

RESEARCH ARTICLE

A competitive scheme for storing sparse representation of X-Ray medical images

Laura Rebollo-Neira*

Mathematics Department, Aston University, B4 7ET Birmingham, United Kingdom

* l.rebollo-neira@aston.ac.uk

Abstract

A competitive scheme for economic storage of the informational content of an X-Ray image, as it can be used for further processing, is presented. It is demonstrated that sparse representation of that type of data can be encapsulated in a small file without affecting the quality of the recovered image. The proposed representation, which is inscribed within the context of data reduction, provides a format for saving the image information in a way that could assist methodologies for analysis and classification. The competitiveness of the resulting file is compared against the compression standards JPEG and JPEG2000.



OPEN ACCESS

Citation: Rebollo-Neira L (2018) A competitive scheme for storing sparse representation of X-Ray medical images. PLoS ONE 13(8): e0201455. <https://doi.org/10.1371/journal.pone.0201455>

Editor: Viktor Vegh, University of Queensland, AUSTRALIA

Received: November 2, 2017

Accepted: July 16, 2018

Published: August 16, 2018

Copyright: © 2018 Laura Rebollo-Neira. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are available at <https://doi.org/10.17036/researchdata.aston.ac.uk.00000370>.

Funding: There was no specific funding for this work.

Competing interests: The author has declared that no competing interests exist.

1 Introduction

Sparse image representation refers to particular techniques for data reduction. Rather than representing the image informational content by the intensity of the pixels, the image is transformed with the aim of reducing the number of data points for reproducing the equivalent information. Lessening the cardinality of medical data is crucial for remote diagnosis and treatments [1–4]. The interest for this matter in the area of medical technology has recently been invigorated by the prospect of earlier disease detection, using neural networks and deep learning methodologies for automatic analysis of X-Ray plates [5–7].

In recent work [8] we have demonstrated that sparse representation, obtained by a large dictionary and greedy algorithms, renders high quality approximation of X-Ray images. The framework was proven to produce approximations which are far more sparser than those arising from traditional transformations such as the Cosine and Wavelet Transforms. Nevertheless, for the approximation to be useful within the context of automatic health care systems and remote reporting, it is necessary to ensure that the high levels of the achieved sparsity are not affected by saving the reduced data in a small file. This paper follows on the study in [8] by presenting a simple scheme to store the sparse approximation of an X-Ray medical image in a file of competitive size with respect to the most commonly used formats, namely JPEG and JPEG2000 (JPEG2).

Although in general X-Ray images are sparser if the approximation is performed in the wavelet domain, for analysis purposes the approximation in the pixel intensity domain may be needed. Hence, we consider here approximations in both domains. It is pertinent to stress that the aim of this work is not to propose yet one more method for image compression, but a

method for economic storage of the informational content of an X-Ray image as it can be used for further processing [7, 9–15]. The comparison of the resulting file size, against those produced by JPEG and JPEG2, is carried out with the purpose of illustrating the effectiveness of the proposed scheme.

2 Sparse image representation

Throughout the paper \mathbb{R} stands for the set of real numbers. Lower boldface letters are used to represent vectors and upper boldface letters to represent matrices. Standard mathematical fonts indicate their corresponding components, e.g., $\mathbf{d} \in \mathbb{R}^N$ is a vector of components $d(i)$, $i = 1, \dots, N$ and $\mathbf{I} \in \mathbb{R}^{N_x \times N_y}$ a matrix of elements $I(i, j)$, $i = 1, \dots, N_x, j = 1, \dots, N_y$.

Suppose that an image, given as a 2D array $\mathbf{I} \in \mathbb{R}^{N_x \times N_y}$ of intensity values, is to be approximated by the linear decomposition

$$\mathbf{I}^k = \sum_{n=1}^k c(n) \mathbf{D}_{\ell_n}, \tag{1}$$

where each $c(n)$ is a scalar and each \mathbf{D}_{ℓ_n} is an element of $\mathbb{R}^{N_x \times N_y}$ normalized to unity, called an ‘atom’, to be selected from a redundant set, $\mathcal{D} = \{\mathbf{D}_n\}_{n=1}^M$, called a ‘dictionary’. The selection of the atoms \mathbf{D}_{ℓ_n} in (1) is carried out according to an optimality criterion.

The goal of a sparse representation is to produce an approximation \mathbf{I}^k , of an image \mathbf{I} , using as few atoms as possible. The mathematical methods for performing the task are either based on the minimization of the l_1 -norm [16, 17] or are greedy strategies for stepwise selection of atoms from the dictionary [18, 19]. When using large dictionaries, the latter are especially suited for practical applications. We focus here on the greedy algorithm known as Orthogonal Matching Pursuit (OMP), which is very effective at furnishing sparse solutions. In spite of the fact that the method is computationally intensive, the implementation described in the next section is very efficient in terms of processing time.

2.1 Orthogonal Matching Pursuit in 2D

The OMP method was introduced in [19] and has been implemented by a number of different algorithms. We describe a particular implementation for 2D, henceforth referred to as OMP2D. The dictionary is restricted to be separable, i.e. a 2D dictionary \mathcal{D} which corresponds to the tensor product of two 1D dictionaries $\mathcal{D} = \mathcal{D}^x \otimes \mathcal{D}^y$. Our implementation of OMP2D is based on adaptive biorthogonalization and Gram-Schmidt orthogonalization procedures, as proposed in [20] for the one dimensional case. For the convenience of the interested researcher we include the description of its generalization to separable 2D dictionaries, as it is implemented by the MATLAB and C++ MEX functions we have made available.

Given a gray level intensity image, $\mathbf{I} \in \mathbb{R}^{N_x \times N_y}$, and two 1D dictionaries $\mathcal{D}^x = \{\mathbf{d}_n^x \in \mathbb{R}^{N_x}\}_{n=1}^{M_x}$ and $\mathcal{D}^y = \{\mathbf{d}_m^y \in \mathbb{R}^{N_y}\}_{m=1}^{M_y}$, we approximate the array $\mathbf{I} \in \mathbb{R}^{N_x \times N_y}$ by the atomic decomposition of the form:

$$\mathbf{I}^k = \sum_{n=1}^k c_n \mathbf{A}_n = \sum_{n=1}^k c_n \mathbf{d}_{\ell_n}^x (\mathbf{d}_{\ell_n}^y)^T, \tag{2}$$

where $(\mathbf{d}_{\ell_n}^y)^T$ indicates the transpose of the column vector $\mathbf{d}_{\ell_n}^y$. The OMP2D approach determines the atoms $\mathbf{A}_n = \mathbf{d}_{\ell_n}^x (\mathbf{d}_{\ell_n}^y)^T$ in (2) as follows:

On setting $k = 0$ and $\mathbf{R}^0 = \mathbf{I}$ at iteration $k + 1$ the algorithm selects the elements $\mathbf{d}_{\ell_{k+1}}^x \in \mathcal{D}^x$ and $\mathbf{d}_{\ell_{k+1}}^y \in \mathcal{D}^y$ corresponding to the indices obtained as

$$\ell_{k+1}^x, \ell_{k+1}^y = \arg \max_{\substack{n=1, \dots, M_x \\ m=1, \dots, M_y}} |(\mathbf{d}_n^x)^T \mathbf{R}^k \mathbf{d}_m^y|,$$

with (3)

$$\mathbf{R}^k = \mathbf{I} - \sum_{n=1}^k c(n) \mathbf{d}_n^x (\mathbf{d}_n^y)^T.$$

The coefficients $\{c(n)\}_{n=1}^k$ in (3) are such that $\|\mathbf{R}^k\|_F$ is minimum ($\|\cdot\|_F$ being the Frobenius norm induced by the Frobenius inner product $\langle \cdot, \cdot \rangle_F$). This is guaranteed by calculating the coefficients as

$$c_n = \langle \mathbf{B}_n^k, \mathbf{I} \rangle_F, \quad n = 1, \dots, k, \tag{4}$$

where the matrices $\{\mathbf{B}_n^k\}_{n=1}^k$ are recursively constructed, at each iteration, to account for each newly selected atom. Starting from $\mathbf{B}_1^1 = \mathbf{W}_1 = \mathbf{A}_1 = \mathbf{d}_{\ell_1^x}^x (\mathbf{d}_{\ell_1^y}^y)^T$ the set of matrices is upgraded and updated through the formula:

$$\mathbf{B}_n^{k+1} = \mathbf{B}_n^k - \mathbf{B}_{k+1}^{k+1} \langle \mathbf{A}_{k+1}, \mathbf{B}_n^k \rangle_F, \quad n = 1, \dots, k,$$

where

$$\mathbf{B}_{k+1}^{k+1} = \mathbf{W}_{k+1} / \|\mathbf{W}_{k+1}\|_F^2, \quad \text{with} \tag{5}$$

$$\mathbf{W}_{k+1} = \mathbf{A}_{k+1} - \sum_{n=1}^k \frac{\mathbf{W}_n}{\|\mathbf{W}_n\|_F^2} \langle \mathbf{W}_n, \mathbf{A}_{k+1} \rangle_F.$$

For numerical accuracy of the orthogonal set $\{\mathbf{W}_n\}_{n=1}^{k+1}$ at least one re-orthogonalization step is usually needed. It requires the recalculation of the element \mathbf{W}_{k+1} as

$$\mathbf{W}_{k+1} \leftarrow \mathbf{W}_{k+1} - \sum_{n=1}^k \frac{\mathbf{W}_n}{\|\mathbf{W}_n\|_F^2} \langle \mathbf{W}_n, \mathbf{W}_{k+1} \rangle_F. \tag{6}$$

The algorithm iterates until for a given parameter ρ the stopping criterion $\|\mathbf{I} - \mathbf{I}^k\|_F^2 < \rho$ is met.

2.2 Approximation by partitioning

The OMP2D approach described above is to be applied on an image partition. This implies the division of the image into small disjoint blocks $\mathbf{I}_q, q = 1, \dots, Q$, which without loss of generality we assume to be square of dimension $N_b \times N_b$. Denoting each element of the image partition by \mathbf{I}_q , these are approximated, independently of each other, to produce the approximations $\mathbf{I}_q^{k_q}$, $q = 1, \dots, Q$, of the form (2). The superscript k_q indicates the number of atoms intervening in the decomposition of the block q . As already mentioned, in spite of the fact that the approximation of most X-Ray images is significantly sparser if performed in the wavelet domain, results in the pixel intensity domain may be required for particular applications.

The approximation in the wavelet domain involves a transformation, via a Wavelet Transform, which converts the image into the array to be approximated. The inverse transformation

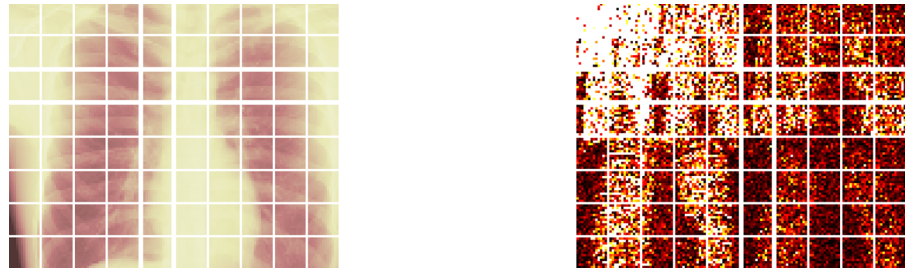


Fig 1. Portion of a chest image. Illustration of an X-Ray image partition in the pixel intensity domain (left graph) and in the wavelet domain (right graph). The image size is 256×320 pixels and the partition is illustrated by blocks of 32×32 pixels. The colours are added for visual effect.

<https://doi.org/10.1371/journal.pone.0201455.g001>

is taken after the approximation is concluded. Fig 1 provides graphical illustrations of image partitions in the pixel intensity and the wavelet domain.

Remark 1: The OMP2D algorithm described in Sec. 2.1 is very effective up to some block size. While the actual size depends on the sparsity of the image, previous studies indicate that in general for block sizes greater than 24×24 an alternative implementation is advisable. The alternative implementation, called Self Projected Matching Pursuit (SPMP2D) [8, 21] is not only dedicated to tackling large dimensional problems, but also potentially suitable for implementation in Graphic Processing Unit (GPU) programming. However, because for X-Ray medical images a partition into blocks of size 16×16 is a good compromise between sparsity and processing time, we have focussed on the implementation of OMP2D as given in Sec. 2.1. Other possibilities for approximating a partition are discussed in [8, 22].

2.3 Mixed dictionaries

The available literature in relation to the construction of dictionaries for image representation is mainly concerned with methodologies for learning atoms from training data [23–28]. Those methodologies are not designed for learning the types of dictionaries of our interest though. We restrict the dictionary to be separable, in order to reduce the computational burden and memory requirements. In previous works [8, 21, 29] we have demonstrated that very large separable dictionaries, which are easy to construct, render high levels of sparsity. Such dictionaries are not specific to a particular class of images. A discrimination is only made to take into account whether the approximation is carried out in the pixel intensity or in the wavelet domain. In each domain we use mixed dictionaries which are similar in nature, but not equal. They have the trigonometric dictionaries \mathcal{D}_C^x and \mathcal{D}_S^x , defined below, as common components.

$$\mathcal{D}_C^x = \{w_c(n) \cos \frac{\pi(2i-1)(n-1)}{2M}, i = 1, \dots, N\}_{n=1}^M$$

$$\mathcal{D}_S^x = \{w_s(n) \sin \frac{\pi(2i-1)(n)}{2M}, i = 1, \dots, N\}_{n=1}^M,$$

where $w_c(n)$ and $w_s(n)$, $n = 1, \dots, M$ are normalization factors, and usually $M = 2N$.

For approximations in the wavelet domain we add the dictionary \mathcal{D}_{Lw}^x , as proposed in [8], which is built by translation of the prototype atoms in the right graph of Fig 2. The mixed dictionary \mathcal{D}_{wd}^x , to be used only in the wavelet domain, is formed as $\mathcal{D}_{wd}^x = \mathcal{D}_C^x \cup \mathcal{D}_S^x \cup \mathcal{D}_{Lw}^x$ and $\mathcal{D}_{wd}^y = \mathcal{D}_{wd}^x$.

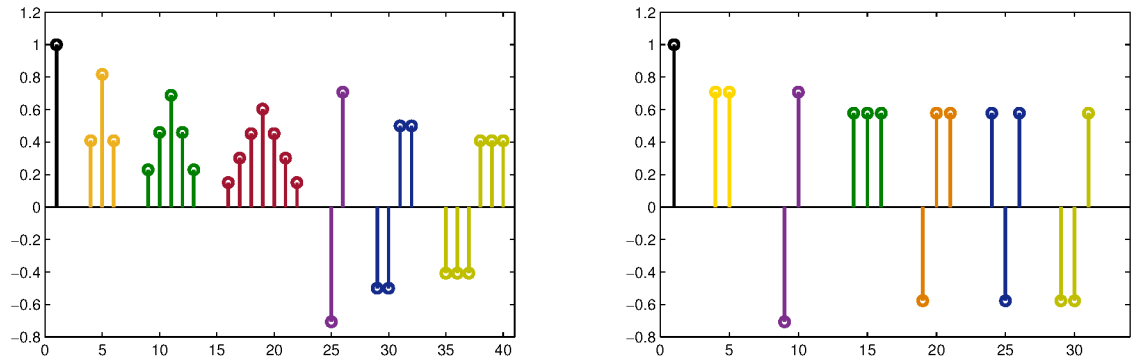


Fig 2. Prototype atoms, which generate by translation the dictionaries D_{LP}^x (left) and D_{Lw}^x (right).

<https://doi.org/10.1371/journal.pone.0201455.g002>

For approximations in the pixel intensity domain we add the dictionary, D_{LP}^x , which is built by translation of the prototype atoms in the left graph of Fig 2. In this case the mixed dictionary D_{pd}^x is formed as $D_{pd}^x = D_C^x \cup D_S^x \cup D_{LP}^x$ and $D_{pd}^y = D_{pd}^y$.

The corresponding 2D dictionaries $D_{wd} = D_{wd}^x \otimes D_{wd}^y$ and $D_{pd} = D_{pd}^x \otimes D_{pd}^y$ are very large, but not needed as such. All calculations are carried out using the 1D dictionaries.

3 Coding strategy

For the sparse representation of an X-Ray image to be useful within the current trend of medical technology developments it should be suitable to be encapsulated in a small file. Accordingly, the coefficients of the atomic decompositions need to be converted into integer numbers. This operation is known as quantization. We adopt a simple and commonly used uniform quantization technique. For $q = 1, \dots, Q$ the absolute value coefficients $|c_q(n)|, n = 1, \dots, k_q$ are converted to integers as follows:

$$c_q^\Delta(n) = \begin{cases} \lceil \frac{|c_q(n)| - \theta}{\Delta} \rceil, & \text{if } |c_q(n)| \geq \theta \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where $\lceil x \rceil$ indicates the smallest integer number greater than or equal to x , Δ is the quantization parameter, and θ the threshold to disregard coefficients of small magnitude. The signs of the coefficient are encoded separately, as a vector s_q , using a binary alphabet.

In order to store the information about the particular atoms present in the approximation of each block, we proceed as follows: Firstly each pair of indices $(\ell_n^{x,q}, \ell_n^{y,q})$ corresponding to the atoms in the decompositions of the block I_q is mapped into a single index $o_q(n)$. Then the set $o_q(1), \dots, o_q(k_q)$ is sorted in ascending order $o_q(n) \rightarrow \tilde{o}_q(n), n = 1, \dots, k_q$. This guarantees that, for each q -value, $\tilde{o}_q(i) < \tilde{o}_q(i + 1), i = 1, \dots, k_q - 1$. The order of the indices induces an order in the unsigned coefficients, $c_q^\Delta \rightarrow \tilde{c}_q^\Delta$ and in the corresponding signs $s_q \rightarrow \tilde{s}_q$. The advantage introduced by the ascending order of the indices is that they can be stored as smaller positive numbers, by taking differences between two consecutive values. Certainly by defining $\delta_q(n) = \tilde{o}_q(n) - \tilde{o}_q(n - 1), n = 2, \dots, k_q$ the string $\tilde{o}_q(1), \delta_q(2), \dots, \delta_q(k_q)$ stores the indices for the block q with unique recovery. The number 0 is then used to separate the strings corresponding to different blocks.

$$st_{ind} = [\tilde{o}_1(1), \delta_1(2), \dots, \delta_1(k_1), 0, \tilde{o}_2(1), \delta_2(2), \dots, \delta_2(k_2), 0, \dots, \tilde{o}_{k_Q}(1), \delta_{k_Q}(2), \dots, \delta_{k_Q}(k_Q)]. \quad (8)$$

The quantized magnitude of the re-ordered coefficients are concatenated in the strings st_{cf} as follows:

$$st_{cf} = [\tilde{c}_1^\Delta(1), \dots, \tilde{c}_1^\Delta(k_1), \tilde{c}_2^\Delta(1), \dots, \tilde{c}_2^\Delta(k_2), \dots, \tilde{c}_{k_Q}^\Delta(1), \dots, \tilde{c}_{k_Q}^\Delta(k_Q)]. \tag{9}$$

Using 0 if the sign is positive and 1 if it is negative, the signs of the coefficients are placed in the string, st_{sg} as

$$st_{sg} = [\tilde{s}_1(1), \dots, \tilde{s}_1(k_1), \tilde{s}_2(1), \dots, \tilde{s}_2(k_2), \dots, \tilde{s}_{k_Q}(1), \dots, \tilde{s}_{k_Q}(k_Q)]. \tag{10}$$

The next encoding/decoding scheme summarizes the above described procedure.

Encoding

- Given an image partition $\mathbf{I}_q \in \mathbb{R}^{N_b \times N_b}$, $q = 1, \dots, Q$ use OMP2D (or other method) to approximate each element of the partition by the atomic decomposition:

$$\mathbf{I}_q^{k_q} = \sum_{n=1}^{k_q} c_q(n) \mathbf{d}_n^{x,q} (\mathbf{d}_n^{y,q})^T. \tag{11}$$

- The approximation is carried out on each block, independently of the others, until the stopping criterion is reached.
- For each q , quantize as in (7) the magnitude of the coefficients in the decomposition (11) to obtain $c_q^\Delta(n)$, $n = 1, \dots, k_q$. Store the signs of the non-zero coefficient as components of a vector \mathbf{s}_q .
- For each q , map the pair of indices $(\ell_n^{x,q}, \ell_n^{y,q})$, $n = 1, \dots, k_q$ in (11) into a single index $o_q(n)$, $n = 1, \dots, k_q$ and sort these numbers in ascending order to have the re-ordered sets: $\tilde{o}_q(1) \dots, \tilde{o}_q(k_q)$; $\tilde{c}_q^\Delta(1), \dots, \tilde{c}_q^\Delta(k_q)$ and $\tilde{s}_q(1), \dots, \tilde{s}_q(k_q)$ to create the strings: st_{ind} , as in (8), and st_{cf} , and st_{sg} as in (9) and (10) respectively.

Let’s recall the content of the file encoded by the above steps:

st_{ind} contains the difference of indices corresponding to the atoms in the approximation of each of the blocks in the image partition.

st_{cf} contains the magnitude of the corresponding coefficients (quantized to integer numbers).

st_{sg} contains the signs of the coefficients in binary format. The quantization parameter Δ also needs to be stored in the file. We fix $\theta = 1.3\Delta$ for all the images.

Decoding

- Recover the indices from their difference. This operation also gives the information about the number of coefficients in each block.
- Read the quantized unsigned coefficients from the string st_{cf} and transform them into real numbers as $|\tilde{c}_q^\Delta(n)| = \Delta \tilde{c}_q^\Delta(n) + (\theta - \Delta/2)$. Read the corresponding signs from the string st_{sg} .

- Recover the approximated partition, for each block, through the liner combination

$$\mathbf{I}_q^{r,k_q} = \sum_{n=1}^{k_q} \tilde{s}_q(n) |\tilde{c}_q^r(n)| \mathbf{d}_{\ell_n^x, q}^x (\mathbf{d}_{\ell_n^y, q}^y)^T.$$

- Assemble the recovered image as

$$\mathbf{I}^{r,K} = \hat{\mathcal{J}}_{q=1}^Q \mathbf{I}_q^{r,k_q},$$

where the $\hat{\mathcal{J}}$ indicates the operation for joining the blocks to restore the image.

Note: The MATLAB scripts for implementing the scheme and reproducing the results presented in the next section have been made available on [30].

3.1 Evaluation metrics

The quality of the approximation is quantified by the Mean Structural SIMilarity (MSSIM) index [31, 32] and the classical Peak Signal-to-Noise Ratio (PSNR), calculated as

$$\text{PSNR} = 10 \log_{10} \left(\frac{(2^8 - 1)^2}{\text{MSE}} \right), \quad \text{with} \quad \text{MSE} = \frac{\|\mathbf{I} - \mathbf{I}^{r,K}\|_F^2}{N_x N_y}. \quad (12)$$

The required MSSIM is set to be $\text{MSSIM} \geq 0.997$. This limit guarantees that the approximation is indistinguishable from the image, in the original size (it should be noticed that the MSSIM between an image and itself is 1). For the comparison with standard formats all the PSNRs are fixed as values for which JPEG and JPEG2 produce the required MSSIM. For producing a requested PSNR with the sparse representation approach we proceed as follows: the approximation routine is set to yield a slightly larger value of PSNR and the required one is then obtained by tuning the quantization parameter Δ .

As a measure of sparsity we use the Sparsity Ratio, which is defined as [33]:

$$\text{SR} = \frac{\text{Number of pixels in the image}}{\text{Number of coefficients in the representation}}. \quad (13)$$

Accordingly, the sparsity of a representation is manifested as a high value of SR.

In addition to the SR, which is a global measure of sparsity, a meaningful description of the variation of the image content throughout the partition is rendered by the local sparsity ratio, which is given as

$$\text{sr}(q) = \frac{N_b^2}{k_q}, \quad q = 1, \dots, Q, \quad (14)$$

where k_q is the number of coefficients in the decomposition of the q -block and N_b^2 is the number of pixels in the block.

3.2 Data sets

We illustrate the effectiveness of the proposed encoding scheme by storing the outputs of high quality sparse approximation of two data sets: 1) the Lukas 2D 8 bit medical image corpus, available on [34] and 2) the sample of 25 X-ray chest images taken randomly from the National Institute of Health (NIH) dataset available on [35].

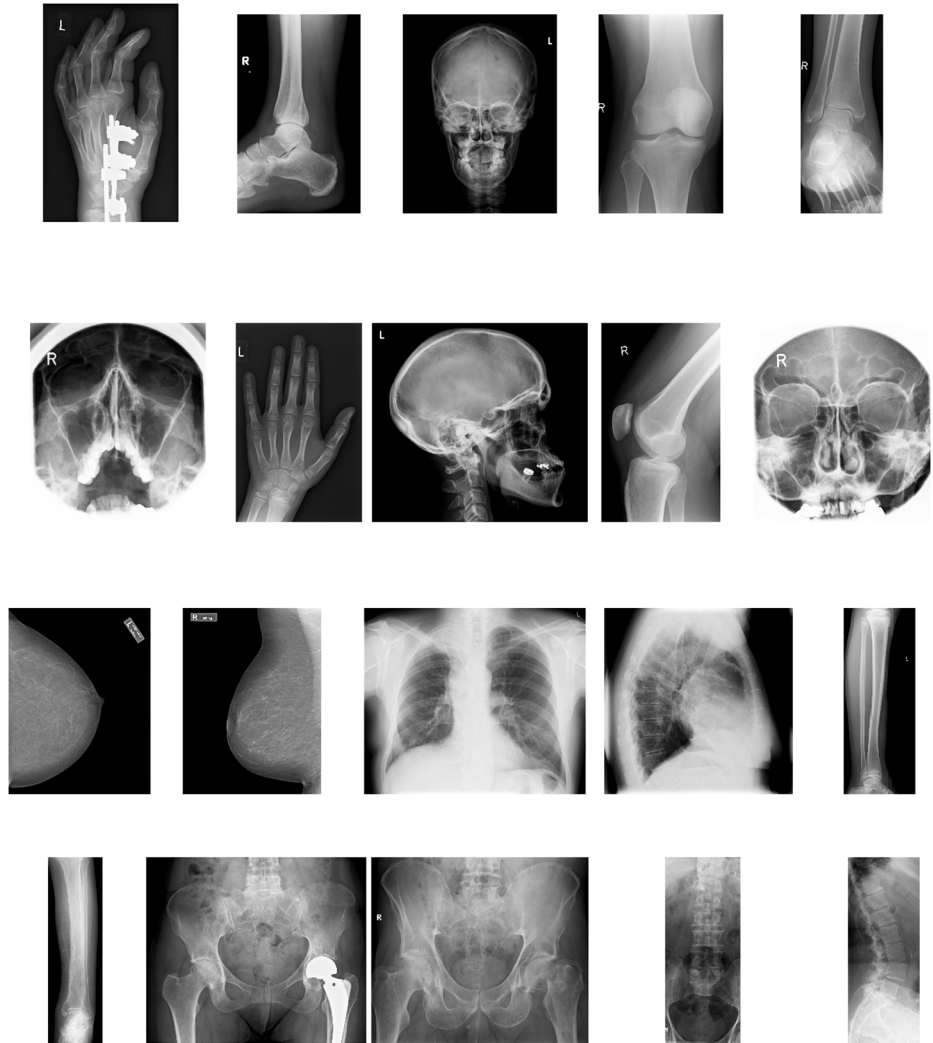


Fig 3. Lukas data set. The 20 images in the Lukas Corpus [34] listed in Table 1. First row: Hand₁, Foot₀, Head₀, Knee₁, Foot₁. Second row: Sinus₀, Hand₀, Head₁, Knee₀, Sinus₁. Third row: Breast₀, Breast₁, Thorax₀, Thorax₁, Leg₀. Fourth row: Leg₁, Pelvis₁, Pelvis₀, Spine₁, Spine₀. The average size of these images is 1943 × 1364 pixels.

<https://doi.org/10.1371/journal.pone.0201455.g003>

The Lukas corpus consists of 20 images which cover different parts of the human body, as shown in Fig 3. The average size of the images in the set is 1943 × 1364 pixels.

The chest X-Ray images are available in the raster graphics file format portable network graphics (png). Information related to the NIH X-Ray collection is given in [36]. The 25 images used here have been placed on [30]. The sample consists of common views of the chest in radiology. The posteroanterior, anteroposterior, and lateral views, which are obtained by changing the relative orientation of the body and the direction of the X-Ray beam. The images in this set are of similar size and smaller than in the Lukas corpus. The average size is 496 × 512 pixels.

4 Results

The approximation of all the images in the Lukas corpus are performed in both the pixel intensity and the wavelet domain. The size of the blocks in the image partition is fixed taking into

Table 1. Comparison of size rate (in bpp) for the Lukas corpus, listed in the first column. The third column shows the bpp values corresponding to the sparse representation in the pixel domain (S_{pd}). The fourth column shows the corresponding results in the wavelet domain (S_{wd}). The fifth and sixth columns are the bpp values for the formats JPEG and JPEG2, respectively. All the approaches render a MSSIM ≥ 0.997 and the values of PSNR listed in the second column.

Image	dB	S_{pd}	S_{wd}	JPEG	JPEG2
1 Hand ₁	48.1	0.443	0.286	0.436	0.234
2 Foot ₀	48.6	0.462	0.306	0.449	0.247
3 Head ₀	47.4	0.441	0.320	0.419	0.244
4 Knee ₁	48.0	0.488	0.316	0.485	0.286
5 Foot ₁	48.1	0.624	0.391	0.566	0.335
6 Sinus ₀	47.1	0.676	0.416	0.575	0.334
7 Hand ₀	48.8	0.612	0.424	0.586	0.371
8 Head ₁	46.4	0.599	0.452	0.546	0.346
9 Knee ₀	49.1	0.612	0.419	0.573	0.364
10 Sinus ₁	45.8	0.697	0.453	0.594	0.358
11 Breast ₀	44.3	0.519	0.465	0.605	0.393
12 Breast ₁	44.3	0.691	0.629	0.832	0.521
13 Thorax ₀	44.1	0.827	0.686	0.874	0.629
14 Thorax ₁	43.4	0.816	0.693	0.924	0.619
15 Leg ₀	48.9	1.000	0.867	1.152	0.759
16 Leg ₁	49.2	1.384	1.240	1.584	1.056
17 Pelvis ₁	44.3	1.596	1.418	1.828	1.248
18 Pelvis ₀	44.4	1.606	1.435	1.827	1.310
19 Spine ₁	47.0	2.131	1.922	2.463	1.630
20 Spine ₀	47.4	2.395	2.298	2.764	1.902
Mean Value	46.7	0.938	0.727	1.004	0.659
Std	1.9	0.573	0.521	0.707	0.500

<https://doi.org/10.1371/journal.pone.0201455.t001>

account previously reported results [8, 21], which indicate that 16×16 is a good trade-off between the resulting sparsity and the processing time. The information about the sizes of the corresponding files is given in bits per pixel (bpp) in the third and fourth columns of Table 1. The results for JPEG and JPEG2 are placed in the fifth and sixth columns, respectively. The last two rows of the table are the mean value of standard deviation (std) of the corresponding columns.

As shown in Table 1, all the files corresponding to approximations in the wavelet domain (S_{wd}) are smaller than those corresponding to approximations in the pixel intensity domain (S_{pd}), and also smaller than the JPEG ones. On average the files with the sparse representation in the wavelet domain are 22% smaller than those with the representation in the pixel domain, and 27% smaller than the JPEG files. In the present form JPEG2 produces the smallest files (on average 9% smaller than the files with the representation in the wavelet domain). However, both JPEG and JPEG2 formats involve an entropy coding step, which is not included in our scheme. Instead, the outputs of our algorithm are stored in HDF5 format [37, 38]. In the provided software [30] this is implemented in a straightforward manner using the MATLAB function `save`. On the contrary, adding an entropy coding step to the software would increase the processing time.

A very interesting feature of the numerical results is that the quantization process, intrinsic to the economic store of the coefficients in the image approximation does not reduce the sparsity. For the sake of comparison in addition to calculating the SRs obtained with the dictionary approach in both domains, before and after quantization, we have also calculated the

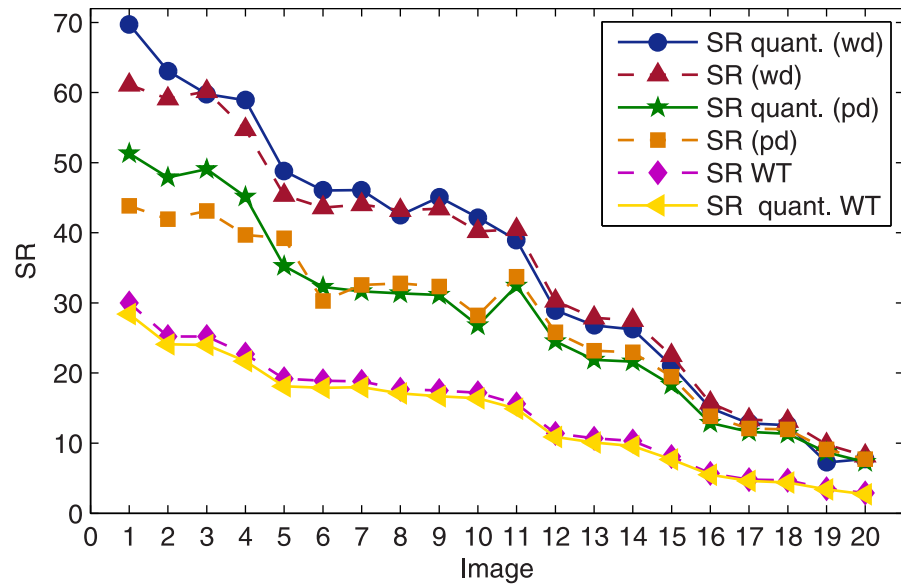


Fig 4. Comparison of the SRs, before and after quantization, corresponding to the dictionary approach in both the pixel intensity and the wavelet domain and to the wavelet approximation by nonlinear thresholding.

<https://doi.org/10.1371/journal.pone.0201455.g004>

corresponding SRs produced by nonlinear thresholding of the wavelet coefficients. The results are shown in Fig 4.

As already discussed, since the quantization of coefficients degrades quality, to achieve the required PSNR the approximation of the image has to be carried out up to a higher PSNR value. Nevertheless, because the quantization process maps some coefficients to zero, for the corpus of 20 images in this study, quantization does not affect sparsity. On the contrary, as can be observed in Fig 4, for the sparsest images (first 5 images in Table 1) sparsity actually benefits from quantization. It is also clear that for the images in the upper part of the table the SR in the wavelet domain is significantly larger than in the pixel intensity domain. However, the level of sparsity achieved by the dictionary approach is, in both domains, significantly higher than that achieved by nonlinear thresholding of the wavelet coefficients. If the dictionary approach operates in the wavelet domain, then after quantization the mean value gain in SR with respect to thresholding of the wavelet coefficients is 163%, with standard deviation of 17%. In the pixel intensity domain the corresponding gain is 113%, with standard deviation of 30%.

It is worth mentioning that the local sparsity ratio (c.f. (14)) can be used to produce a digital summary of the images. Indeed, each of the graphs of Fig 5 depicts the inverse of the local sparsity ratio in the pixel domain, for nine of the images in the Lukas corpus. Each of the points in a graph represents the number of coefficients in the atomic decomposition of each block in the image partition. Hence, the number of points in each of the graphs of Fig 5 is equal to the number of blocks in the corresponding image partition. Note that showing the inverse of the local sparsity ratio implies that the brightest pixels correspond to the least sparse blocks. By comparing the graphs in Fig 5 with the images in Fig 3 it is clear that each of the graphs in Fig 5 corresponds to one of the images in the Lukas corpus: Hand₀, Foot₁, Head₁, Sinus₁, Leg₀, Thorax₀, Breast₀, Pelvis₁ and Spine₁. This suggests that the digital summary of the images could be of assistance to classification and feature extraction techniques. For recent publications in that area of application see [39–45].

The approximation of the 25 chest X-Ray images is carried out to achieve a PSNR of 45dB for all the images. This guarantees a MSSIM of at least 0.99 for every image. The resulting size

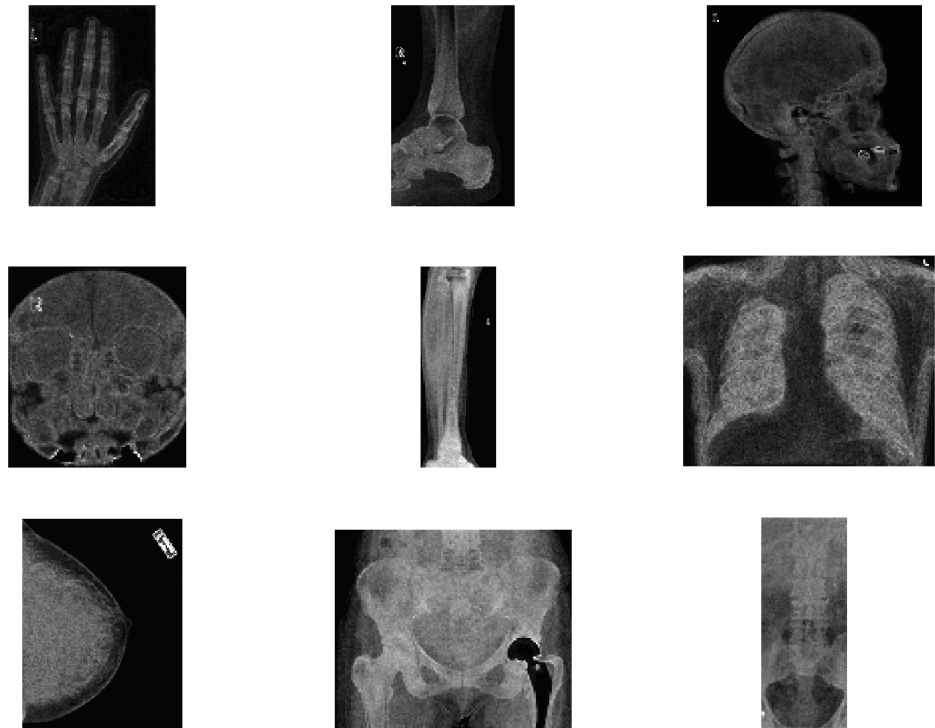


Fig 5. Digital summary of nine of the images in the Lukas corpus. The points in each graph are the inverse of the local sparsity ratio for nine of the images in Fig 3. The sizes of the graphs are (from top left to bottom right) 124×78 , 140×86 , 97×104 , 99×88 , 132×38 , 107×128 , 1138×105 , 100×118 and 131×53 pixels.

<https://doi.org/10.1371/journal.pone.0201455.g005>

of the files lead to the same remarks as the values in the previous example displayed in Table 1. Indeed, as shown in Fig 6, the size rate produced by the dictionary approach in the pixel intensity domain (S_{pd}) is competitive with the JPEG format for the same quality. The files produced in the wavelet domain (S_{wd}) are smaller than the JPEG files but larger than the JPEG2 ones. However, let's recall that the entropy coding step, which is part of the JPEG and JPEG2 compression standards, is not included in our scheme. This results in larger files but fast processing. Certainly, using MATLAB environment and a small notebook 2.9GHz dual core i7 3520M CPU and 4GB of memory, the average time for carrying out the approximation and creating the files of the 25 images with the dictionary approach is 1.7 s per image (with standard deviation 0.5). This time is the average of five independent runs in the time domain and another five independent runs in the wavelet domain.

In this case the sparsity of the image representation benefits from quantization even more than in the previous case: The dictionary approach in the pixel intensity domain yields a mean value SR (\overline{SR}) equal to 25.4 before quantization and $\overline{SR} = 36.7$ and after quantization. In the wavelet domain $\overline{SR} = 35.4$ before quantization, while $\overline{SR} = 52.0$ after quantization.

5 Conclusions

A scheme for encapsulating the sparse representation of an X-Ray image in a small file has been proposed. The approach operates within the over-complete dictionary framework, which achieves a notable level of sparsity in the image representation. An interesting feature, emerging from the numerical results illustrating the approach, is that the process of devising the small file does not affect the sparsity of the representation. We consider this a very important

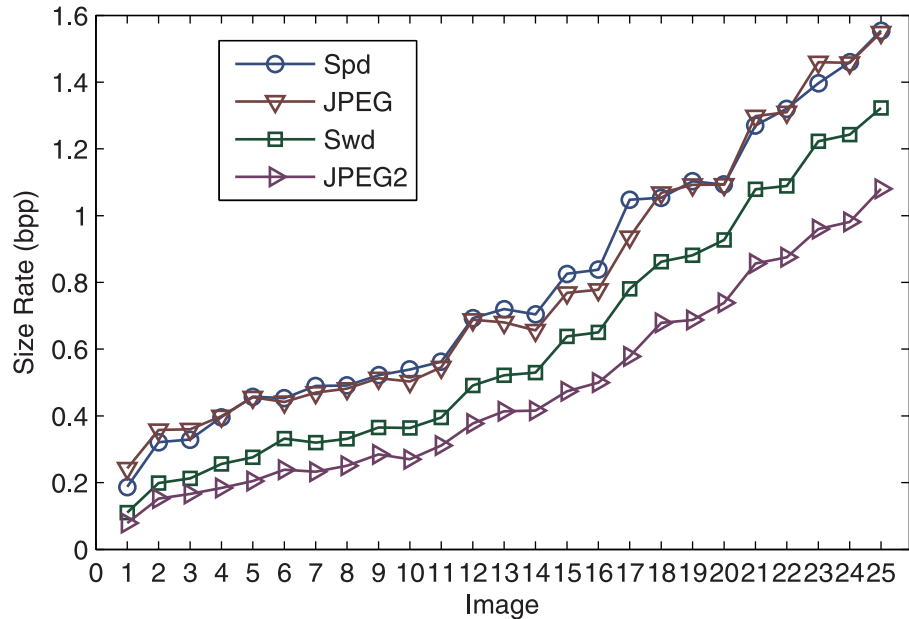


Fig 6. Comparison of the file size rate, in bpp, against the JPEG and JPEG2 formats for the 25 chest images. S_{pd} and S_{wd} correspond to the files containing the representation of the dictionary approach in the pixel and wavelet domains, respectively.

<https://doi.org/10.1371/journal.pone.0201455.g006>

outcome, because the central aim of the approach is to reduce, as much as possible, the cardinality of the data representing the informational content of the images. Additionally, the competitiveness of the file size was compared against the standard formats JPEG and JPEG2. The MATLAB routines, for implementing all the steps of the process to reproduce the results in Table 1 and Fig 6, have been made available on a dedicated website [30].

As a final remark it should be mentioned that the quality of the image representation in the present study is very high. A lower value of MSSIM might be acceptable as diagnostically lossless representation of X-Ray images [32]. Within the proposed scheme the only effects of setting a lower value of MSSIM are: a) the approximation routine runs faster, since less atoms are chosen, and b) the storage file is of course smaller. The right value of MSSIM is to be set according to how the file is to be used whilst taking into account specialized opinions.

Acknowledgments

Thanks are due to Prof. Rainer Köster and Dr. Jürgen Abel for making available the Lukas corpus of 8-bit 2D medical images [34].

Author Contributions

Conceptualization: Laura Rebollo-Neira.

Investigation: Laura Rebollo-Neira.

Methodology: Laura Rebollo-Neira.

Software: Laura Rebollo-Neira.

Validation: Laura Rebollo-Neira.

Writing – original draft: Laura Rebollo-Neira.

Writing – review & editing: Laura Rebollo-Neira.

References

1. J. Dheebaa, N. Albert Singhb, S. Tamil Selvi, “Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach”, *Journal of Biomedical Informatics*, 49, 45–52 (2014). <https://doi.org/10.1016/j.jbi.2014.01.010>
2. Liu X., Zheng Z., “A new automatic mass detection method for breast cancer with false positive reduction”, *Neurocomputing*, 152, 388–402 (2015). <https://doi.org/10.1016/j.neucom.2014.10.040>
3. Zhang X., Dou H., Ju T., Xu J., Zhang S., “Fusing Heterogeneous Features From Stacked Sparse Auto-encoder for Histopathological Image Analysis”, *IEEE Journal of Biomedical and Health Informatics*, 20, 1377–1383 (2016). <https://doi.org/10.1109/JBHI.2015.2461671> PMID: 26241980
4. Yoon J., Davtyan C., van der Schaar M, “Discovery and Clinical Decision Support for Personalized Healthcare”, *IEEE Journal of Biomedical and Health Informatics*, 1133–1145 (2016). <https://doi.org/10.1109/JBHI.2016.2574857> PMID: 27254875
5. Bar Y., Diamant I., Wolf L., Lieberman S., Konen E., Greenspan H., “Chest pathology detection using deep learning with non-medical training”, *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015.
6. Lakhani P., Sundaram B., “Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks”, *Radiology*, (2017). <https://doi.org/10.1148/radiol.2017162326> PMID: 28436741
7. Nithila E. E., Kumar S.S., “Automatic detection of solitary pulmonary nodules using swarm intelligence optimized neural networks on CT images”, *Engineering Science and Technology, an International Journal*, 20, 1192–1202 (2017). <https://doi.org/10.1016/j.jestch.2016.12.006>
8. Rebollo-Neira L., “Effective sparse representation of X-ray medical images” *International Journal for Numerical Methods in Biomedical Engineering*, (2017). <https://doi.org/10.1002/cnm.2886> PMID: 28388837
9. Pereira D. C., Ramos R. P., Zanchetta do Nascimento M., “Segmentation and detection of breast cancer in mammograms combining wavelet analysis and genetic algorithm”, *Computer Methods and Programs in Biomedicine*, 114, 88–101 (2014). <https://doi.org/10.1016/j.cmpb.2014.01.014> PMID: 24513228
10. Wang Z., Yu G., Kang Y., Zhao Y., Qu Q., “Breast tumor detection in digital mammography based on extreme learning machine”, *Neurocomputing*, 128, 175–184 (2014). <https://doi.org/10.1016/j.neucom.2013.05.053>
11. Liu B., Wang M., Foroosh H., Tappen M., Pensky M., “Sparse Convolutional Neural Networks” *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255, 2015 10.1109/CVPR.2015.7298681.
12. Summers R.M., “Deep Learning and Computer-Aided Diagnosis for Medical Image Processing: A Personal Perspective”, in *Deep Learning and Convolutional Neural Networks for Medical Image Computing*, 3–10 Eds: Lu L., Zheng Y., Carneiro G., Yang L, Springer International Publishing, 3–10, 2017.
13. Shen D., Wu G., and Suk H-II, “Deep Learning in Medical Image Analysis”, *Annual Review of Biomedical Engineering*, 19, 221–248 (2017). <https://doi.org/10.1146/annurev-bioeng-071516-044442> PMID: 28301734
14. Carneiro G., Zheng Y., F Xing. L Yang. “Review of Deep Learning Methods in Mammography, Cardiovascular, and Microscopy Image Analysis”, in *Deep Learning and Convolutional Neural Networks for Medical Image Computing*, 11–32, Eds: Lu L., Zheng Y., Carneiro G., Yang L, Springer International Publishing, 3–10, 2017.
15. Qayyum A., Anwar S.M., Awais M., Majid M., “Medical image retrieval using deep convolutional neural network”, *Neurocomputing*, 266, 8–20 (2017). <https://doi.org/10.1016/j.neucom.2017.05.025>
16. Chen S. S., Donoho D. L., Saunders M. A., “Atomic Decomposition by Basis Pursuit”, *SIAM Journal on Scientific Computing*, 20, 33–61 (1998). <https://doi.org/10.1137/S1064827596304010>
17. Elad M., *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer (2010).
18. Mallat S., Zhang Z., “Matching pursuit with time-frequency dictionaries.” *IEEE Transactions on Signal Processing*, Vol (41,12) 3397–3415 (1993). <https://doi.org/10.1109/78.258082>
19. Pati Y.C., Rezaiifar R., and P.S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” *Proc. of the 27th ACSSC*, 1, 40–44 (1993).
20. Rebollo-Neira L., Lowe D., “Optimized orthogonal matching pursuit approach”, *IEEE Signal Processing Letters*, 9, 137–140 (2002). <https://doi.org/10.1109/LSP.2002.1001652>

21. Rebollo-Neira L., Bowley J., "Sparse representation of astronomical images", *Journal of The Optical Society of America A*, 30, 758–768 (2013). <https://doi.org/10.1364/JOSAA.30.000758>
22. Rebollo-Neira L., Matiol R., Bibi S., "Hierarchized block wise image approximation by greedy pursuit strategies," *IEEE Signal Processing Letters*, 20, 1175–1178 (2013). <https://doi.org/10.1109/LSP.2013.2283510>
23. Kreutz-Delgado K., Murray J. F., Rao B. D., Engan K., Lee Te-Won, and Sejnowski T. J., "Dictionary Learning Algorithms for Sparse Representation", *Neurocomputing*, 15, 349–396 (2003).
24. Aharon M., Elad M., and Bruckstein A., "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", *IEEE Transactions on Signal Processing*, 54, 4311–4322 (2006). <https://doi.org/10.1109/TSP.2006.881199>
25. Rubinstein R., Zibulevsky M., and Elad M., "Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation", *IEEE Transactions on Signal Processing*, 58, 1553–1564 (2010). <https://doi.org/10.1109/TSP.2009.2036477>
26. Tošić I. and Frossard P., "Dictionary Learning: What is the right representation for my signal?", *IEEE Signal Processing Magazine*, 28, 27–38 (2011). <https://doi.org/10.1109/MSP.2010.939537>
27. Zepeda J., Guillemot C., Kijak E., "Image Compression Using Sparse Representations and the Iteration-Tuned and Aligned Dictionary", *IEEE Journal of Selected Topics in Signal Processing*, 5, 1061–1073 (2011). <https://doi.org/10.1109/JSTSP.2011.2135332>
28. Srinivas M., Naidu R.R., Sastry C.S., Krishna Mohana C., "Content based medical image retrieval using dictionary learning", *Neurocomputing*, 168, 880–895 (2015). <https://doi.org/10.1016/j.neucom.2015.05.036>
29. Rebollo-Neira L., Bowley J., Constantinides A. and Plastino A., "Self contained encrypted image folding", *Physica A*, 391, 5858–5870 (2012). <https://doi.org/10.1016/j.physa.2012.06.042>
30. <http://www.nonlinear-approx.info/examples/node08.html>
31. Wang Z., Bovik A. C., Sheikh H.R., Simoncelli E. P., "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, 13, 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861> PMID: 15376593
32. Kowalik-Urbaniak I., Brunet D., Wang J., Koff D., Smolarski-Koff N., Vrscaj E., Wallace B., Wang Z., "The quest for 'diagnostically lossless' medical image compression: a comparative study of objective quality metrics for compressed medical images", *Proc. SPIE 9037, Medical Imaging 2014: Image Perception, Observer Performance, and Technology Assessment*, 903717 (March 11, 2014);
33. Bowley J., Rebollo-Neira L., Sparsity and something else: an approach to encrypted image folding, *IEEE Signal Processing Letters*, 8, 189–192 (2011). <https://doi.org/10.1109/LSP.2011.2106496>
34. <http://www.data-compression.info/Corpora/LukasCorpus/>
35. https://openi.nlm.nih.gov/imgs/collections/NLMCXR_png.tgz
36. Wang X., Peng Y., Lu L., Lu Z., Bagheri M., Summers R.M., "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases", *IEEE CVPR 2017*, http://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_ChestX-ray8_Hospital-Scale_Chest_CVPR_2017_paper.pdf
37. <https://www.hdfgroup.org/>
38. L. A. Wasser, "Hierarchical Data Formats—What is HDF5?", <http://neondatakills.org/HDF5/About> (2015).
39. Lin C., Chen W., Qiu C., Wu Y., Krishnan S., Q Zou. "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy", *Neurocomputing*, 123, 424–435 (2014). <https://doi.org/10.1016/j.neucom.2013.08.004>
40. Zou Q., Zeng J., Cao L., Ji R., "A novel features ranking metric with application to scalable visual and bio-informatics data classification" *Neurocomputing*, 173, 346–354 (2016). <https://doi.org/10.1016/j.neucom.2014.12.123>
41. Zhu P., Zuo W., Zhang L., Hu Q., Shiu S.C.K., "Unsupervised feature selection by regularized self-representation", *Pattern Recognition*, 48, 438–446 (2015). <https://doi.org/10.1016/j.patcog.2014.08.006>
42. Zhu P., Zhu W., Wang W., Zuo W., Hu Q., "Non-convex regularized self-representation for unsupervised feature selection", *Image and Vision Computing*, 60, 22–29 (2017). <https://doi.org/10.1016/j.imavis.2016.11.014>
43. Zhu P., Hu Q., Han Y., Zhang C., Du Y., "Combining neighborhood separable subspaces for classification via sparsity regularized optimization", *Information Sciences*, 370, 270–287 (2016). <https://doi.org/10.1016/j.ins.2016.08.004>

44. Zhu P., Zhu W., Hu Q., Zhang C., Zuo W., "Subspace clustering guided unsupervised feature selection", *Pattern Recognition*, 60, 364–374 (2017). <https://doi.org/10.1016/j.patcog.2017.01.016>
45. Zhu P., Xu Q., Hu Q., Zhang C., Zhao H., "Multi-label feature selection with missing labels", *Pattern Recognition*, 74, 488–502 (2018). <https://doi.org/10.1016/j.patcog.2017.09.036>