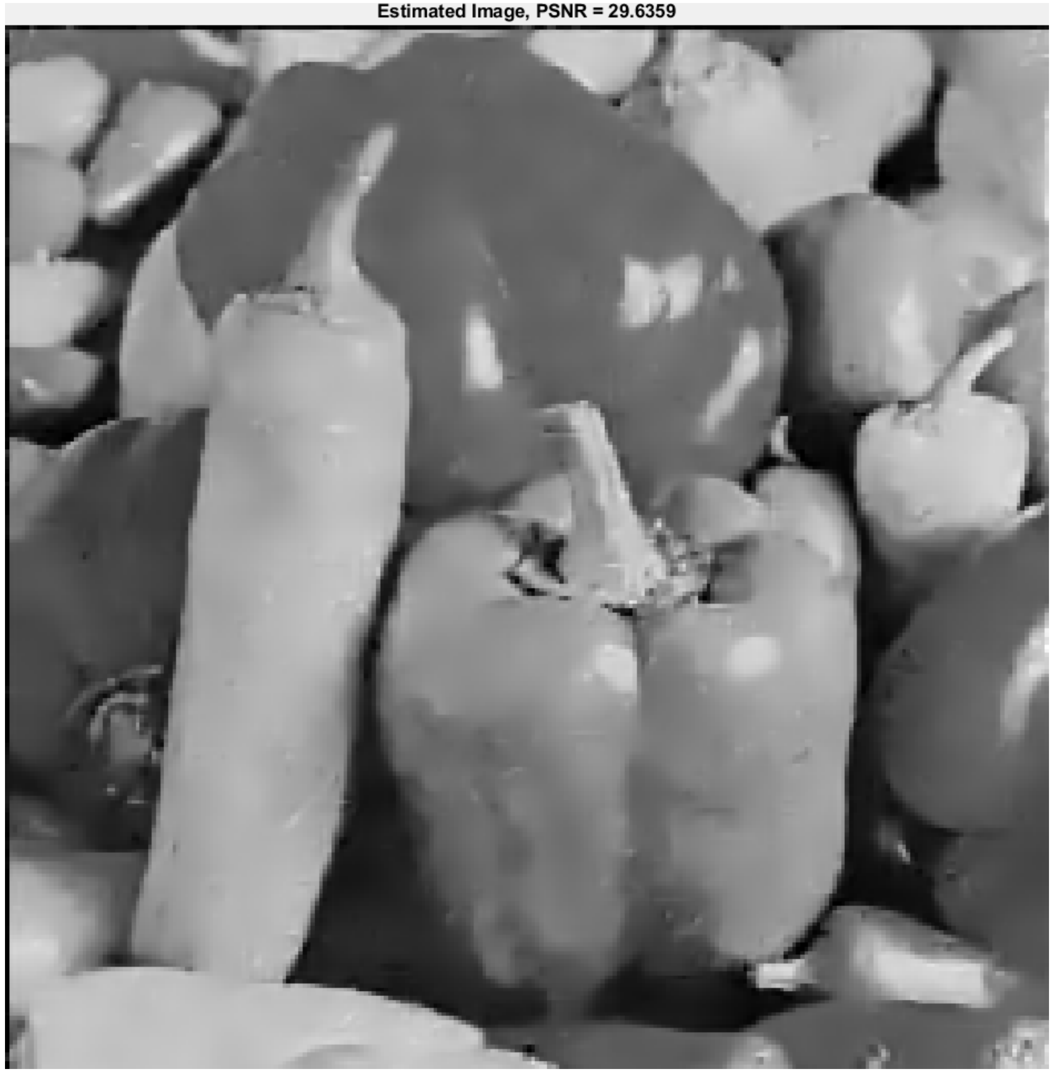


Dead pixels

# Image Inpainting



Dead pixels



Estimated Image, PSNR = 29.6359

# Assignment

## Image Inpainting Enforcing Sparsity

# Denoising via Sparse Coding

Take the setup of Assignment 3 (denoising via DCT)

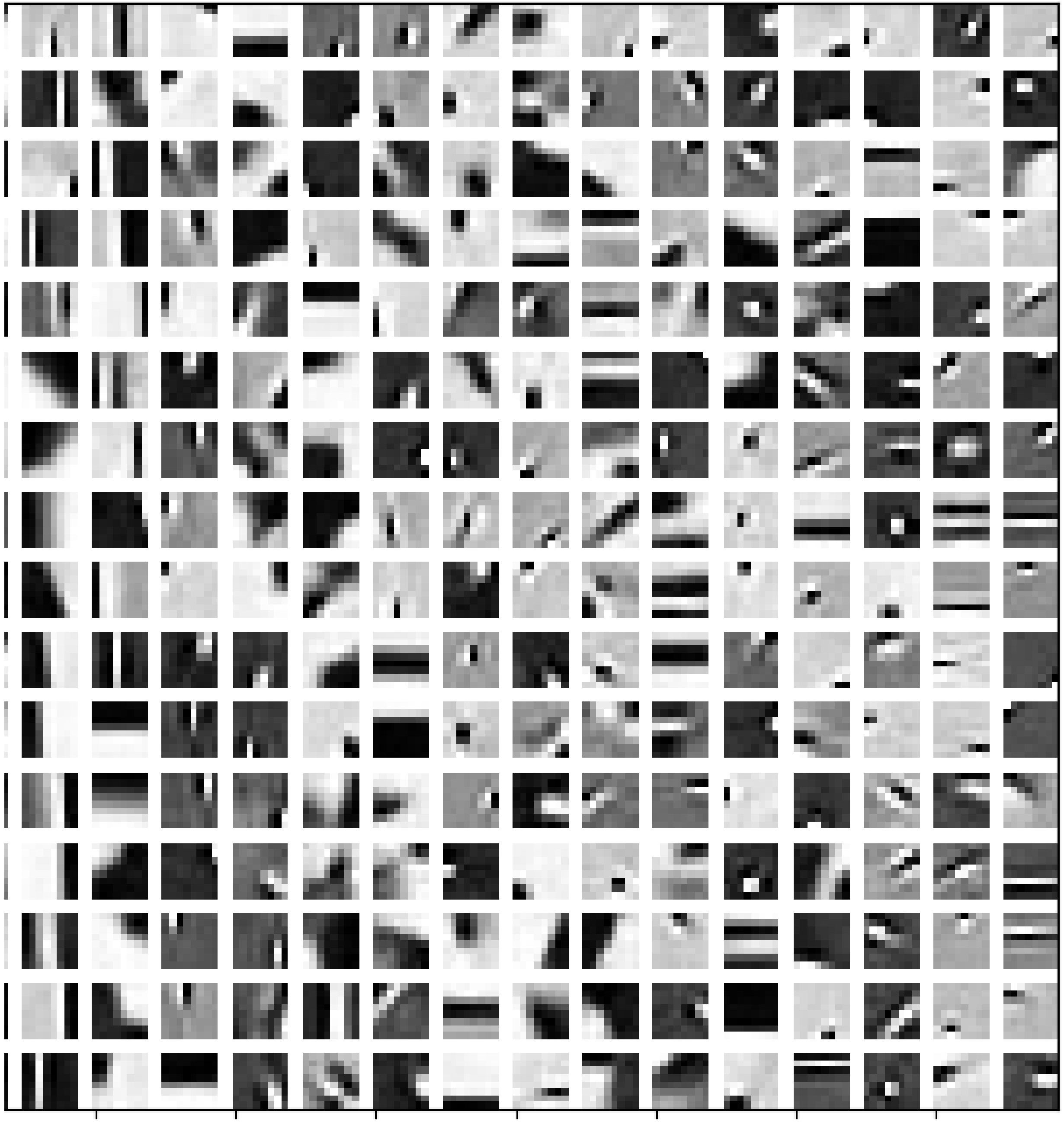
- Load the dictionary provided (learned from natural images)
  - Add a constant atom and avoid average subtraction
- Replace the analisys and the thresholding of patch $s_i$ with the sparse coding using the OMP with respect to the inpainted dictionary $P_i D$. Use as a threshold for residual

$$\delta_i = 1.15 \cdot p \cdot \sigma \cdot \sqrt{\frac{p^2 - m}{p^2}}$$

  being $m$ the number of zero entries in $s_i$
- Perform the synthesys of each patch using the original dictionary $D$

# The Dictionary from KSVD

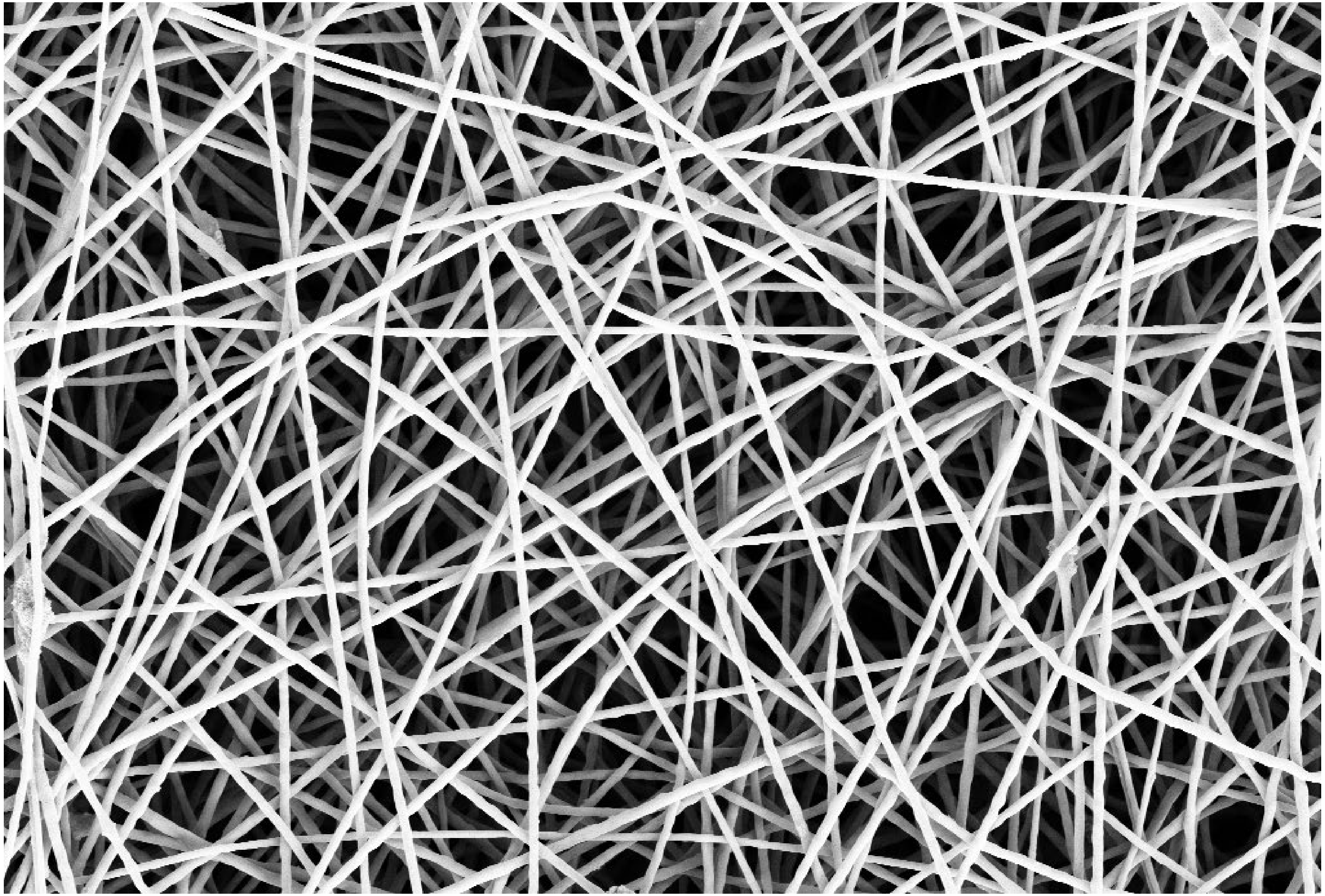
+ remember to add a constant atom!



G. Boracchi

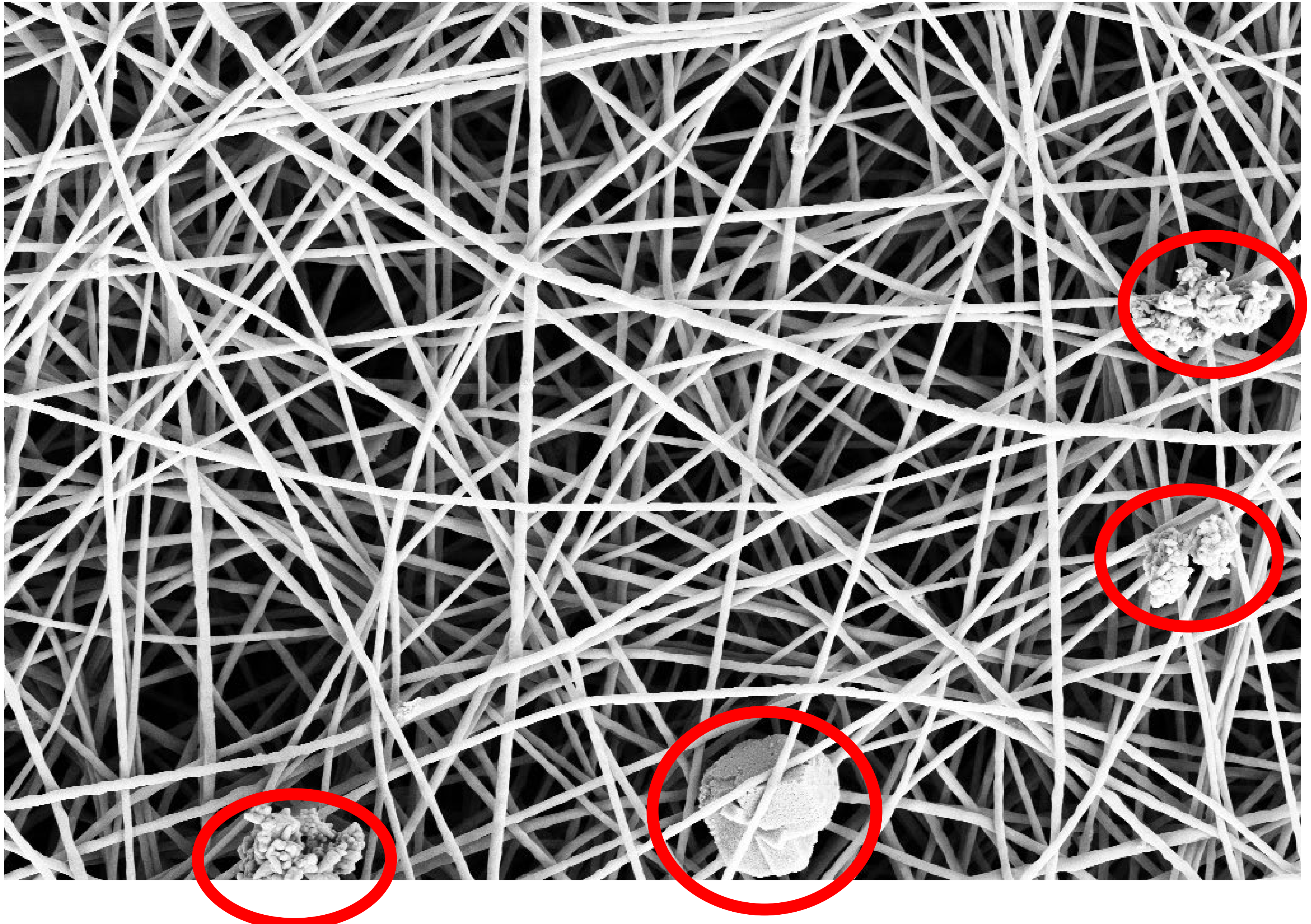
# Anomaly Detection

# The anomaly detection problem

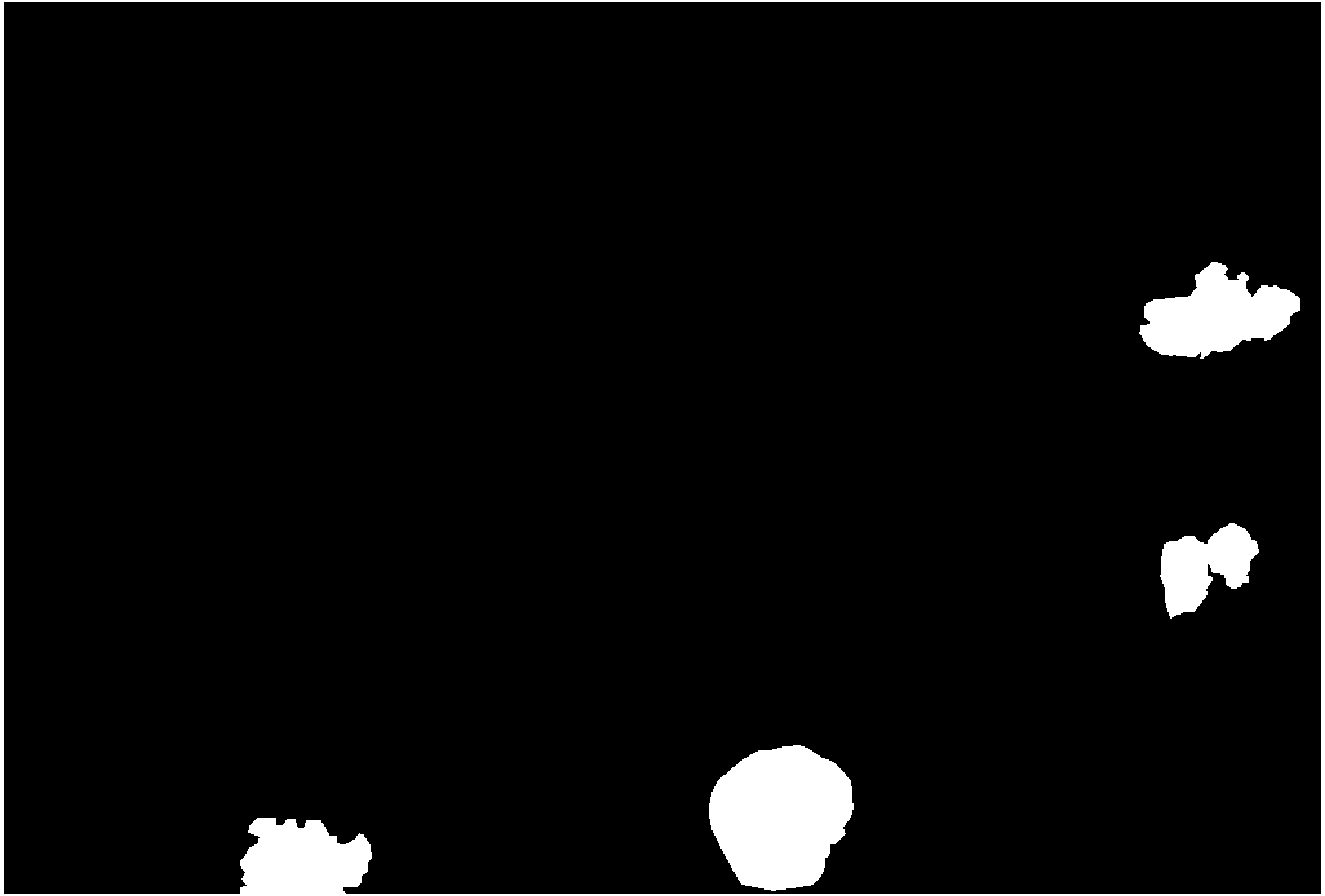


G. Boracchi

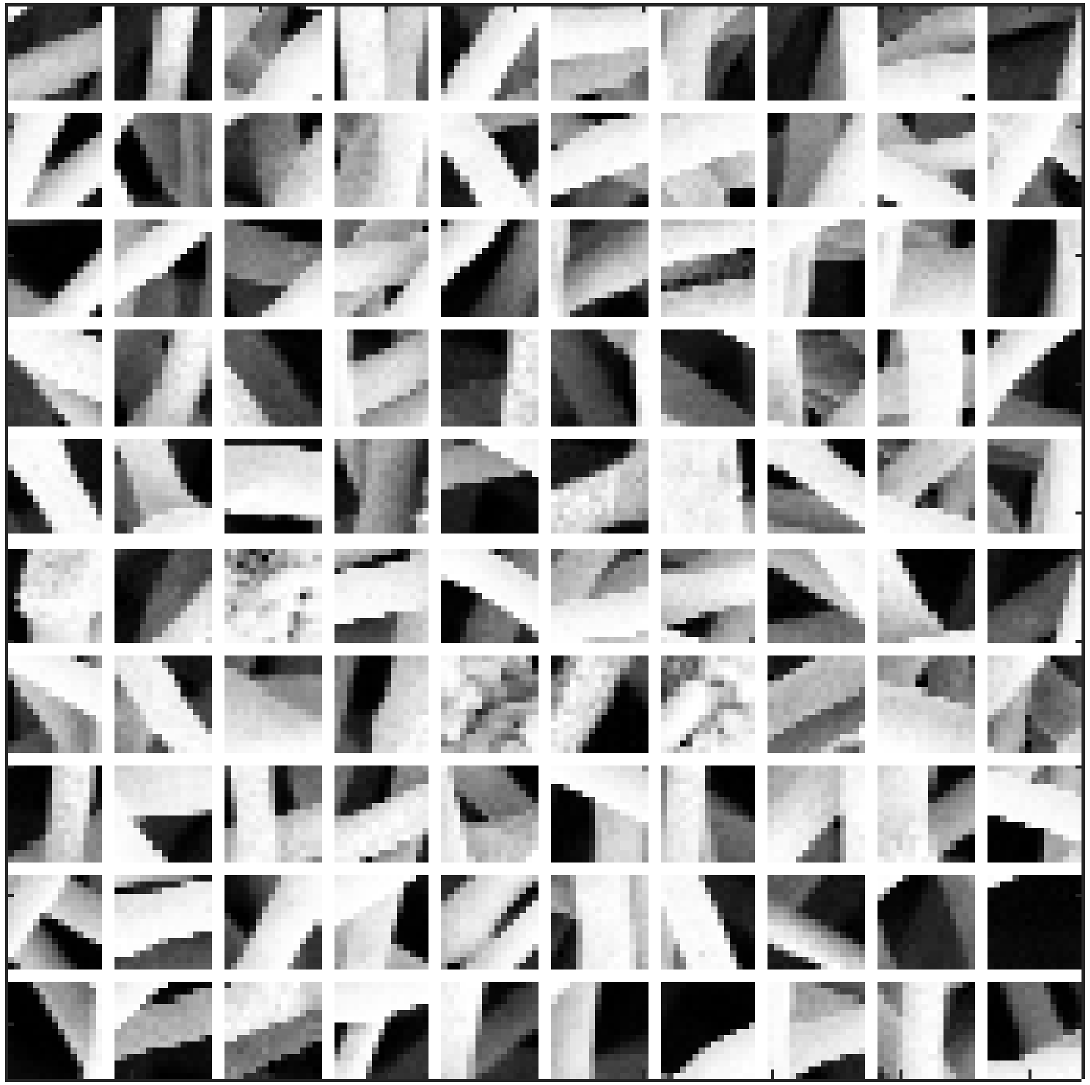
# The anomaly detection problem



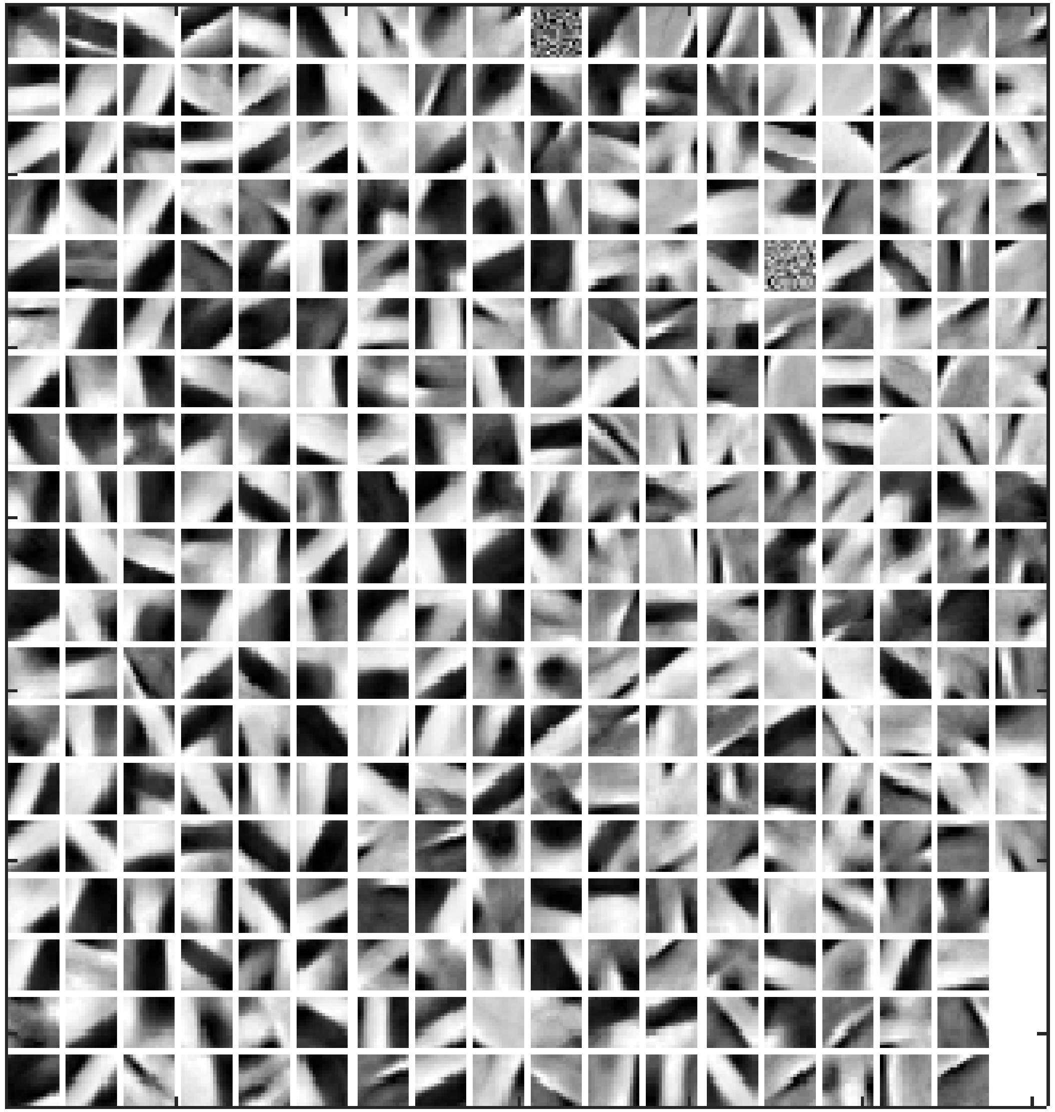
# The anomaly mask



## Normal Patches

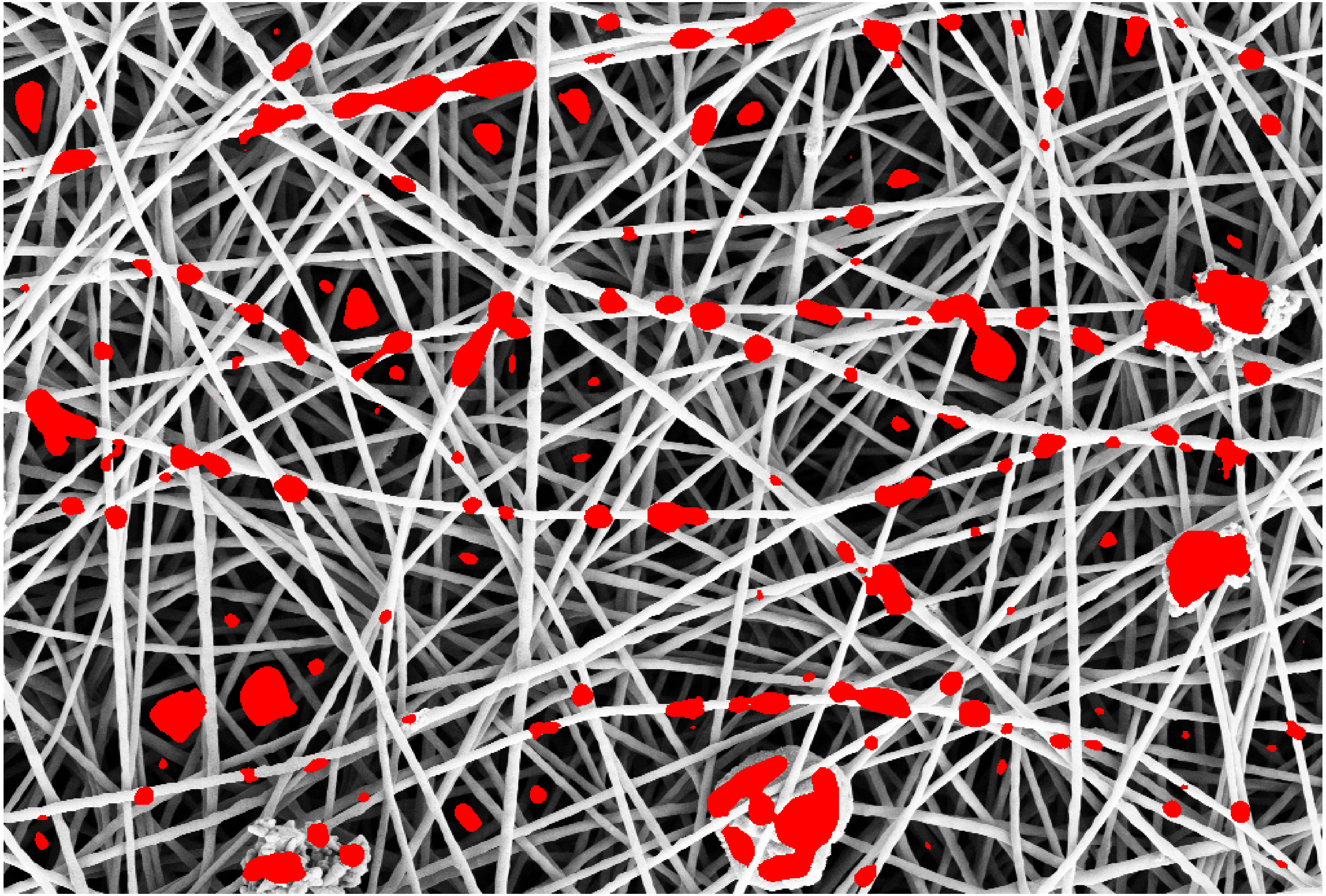


## Learned Dictionary



G. Boracchi

# Detections



G. Boracchi

# The Typical approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

**Remark**: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

# The Typical approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score**, or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and ave sistent with neighbourhoods

The background model is used to **bring an image patch into the “random variable world”**

**Remark**: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

# The Typical approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and tent with neighbourhoods

**Remark**: Statistical- background model the statistic maly score

Once "having applied" the background model, one can use **anomaly detection methods for the "random variable world"**. This might require **fitting an additional model**

# The Typical approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

**Remark**: Statistical-based approaches seen before uses as background model the statistica naly score

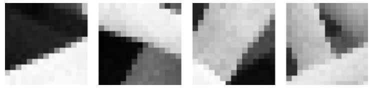
**And it is important to control the False Positive Rate**

# The three major ingredients

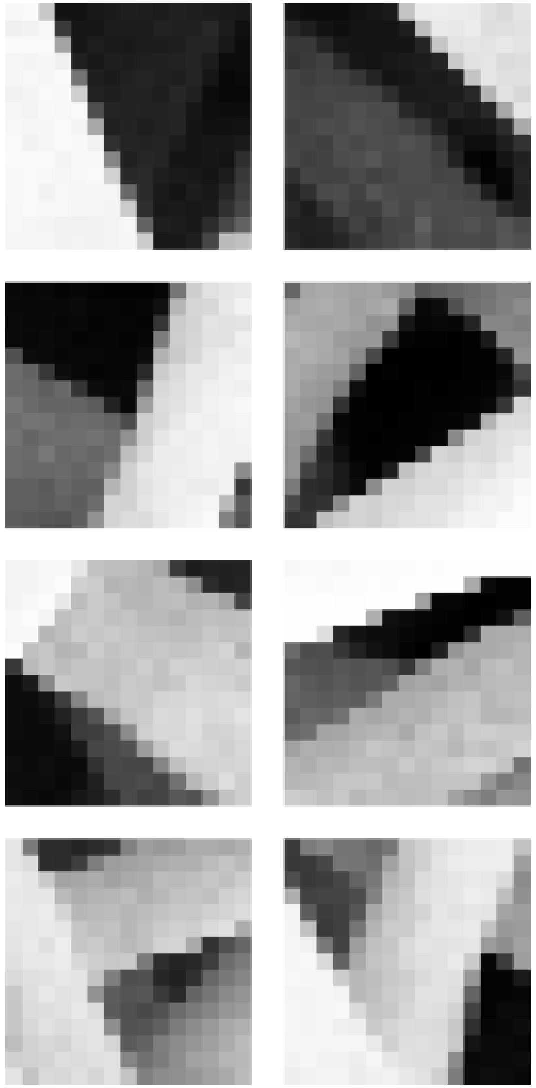
Most detection algorithms have three major ingredients:

- The **background model** $\mathcal{M}$, learned from normal data
- The **statistic / anomaly score**: $\mathrm{err}(\boldsymbol{s}), \mathcal{L}(\boldsymbol{s}), \mathcal{A}(\boldsymbol{s}), \dots$
- **Decision rule** to detect, e.g. $\mathrm{err}(\boldsymbol{s}) \gtreqless \gamma$ possibly controlling the FPR, as in other statistical detection methods

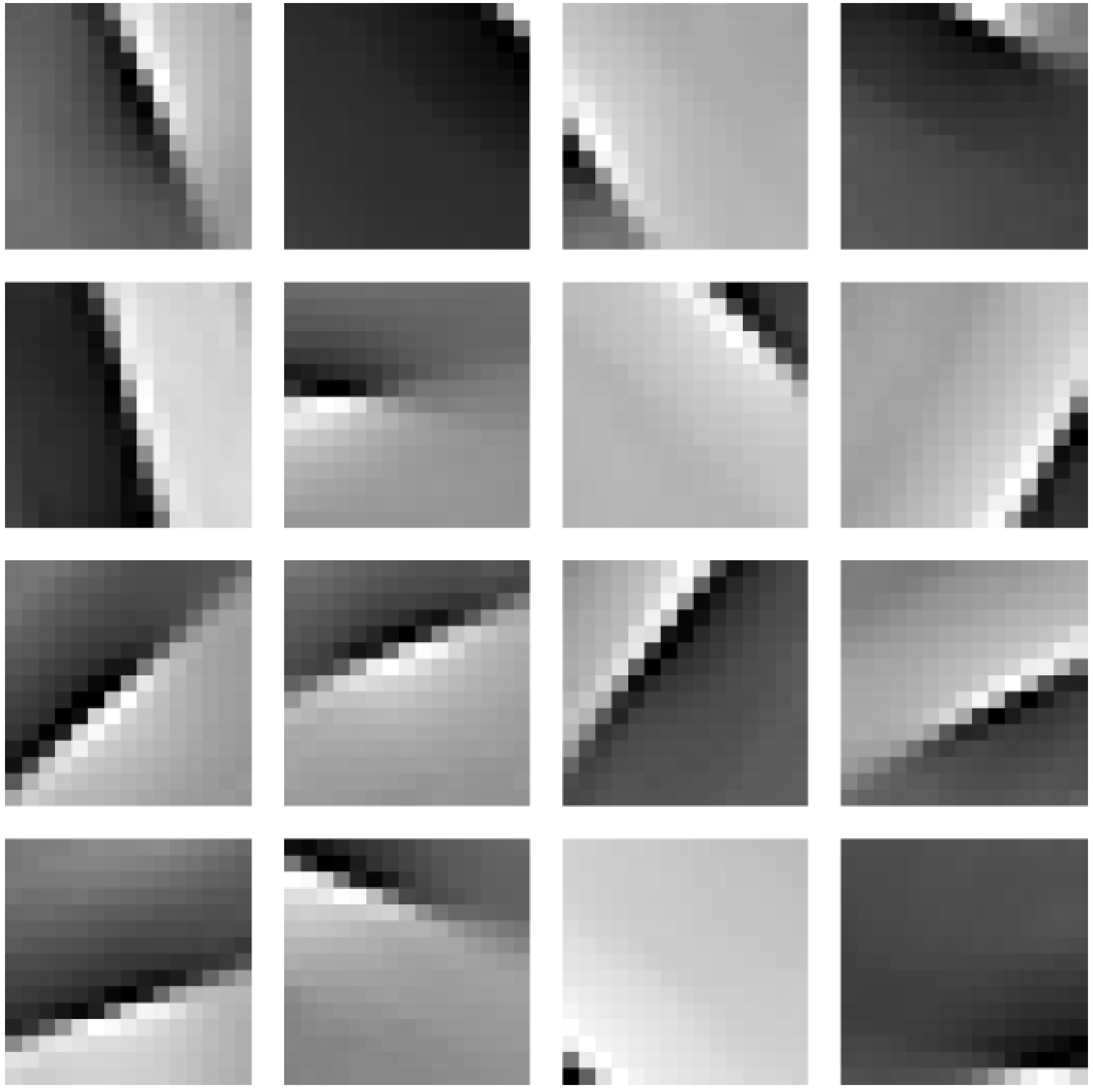
# A Dictionary learned from normal patches



Example of training patches



Few learned atoms (BPDN-based learning)



# SPARSE REPRESENTATIONS AS FEATURE EXTRACTORS

To assess the conformance of $\boldsymbol{s}_c$ with $D$ we solve the following

**Sparse coding:**

$$\boldsymbol{x}_c = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|D\boldsymbol{x} - \boldsymbol{s}_c\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \qquad \lambda > 0$$

which is the BPDN formulation and we solve using ADMM.

The penalized $\ell^1$ formulation has more degrees of freedom in the reconstruction, **the conformance of $\boldsymbol{s}$ with $\boldsymbol{D}$ have to be assessed monitoring both terms of the functional**

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. *"Distributed optimization and statistical learning via the alternating direction method of multipliers"* 2011

# Features extracted from sparse coding

Features then include both the **reconstruction error**

$$\mathrm{err}(\boldsymbol{s}_c) = \ \|D\boldsymbol{x}_c - \boldsymbol{s}_c\|_{\mathbf{2}}^{\mathbf{2}}$$

and **the sparsity** of the representation

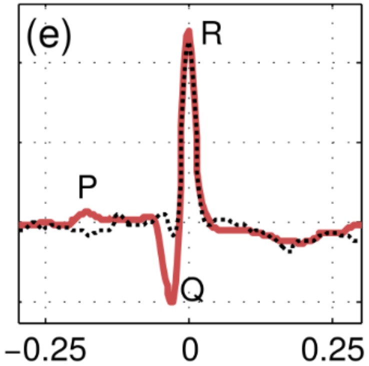
$$\|\boldsymbol{x}_c\|_{\mathbf{1}}$$

Thus obtaining **a data-driven feature vector**

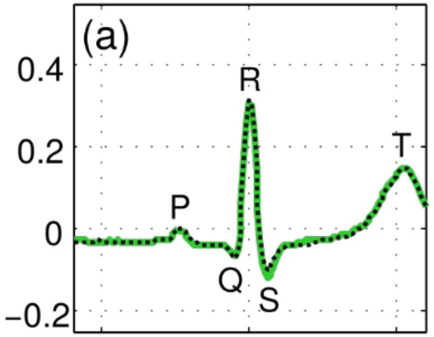
$$\boldsymbol{f_c} = \begin{bmatrix} \|D\boldsymbol{x}_c - \mathbf{s}\|_{\mathbf{2}}^{\mathbf{2}} \\ \|\boldsymbol{x}_c\|_1 \end{bmatrix}$$

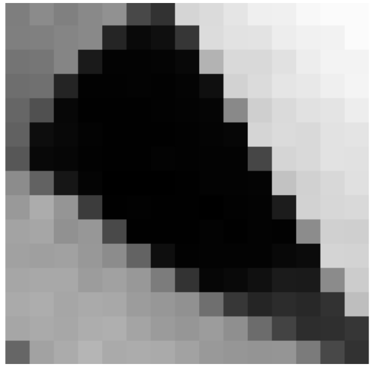
# Density-based monitoring

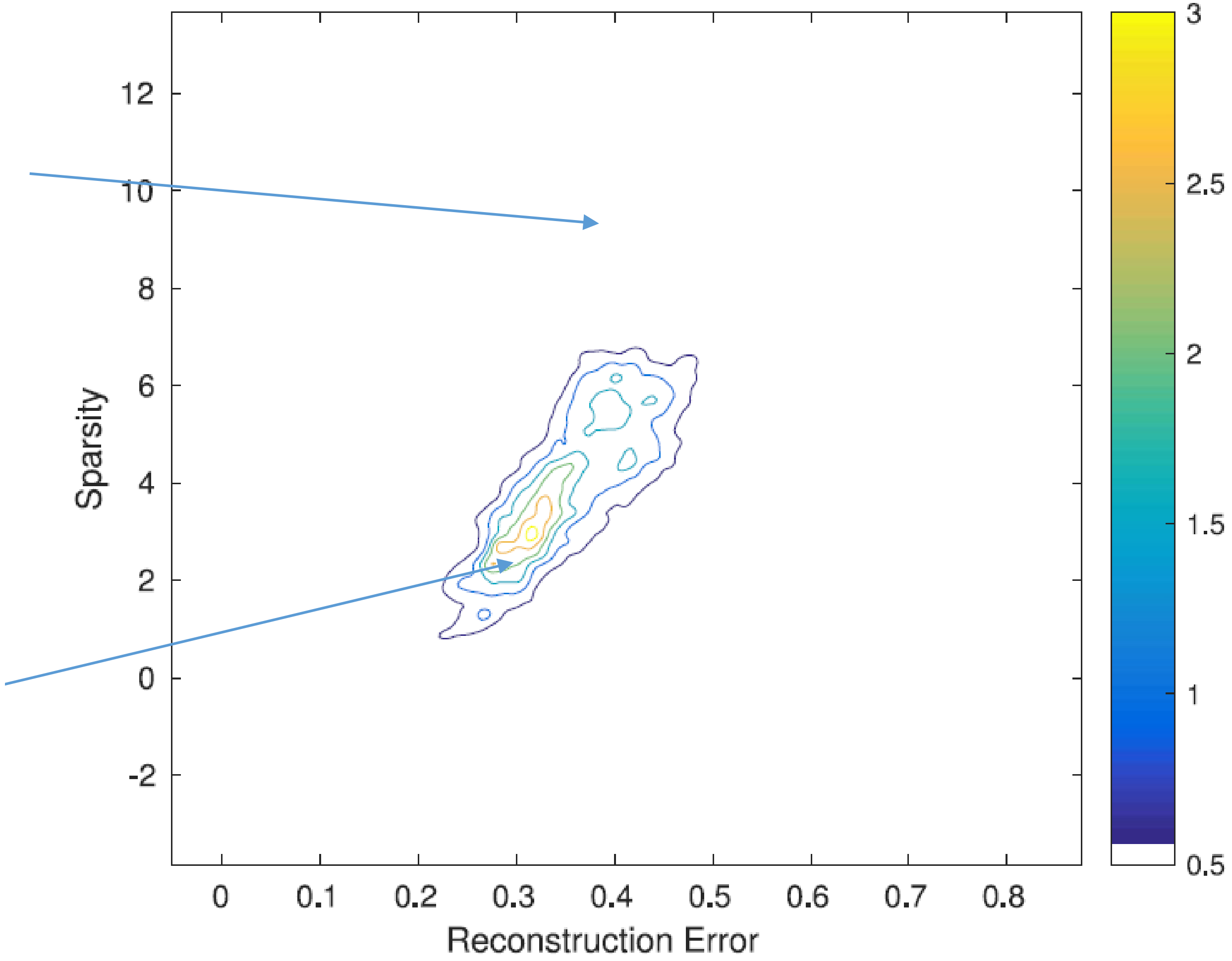
**Anomalies**



**Normal data**







# FEATURES EXTRACTED FROM SPARSE CODING

**Training:**

- **Learn** from $S$ the dictionary $D$
- **Compute** the sparse representation w.r.t. $D$, thus features $\boldsymbol{x}$ over the validation set $V$, such that $V \cap S = \emptyset$
- **Learn** from $V$, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$ and the threshold $\gamma$.

The model for anomaly detection is $(D, \hat{\phi}_0, \gamma)$

**Testing:**

- Perform sparse coding of a test signal $\boldsymbol{s}$, thus get the feature vector $\boldsymbol{x}$
- Detect anomalies when $\mathcal{A}(\boldsymbol{s}) = \hat{\phi}_0(\boldsymbol{x}) < \gamma$

D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

# FEATURES EXTRACTED FROM SPARSE CODING

**Training:**

- **Learn** from $S$ the dictionary $D$
- **Compute** the sparse representation w.r.t. $D$, thus features $\boldsymbol{x}$ over the validation set $V$, such that $V \cap S = \emptyset$
- **Learn** hreshold $\gamma$.

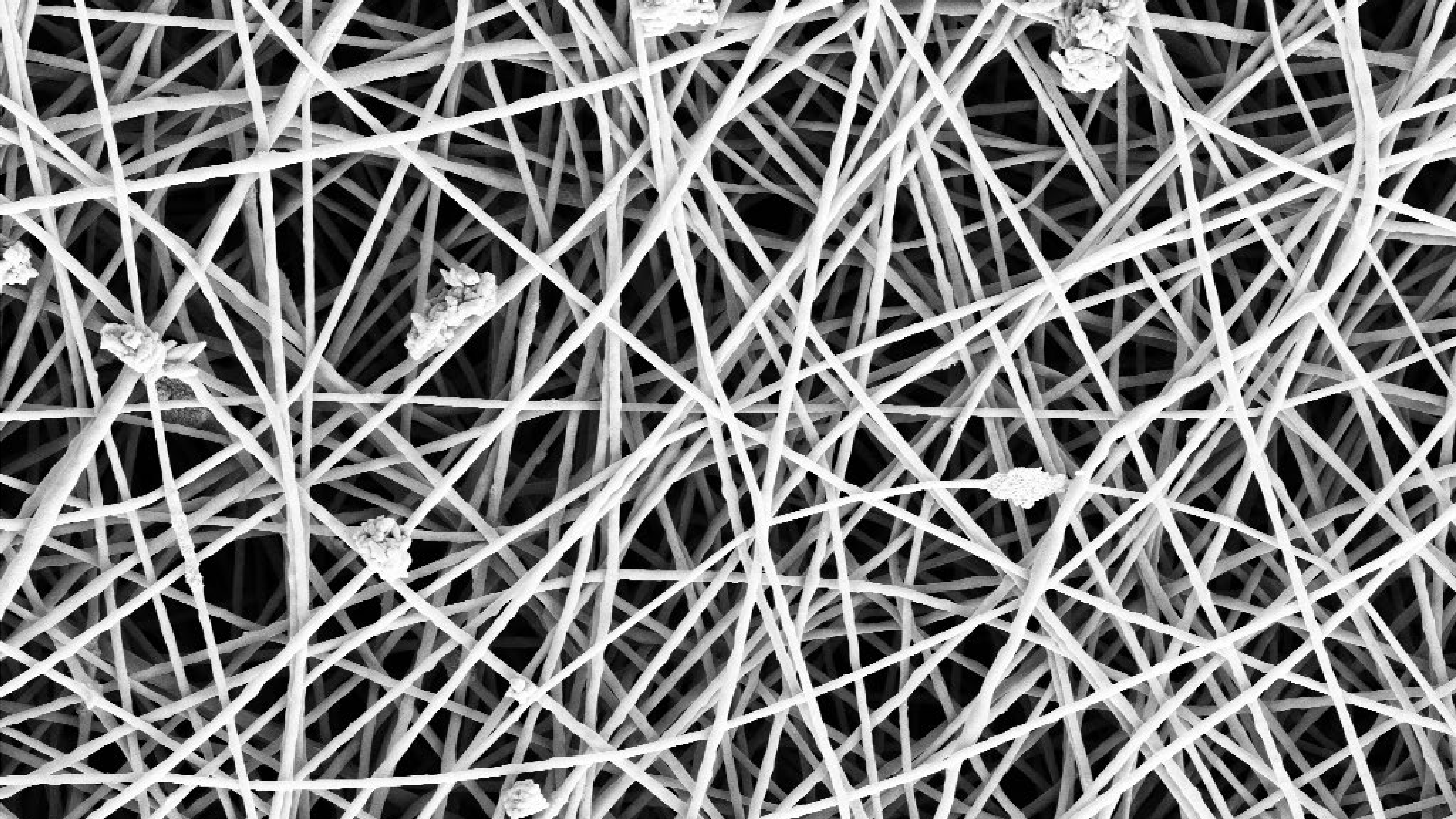
The mod

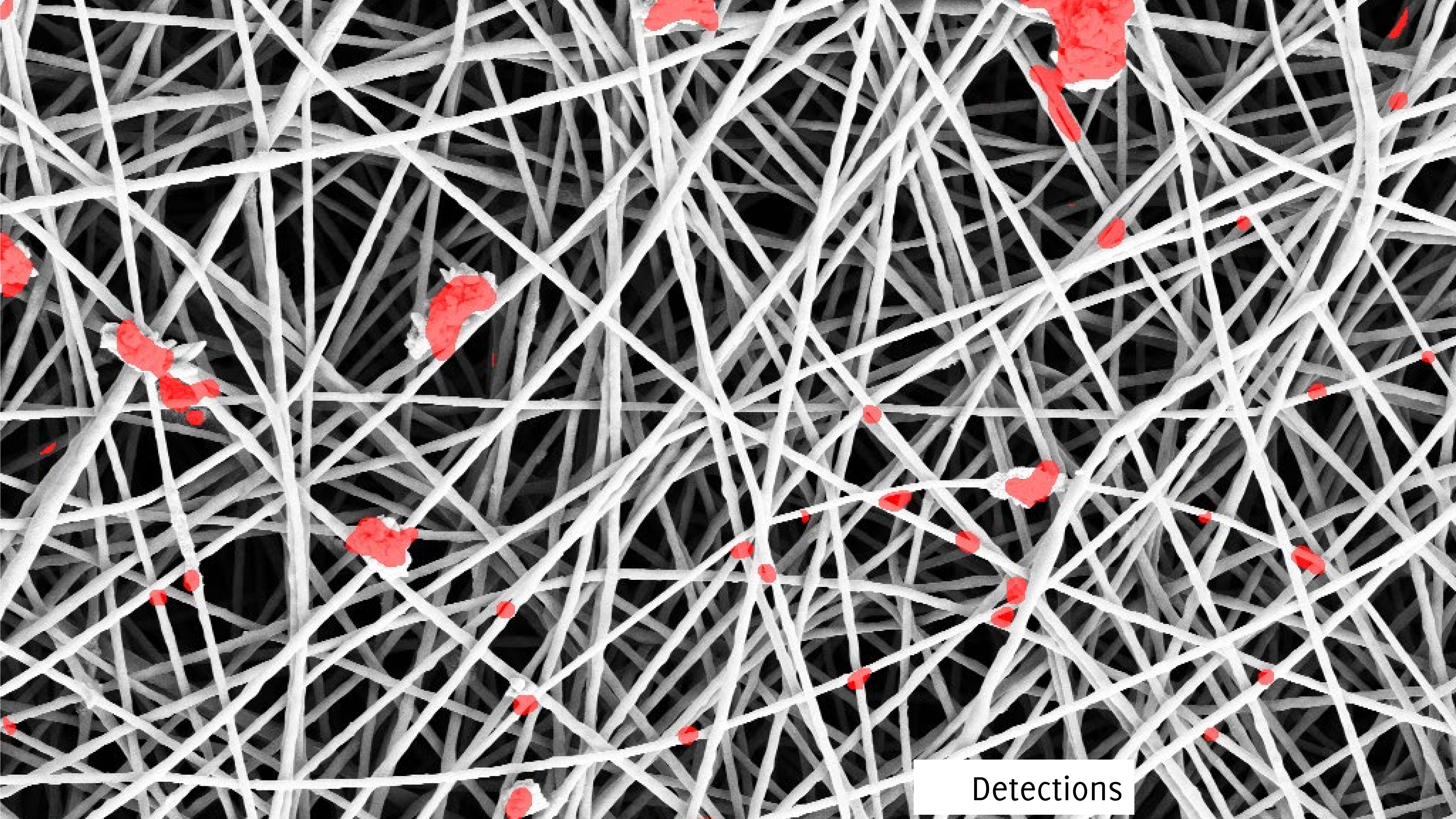
**Testing:**

- Perfor
- Detect anomalies when $\phi_0(\boldsymbol{x}) < \gamma$

**This is rather a flexible solution and can be adapted when operating conditions changes (e.g. heartrate changes, images are acquired at different zooming level)**

D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472




Detections

# Convolutional Sparsity

**Convolutional sparse models** are a recent development of sparse representations

$$\boldsymbol{s} \approx \sum_{i=1}^{n} \boldsymbol{d_i} \circledast \boldsymbol{\alpha_i} \,, \qquad \text{s.t. } \boldsymbol{\alpha_i} \text{ is sparse}$$

where a signal $\boldsymbol{s}$ is **entirely encoded** as the sum of $n$ convolutions between a filter $\boldsymbol{d_i}$ and a coefficient map $\boldsymbol{\alpha_i}$

**Pros**:

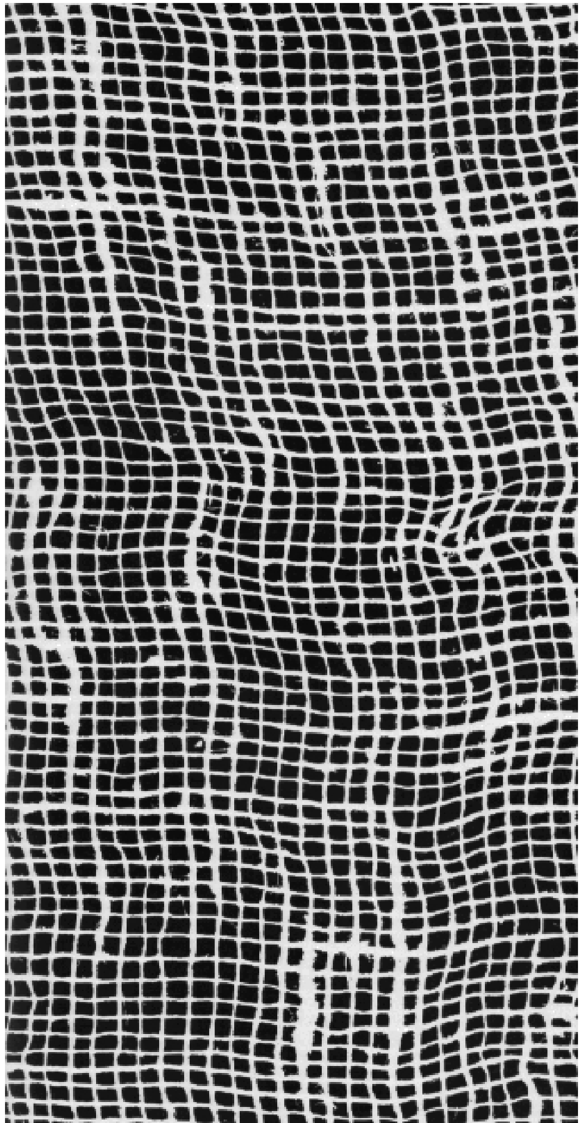
- Translation invariant representation
- Few small filters are typically required
- Filters exhibit very specific image structures
- Easy to use filters having different size

**Collaboration with Los Alamos National Laboratory, NM, USA**

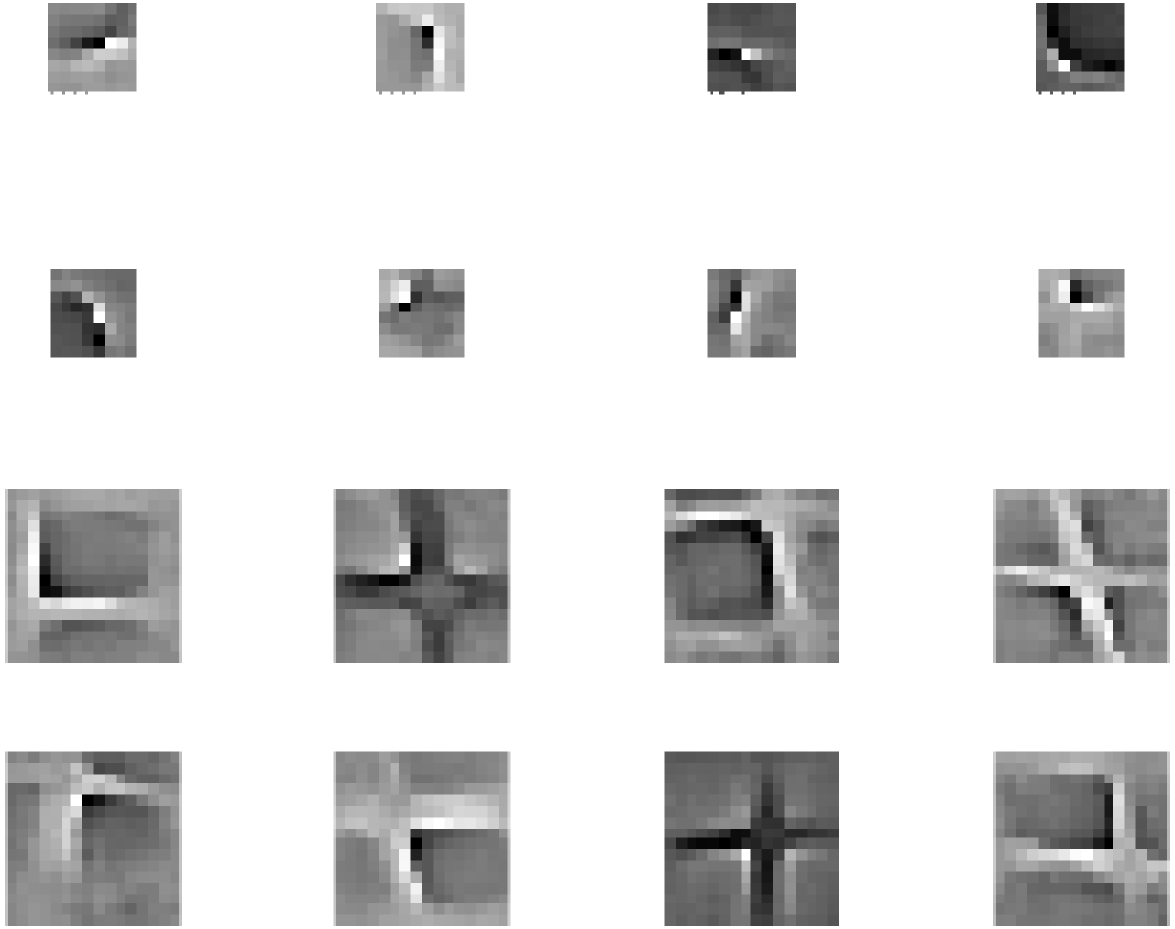
Los Alamos
NATIONAL LABORATORY

# Example of Learned Filters

**Training Image**



**Learned Filters**



# Convolutional Sparsity for Anomaly Detection

If we consider the convolutional sparse coding

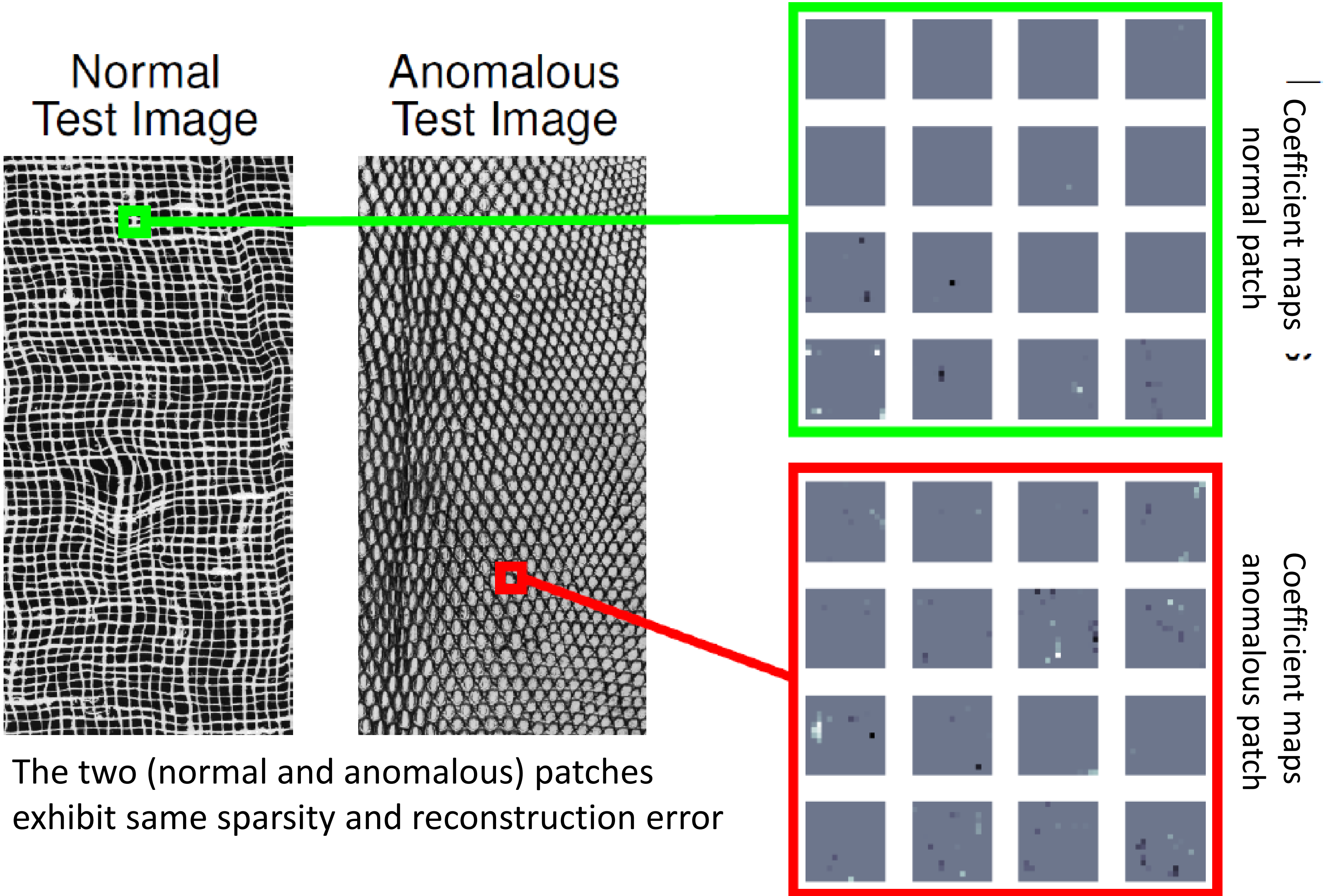
$$\{\hat{\boldsymbol{\alpha}}\} = \underset{\{\boldsymbol{\alpha}\}_n}{\operatorname{argmin}} \left\| \sum_{i=1}^{n} \underline{\boldsymbol{d}_i \circledast \boldsymbol{\alpha}_i - \mathbf{s} \right\|_2^2} + \lambda \sum_{i=1}^{n} \underline{\|\boldsymbol{\alpha}\|_1}$$

we can build the feature vector as:

$$\boldsymbol{x}_c = \begin{bmatrix} \left\| \prod_{c} \left( \sum_{i=1}^{n} \boldsymbol{d}_i \circledast \hat{\boldsymbol{\alpha}}_i - \mathbf{s} \right) \right\|_2^2 \\ \sum_{i=1}^{n} \left\| \prod_{c} \hat{\boldsymbol{\alpha}} \right\|_1 \end{bmatrix}$$

…but unfortunately, detection performance are rather poor

# Sparsity is too loose a criterion for detection



Normal Test Image

Anomalous Test Image

Coefficient maps normal patch

Coefficient maps anomalous patch

The two (normal and anomalous) patches exhibit same sparsity and reconstruction error

# Convolutional Sparsity for Anomaly Detection

## Contributions:

- Design a **feature vector** that accounts for the number of filters that are activated within each region

$$x_c = \begin{bmatrix} \left\| \prod_{\boldsymbol{c}} \left( \sum_{i=1}^{m} \boldsymbol{d_i} \circledast \widehat{\boldsymbol{\alpha}}_{\boldsymbol{i}} - \mathbf{s} \right) \right\|_{\mathbf{2}}^{\mathbf{2}} \\ \sum_{i=1}^{m} \left\| \prod_{\boldsymbol{c}} \widehat{\boldsymbol{\alpha}} \right\|_1 \\ \sum_{i=1}^{m} \left\| \prod_{\boldsymbol{c}} \widehat{\boldsymbol{\alpha}} \right\|_2 \end{bmatrix}$$

**[IJCNN 2015]** D. Carrera, G. Boracchi, A. Foi and B. Wohlberg , *“Detecting Anomalous Structures by Convolutional Sparse Models “* IEEE IJCNN 2015

# Convolutional Sparsity for Anomaly Detection

## Contributions:

- Design a **feature vector** that accounts for the number of filters that are activated within each region
- Design an **efficient sparse coding** algorithm that includes a term penalizing the local group sparsity

$$\{\widehat{\boldsymbol{\alpha}}\} = \underset{\{\boldsymbol{\alpha}\}_m}{\operatorname{argmin}} \left\| \sum_{i=1}^{m} \boldsymbol{d_i} \circledast \boldsymbol{\alpha_i} - \mathbf{s} \right\|_{\mathbf{2}}^{\mathbf{2}} + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}\|_1 + \xi \sum_{\boldsymbol{c}} \sum_{i=1}^{m} \left\| \prod_{\boldsymbol{c}} \boldsymbol{\alpha} \right\|_2$$

**[IJCNN 2015]** D. Carrera, G. Boracchi, A. Foi and B. Wohlberg , *"Detecting Anomalous Structures by Convolutional Sparse Models "* IEEE IJCNN 2015

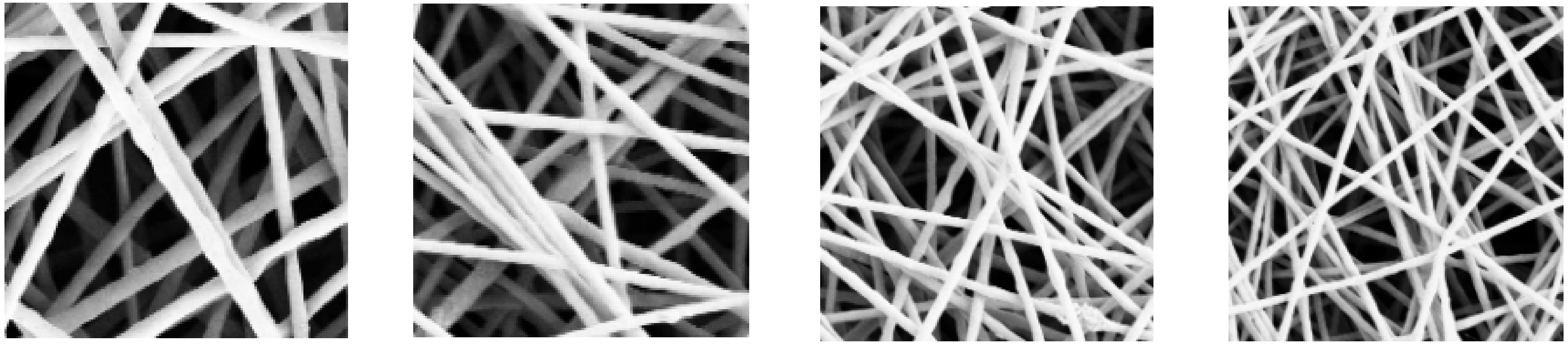
# Counteracting Domain Shift in Anomaly Detection

Adaptation Strategies

# NEED FOR ADAPTATION

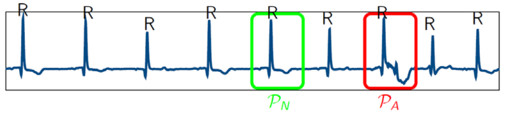
## A challenge often occurring when performing online monitoring

Test data might differ from training data: **need of adaptation, otherwise anomaly detection methods would be ineffective**



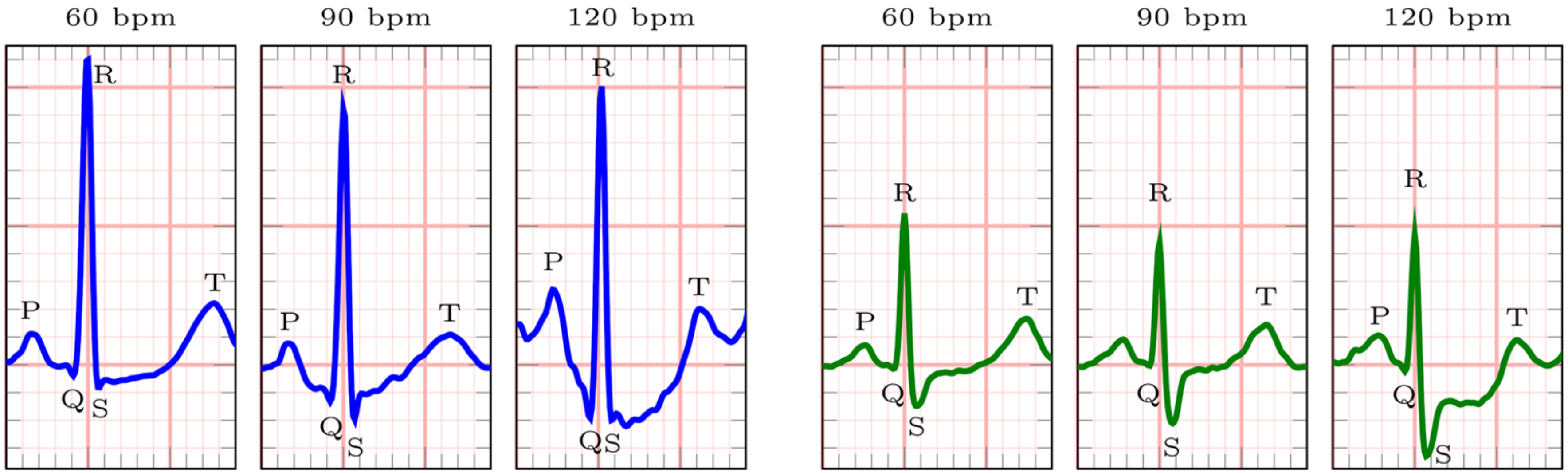
**Defects have to be detected at different zooming levels, that might not be present in the training set.**

# NEED FOR ADAPTATION



## A challenge often occurring when performing online monitoring

Test data might differ from training data: **need of adaptation, otherwise anomaly detection methods would be ineffective**



The **heartbeats** get **transformed** when the **heart rate changes**: **learned models have to be adapted** according to the heart rate.

# MODEL ADAPTATION

In the machine-learning literature these problems go under the name of transfer learning / domain adaptation

**Transfer Learning (TL): adapt a model learned in the *source domain*** (e.g. heartbeats at a given heartrate / fibers at a certain zoom level) **to a *target domain*** (e.g. heartbeats at an higher heartrate / fibers zoomed in or out)

Many TL methods have been designed for supervised / semi-supervised / unsupervised methods, depending on the availability of (annotated) data in the source and target domains.

In most anomaly detection settings, **no labels in the target data are provided** (typically they are not even provided in the source domain)

**S. J. Pan and Q. Yang *"A survey on transfer learning"* IEEE TKDE 2010**

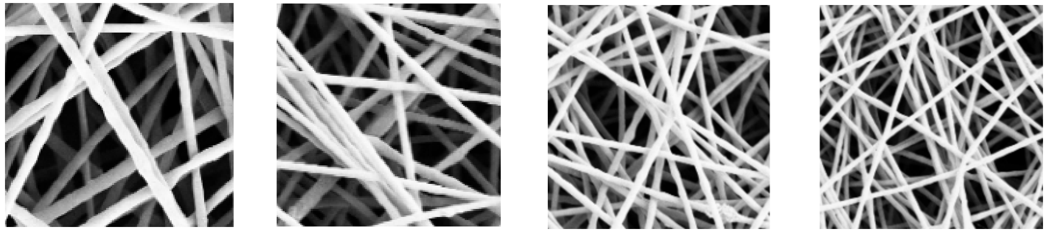
**S. Shekhar, V. M. Patel, H. V. Nguyen, & R. Chellappa, *"Generalized domain-adaptive dictionaries,"* CVPR 2013**

# DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

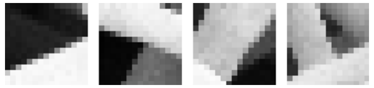
**Solution**:

- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** $D_i$ at each scale
- Combine the learned dictionaries in a **multiscale dictionary** $D$



$$D = [ \quad D_1 \qquad D_2 \qquad D_3 \qquad D_4 \quad ]$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016



# DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

**Solution**:

- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** $D_i$ at each scale
- Combine the learned dictionaries in a **multiscale dictionary** $D$
- **Sparse-coding** including a penalized, **group sparsity term**

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\left|\left|\boldsymbol{s} - D\boldsymbol{a}\right|\right|_2^2 + \lambda\left|\left|\boldsymbol{a}\right|\right|_{\mathbf{1}} + \mu\sum_{\boldsymbol{i}}\left|\left|\boldsymbol{a}\right|\right|_2$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016

# DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

**Solution**:

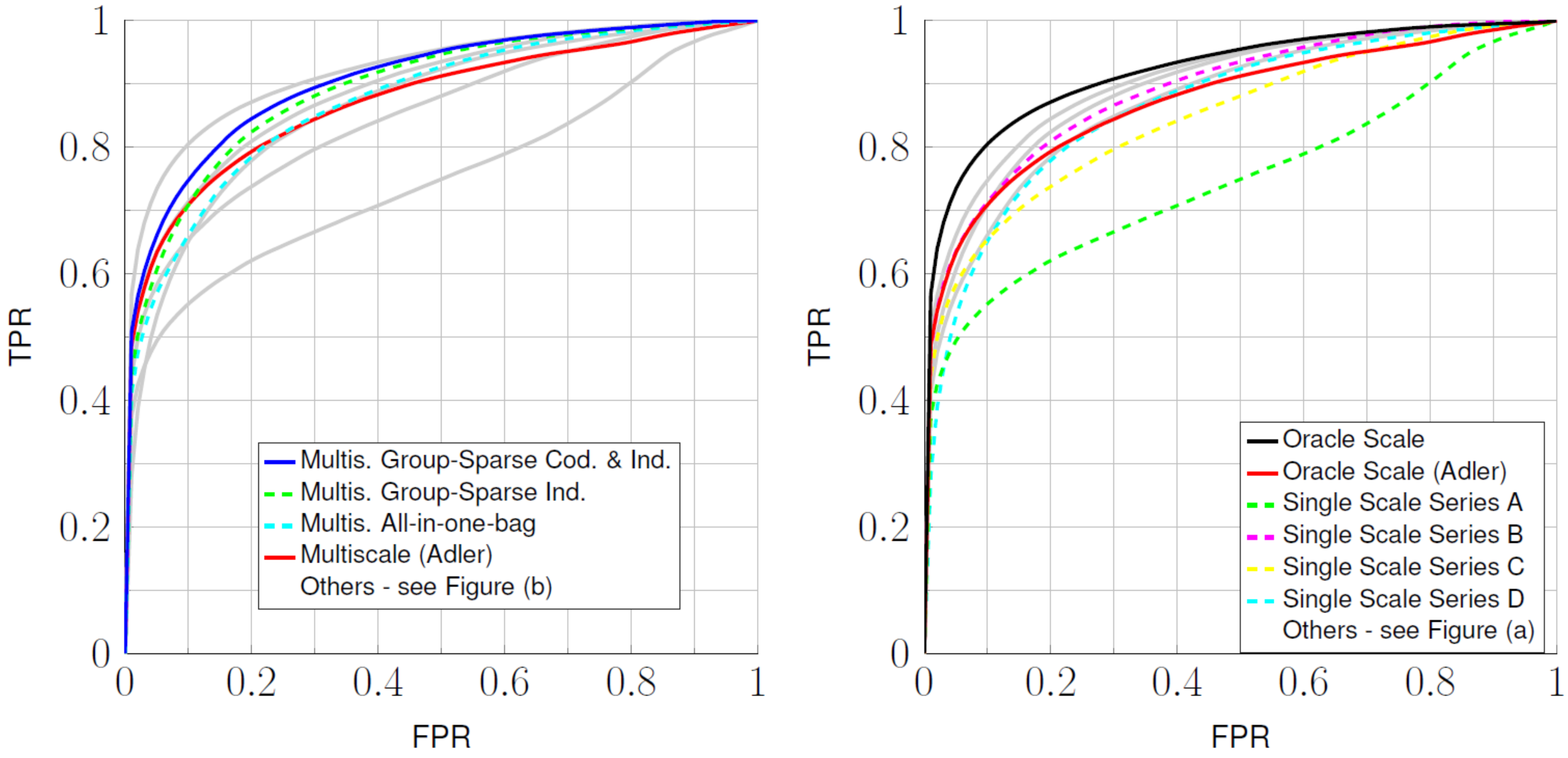
- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** $D_i$ at each scale
- Combine the learned dictionaries in a **multiscale dictionary** $D$
- **Sparse-coding** including a penalized, **group sparsity term**
- Monitor a tri-variate feature vector

$$\boldsymbol{x} = \begin{bmatrix} ||\boldsymbol{s} - D\boldsymbol{\alpha}||_2^2 \\ ||\boldsymbol{\alpha}||_1 \\ \sum_{\boldsymbol{i}} ||\boldsymbol{\alpha_i}||_2 \end{bmatrix}$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016

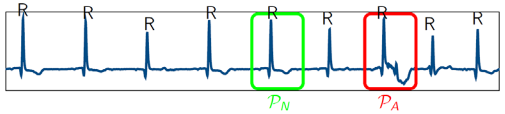
# DOMAIN ADAPTATION ON QUALITY INSPECTION

Performance on SEM image dataset acquired at 4 different zooming levels (A,B,C,D). It is important to include group-sparsity regularization also in the sparse coding stage



D. Carrera, G. Boracchi, A. Foi and B. Wohlberg "Scale-invariant Anomaly Detection With multiscale Group-sparse Models" ICIP 2016

# DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



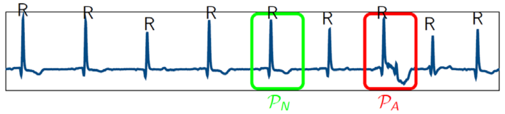
We propose to design linear transformations $F_{r_1,r_0}$ to adapt user-specific dictionaries

$$D_{u,r_1} = F_{r_1,r_0} \cdot \ D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m\times m}$$

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

# DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We propose to design linear transformations $F_{r_1,r_0}$ to adapt user-specific dictionaries

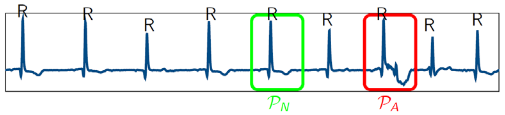
$$D_{u,r_1} = F_{r_1,r_0} \cdot \, D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m \times m}$$

Surprisingly **these transformations can be learned from a publicly available dataset** containing ECG recordings at different heart rates from several users.

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

# LEARNING TRANSFORMATIONS



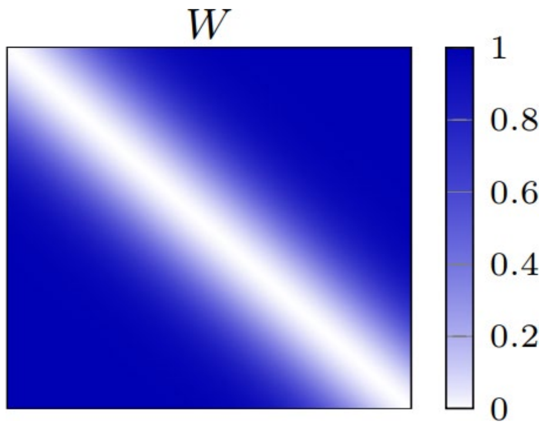
For each pair of heartrates $(r_0, r_1)$ we learn $F_{r_0,r_1}$ by solving the following optimization problem (involving data from $L$ users of the LS-ST Dataset)

$$F_{r_1,r_0} = \underset{F,\{X_u\}}{\operatorname{argmin}} \left( \frac{1}{2} \sum_{u=1}^{L} \left\| S_{u,r_1} - F\, D_{u,r_0}\, X_u \right\|_F^2 + \mu \sum_{u=1}^{L} \|X_u\|_1 + \frac{\lambda}{2} \|W \odot F\|_2^2 + \xi \|W \odot F\|_1 \right)$$

**Data-fidelity for heartbeats transformed by $F$, computed over all the $L$ users**

**Sparsity**

**Weighted elastic net regularization to add stability and steer $F$ towards desirable properties**

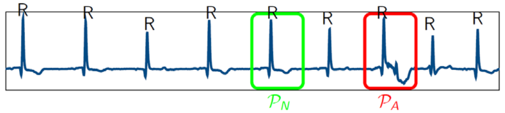


**The matrix $W$ is penalyzing less values along the diagonal of $F$, thus assuming transformation to be local, i.e., involging only neighbouring samples**

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

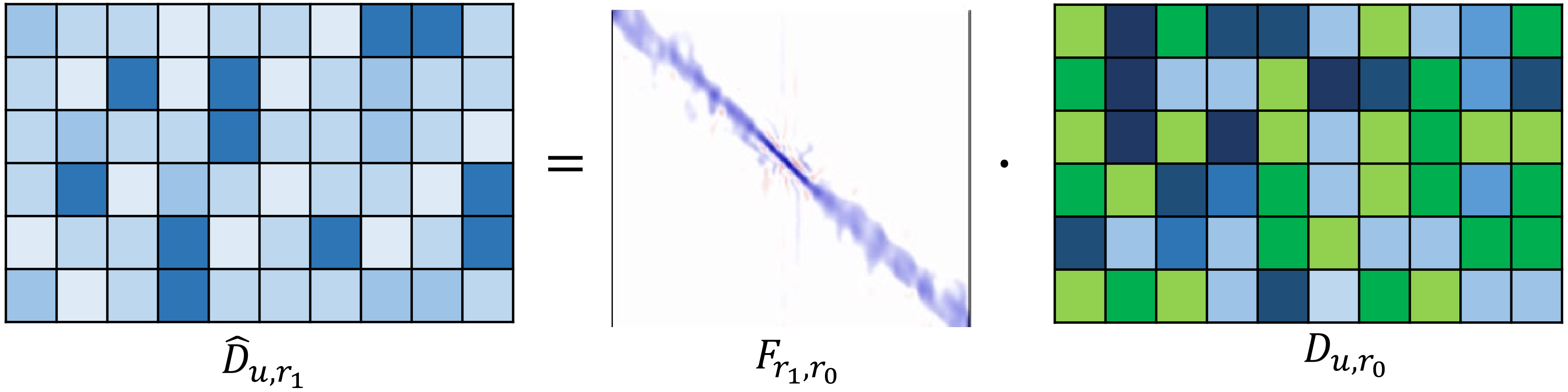
# DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We adapt user-specific dictionaries through $F_{r_1,r_0}$

$$D_{u,r_1} = F_{r_1,r_0} \cdot D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m\times m}$$

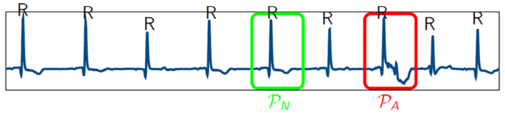
**User-independent transformations enable accurate mapping of user-specific dictionaries**



D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

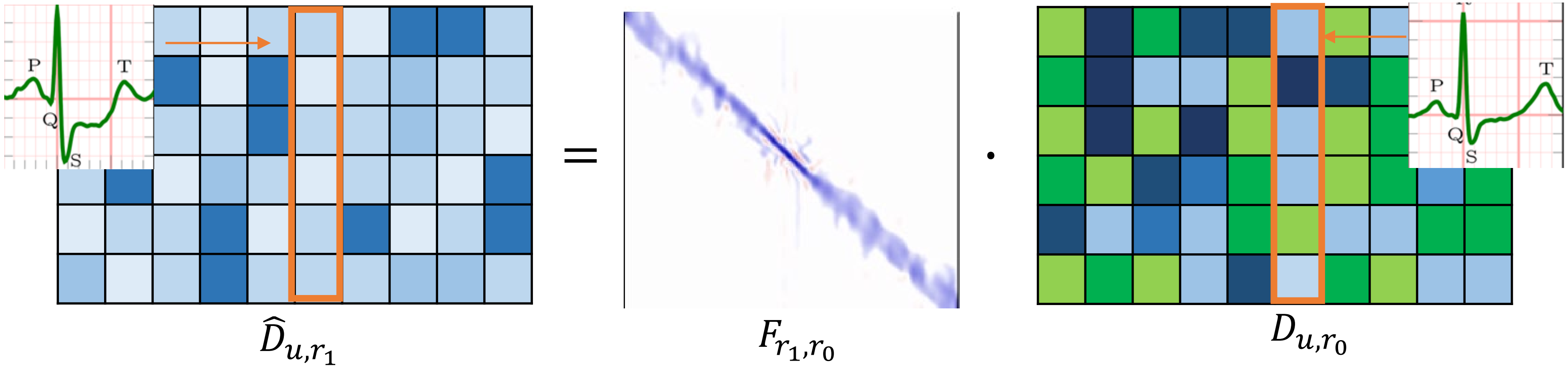
# DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We adapt user-specific dictionaries through $F_{r_1,r_0}$

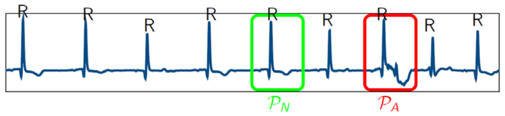
$$D_{u,r_1} = F_{r_1,r_0} \cdot \; D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m\times m}$$

**User-independent transformations enable accurate mapping of user-specific dictionaries**



Carrera D., Rossi B., Fragneto P., and Boracchi G. "Domain Adaptation for Online ECG Monitoring" ICDM 2017,

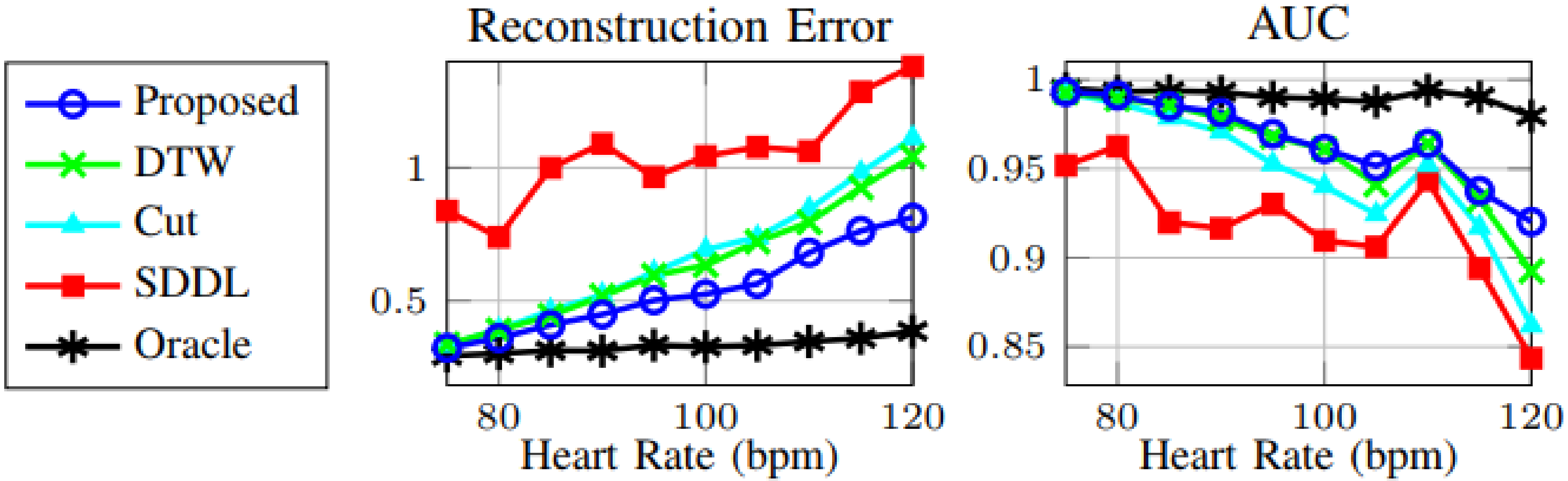
# DICTIONARY ADAPTATION PERFORMANCE



The proposed domain adaptation solution achieves:

- lowest signal reconstruction error
- best anomaly detection performance (AUC)

Among alternative methods for dictionary adaptation



Carrera D., Rossi B., Fragneto P., and Boracchi G. "Domain Adaptation for Online ECG Monitoring” ICDM 2017,

# Assignments & References

# Assignments

- Implement the anomaly detection based on l1 sparse coding
  - Use 15x15 patches
  - You can improve the results by fine tuning all the parameters
- Implement the classification based on sparse representation

# References

- ADMM: Wahlberg, Bo, et al. "An ADMM algorithm for a class of total variation regularized estimation problems." *IFAC Proceedings Volumes* 45.16 (2012): 83-88.
- Anomaly Detection:
  - Carrera, Diego, et al. "Defect detection in SEM images of nanofibrous materials." *IEEE Transactions on Industrial Informatics* 13.2 (2016): 551-561.
  - Carrera, Diego, et al. "Scale-invariant anomaly detection with multiscale group-sparse models." *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016.
- Classification: J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210–227, February 2009. doi:10.1109/tpami.2008.79