Denoising of Dual Energy X-ray Absorptiometry Images and Vertebra Segmentation

NICOLAS ROUSSEL
Denoising of Dual Energy X-ray Absorptiometry Images and Vertebra Segmentation

NICOLAS ROUSSEL

Master in Electrical Engineering
Date: August 25, 2018
Supervisor: Josephine Sullivan
Examiner: Mårten Björkman
School of Electrical Engineering and Computer Science
Abstract

Dual Energy X-ray Absorptiometry (DXA) is a medical imaging modality used to quantify bone mineral density and to detect fractures. It is widely used due to its cheap cost and low radiation dose, however it produces noisy images that can be difficult to interpret for a human expert or a machine.

In this study, we investigate denoising on DXA lateral spine images and automatic vertebra segmentation in the resulting images. For denoising, we design adaptive filters to avoid the frequent apparition of edge artifacts (cross contamination), and validate our results with an observer experiment. Segmentation is performed using deep convolutional neural networks trained on manually segmented DXA images. Using few training images, we focus on depth of the network and on the amount of training data. At the best depth, we report a 94% mean Dice on test images, with no post-processing. We also investigate the application of a network trained on one of our databases to the other (different resolution). We show that in some cases, cross contamination can degrade the segmentation results and that the use of our adaptive filters helps solving this problem. Our results reveal that even with little data and a short training, neural networks produce accurate segmentations. This suggests they could be used for fracture classification. However, the results should be validated on bigger databases with more fracture cases and other pathologies.
Sammanfattning

Dual Energy X-ray Absorptiometry (DXA) är en medicinsk bildbehandlingsmodalitet som används för att kvantifiera bentäthet och uppställa frakturer. Det används i stor utsträckning tack vare sin låga kostnad och sin låga exponering, men producerar brusiga bilder som kan vara svåra att förstå för en mänsklig expert eller en maskin.

Acknowledgements

First, I would like to thank my supervisor at General Electric, Pablo Milioni de Carvalho, for his daily guidance and his very valuable help towards the completion of my work. I am also grateful to Serge Muller for his global supervision, and to Zhijin Li and Giovanni Palma for their punctual intervention on this project. I wish to express my gratitude to Josephine Sullivan, my supervisor at KTH, who helped me a lot for the writing of this document. Finally, I thank every of my co-workers at GE for their good mood and for making our office a great place to work in.
Contents

Introduction ................................................................. 1
  Dual Energy X-Ray Absorptiometry ............................... 1
  Objectives of this thesis ............................................. 4
  Societal aspect and ethical considerations ....................... 6

1 Image quality analysis in DXA ........................................ 9
  1.1 DXA image processing chain .................................... 9
  1.2 Image artifacts .................................................... 15
    1.2.1 DXA datasets ................................................. 15
    1.2.2 List of artifacts ............................................. 16
    1.2.3 Ranking of image artifacts and perspectives of
          improvement .................................................. 19

2 Adaptive filtering for cross contamination reduction .......... 21
  2.1 Review of denoising methods in medical imaging ............ 21
    2.1.1 Denoising methods for CT and DXA ......................... 21
    2.1.2 Filters in image processing .................................. 24
    2.1.3 Illustration of filters on DXA images ....................... 27
    2.1.4 Reference denoising algorithm in enCORE software ...... 29
  2.2 Methods for Anti-Correlated Noise Reduction ............... 30
    2.2.1 Metrics ....................................................... 31
    2.2.2 Adaptive bilateral filtering .................................. 34
    2.2.3 Anisotropic diffusion ....................................... 37
    2.2.4 Weighting of the tissue noise component ................. 40
    2.2.5 Mode of evaluation ......................................... 41
  2.3 Results ............................................................ 43
    2.3.1 Observer experiment ......................................... 43
    2.3.2 Qualitative results ......................................... 44
  2.4 Discussion ........................................................ 46
3 Vertebra segmentation in DXA images

3.1 Literature review

3.1.1 Common segmentation methods

3.1.2 Other work on vertebrae segmentation or localization

3.1.3 Segmentation in enCORE software

3.1.4 Convolutional Neural Networks for classification

3.1.5 Fully Convolutional Networks for segmentation (FCN)

3.2 Classification and quality metrics

3.2.1 Cross-entropy

3.2.2 Dice coefficient

3.2.3 Other common metrics

3.3 Data acquisition

3.3.1 Ground truth annotation

3.3.2 Data preprocessing

3.3.3 Data augmentation

3.4 Methods

3.4.1 Network architecture

3.4.2 Training specifics

3.4.3 Test methodology

3.5 Results

3.5.1 Performance evolution with respect to depth

3.5.2 Performance evolution with respect to data

3.5.3 Example results on iDXA

3.5.4 Generalization to the Prodigy database

3.5.5 Influence of cross contamination on segmentation

3.5.6 Comparison with the current segmentation system

3.6 Discussion

Conclusions

Bibliography

Appendix
Introduction

Dual Energy X-Ray Absorptiometry

Presentation

Dual Energy X-Ray Absorptiometry (DXA, or sometimes DEXA) is a medical imaging modality used to assess the body composition of a patient. It uses the fact that a given material has a unique attenuation coefficient at a specific level of energy. In a DXA system, a X-ray source emits a radiation. The X-ray beam first goes through a filter which splits the beam into low energy and high energy photons. It then passes through the body to be scanned, and the number and energy of transmitted photons are stored.

Figure 1: Schematic representation of a DXA table. The C-arm holds the detector above the patient.
This allows to build two images, a Low Energy and a High energy, which quantify the amount of photons received by the photo-detectors per unit of time. Assuming the body only contains two materials, bone and soft tissue, whose attenuation coefficients are known, one can recover the density of each material for every image pixel corresponding to a photo-detector element. We can finally obtain two quantitative images, namely the bone and soft tissue images, giving the areal density of materials.

Goals of DXA measurements

A DXA scan can generally serve two purposes. First, the bone image is used to measure the Bone Mineral Density. This examen is done to assess osteoporosis, a disease characterized by low bone mass. Because it is generally asymptomatic, osteoporosis is often undervalued. Yet, it is a serious disease and it must be detected to evaluate risks of subsequent fractures caused by bone fragility. Indeed, it is estimated that osteoporosis causes approximately 8.9 million fractures each year [1]. It touches women more frequently than men, with roughly one third of women over age 60 being affected [2]. For this reason, it is common for women of that age to undergo a preventive DXA examen. Among the possible body sites, BMD is often measured on the hip or on the lower spine because these areas are easy to scan and provide reliable results. They give good estimations of fracture risks: it is estimated that a 10 % loss of bone mass for a vertebra doubles the risk of vertebral fracture, and the same loss of bone mass for the hip multiplies the risk by 2.5 [3].

The second main objective of the DXA examen is the direct visualization of fractures on the bone image. Fracture assessment is usually made using quantitative or semi-quantitative methods, based on geometric considerations and by visual inspection by the radiologist. In this thesis, we will focus only on lateral images of the spine, although most methods described can be generalized to the whole body. For lateral spine images, a common method for fracture evaluation is to use Genant’s semi-quantitative method [4]. Vertebral height is measured on three points: anterior position, posterior position, and middle point. Based on ratios of these three heights, Genant classifies vertebral fractures in three types (wedge, biconcave and crush), and gives
a score for severity of the fracture (mild, moderate and severe). This is summarized in Figure 2.

![Genant’s classification of vertebral fractures, by type and degree.](image)

**Physics of DXA**

The X-ray beam’s intensity after going through a thickness $t$ of a material with attenuation coefficient $\mu$ is given by:

$$I = I_0 e^{-\mu t}$$  \hspace{1cm} (1)

where $I_0$ is the initial X-ray intensity (before attenuation). The coefficient $\mu$ depends on the X-ray’s energy: as the energy increases, the radiation gets more penetrating and $\mu$ decreases.

The attenuation results from interaction of photon with atoms of the material, with three mechanisms: coherent scattering, Compton scattering and photoelectric absorption. We can rewrite this formula and break down the attenuation according to these mechanisms, by introducing the volumic mass density $\rho$, the areal mass density $\sigma$:

$$I(t) = I_0 e^{-\frac{\mu}{\rho} t \rho} = I_0 e^{-\frac{\mu}{\sigma} \sigma}$$  \hspace{1cm} (2)

with:

$$\frac{\mu}{\rho} = \left(\frac{\mu}{\rho}\right)_{\text{Photoelectric}} + \left(\frac{\mu}{\rho}\right)_{\text{Compton}} + \left(\frac{\mu}{\rho}\right)_{\text{Coherent}}$$  \hspace{1cm} (3)
4 INTRODUCTION

For the energy range of DXA scans (70 - 100 kVp), it was shown that the attenuation due to coherent scattering is negligible. This is used by Alvarez in [5] to decompose any attenuation coefficient into two terms (photoelectric and Compton effects). The fundamental assumption for a DXA system is that we use a three materials model for the body composition: bone, lean soft tissue and fat soft tissue. The two energy levels allow us to solve for only two materials, so in areas where bone is present we simplify this by considering two materials, bone and soft tissue, and outside bone areas we consider only lean soft tissue and fat soft tissue. In presence of bones, and for monochromatic X-rays (to simplify), the equations of attenuation can be written:

\[ I_L = I_L^0 e^{-[(\rho \mu)_{ST,L} \sigma_{ST} + (\rho \mu)_{B,L} \sigma_{B}]} \]
\[ I_H = I_H^0 e^{-[(\rho \mu)_{ST,H} \sigma_{ST} + (\rho \mu)_{B,H} \sigma_{B}]} \]

in which the subscripts \( L, H, ST \) and \( B \) respectively denote low energy, high energy, soft tissue and bone. By solving this system of equations we obtain the bone areal density \( \sigma_B \) and the soft tissue areal density \( \sigma_{ST} \).

Comparison with other imaging techniques

DXA is not the only imaging system able to generate accurate bone and tissue images for qualitative and quantitative evaluation. Common other techniques are Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). Although both CT and MRI usually give better resolution images, DXA is still widely used because it is less expensive than MRI and uses lower dose than CT. The downside of using a lower dose is that it results in very noisy images, which must be heavily preprocessed before they can be used.

Objectives of this thesis

In this document, we investigate two main tasks in relation to DXA imaging. The first one is the improvement of image quality by the means of an appropriate denoising algorithm. For a low dose modality like DXA, noise removal can be especially challenging but is a necessary step towards acquiring images that can be interpreted by a ra-
diologist. Hence, any subsequent analysis such as vertebral fracture assessment depends heavily on it. Our second task is the automatic segmentation of vertebrae in lateral reconstructed bone images. This segmentation is useful for several reasons. For instance, Bone Mineral Density is measured over regions of interest which are usually the lumbar vertebrae, so better defined regions can potentially mean more accurate densities. But mostly, automatic segmentation plays an important role in helping the radiologist establish a vertebra fracture diagnosis. The segmentation boundaries could be used to accurately locate landmark points for fracture classification, or to define new curvature-based metrics on the shape outline. However, since our database did not include enough cases of fracture, we restrict ourselves to the task of segmentation and insist that we do not develop a framework of automatic fracture classification.

Regarding the structure of the thesis, we present our work on these tasks in a natural order, with first the denoising as a preprocessing for the segmentation which comes afterwards. This document consists of three chapters. Chapter 1 gives an overview of our analysis of image quality. We start with a description of each step in the processing chain from raw photon counts to denoised and reconstructed bone images. We then review the artifacts encountered in our database and rank them by importance, in relation to degradation of quality. The conclusion of this chapter is that we have identified one artifact among others, namely cross contamination, that we will study in depth. In Chapter 2, we tackle the denoising task. After a literature review on denoising and filtering in medical image processing, we develop methods for noise reduction with a special focus on limitation of cross contamination artifacts. The question raised here is to determine whether we can improve on the current system by choosing the right set of filters, and reduce artifacts while preserving good noise levels, texture and resolution. Our contribution is a new strategy for adapting parameters of bilateral filter and anisotropic diffusion, within the Anti-Correlated Noise Reduction framework. We assess our method by an observer experiment which compares our denoising results to the images from the current system’s software. Finally, Chapter 3 is our study of vertebra segmentation in lateral spine images. We investigate this problem with a deep learning-based approach, using fully convolutional neural networks trained end-to-end on our images. Although CNNs are
a common research topic nowadays, very little work is dedicated to their application to DXA. Because of lower dose and contrast than in other imaging modalities, it is difficult to predict how well such systems will perform on our images. Moreover, deep learning usually requires thousands of labeled training data, whereas we had very few images available.

**Societal aspect and ethical considerations**

Development of AI within the medical field entails societal consequences of importance. It is commonly accepted that many diagnosis errors attributed to a human practitioner could be avoided by using a machine assistant, in the form of an AI that helps taking the right decision. Automatic image analysis for medical diagnosis has quickly become a field of interest for the machine learning research community, as deep learning demonstrates excellent results for visual recognition tasks. Human analysis of a DXA scan can be a long and tedious process, which could greatly benefit from using an AI that performs the task in less than one second. Combining the strengths of the human reader (reliability in diagnosis) and of the machine (processing speed) would overall increase the diagnosis efficiency and accuracy. This would result in less errors due to misinterpretation of the results or negligence by lack of time. For automatic fracture assessment on DXA images, this means increasing the detection rate of osteoporotic fractures, which are often warning signs for more subsequent fractures or serious accidents for the elderly. Machine learning systems applied to bone and spine imaging are not a new topic. However, most work published deals with MRI or tomography, and few publications are dedicated to DXA, although it remains a broadly used imaging modality thanks to its cost and radiation dose. Whether we can transfer results of machine systems from MRI or CT to DXA is very much debatable, due to the important differences in image characteristics (contrast, visible organs, noise and artifacts). For this reason, conducting a specific study on DXA is important to evaluate the predicting abilities of a machine learning system dedicated to it.

Although artificial intelligence shows promising results on medical imaging diagnosis tasks, one of its current limitation is the difficulty
in data collection. In many areas, gathering data for reuse can be a sensitive topic, and this is especially true for the medical field where it involves legal but also ethical questions. Hence, it is subject to regulation and as a result the constitution of very large databases usually needed to apply deep learning algorithms is made difficult. Using a patient’s scan for purposes of scientific research or software development requires the patient’s prior approval, and the condition that the data be anonymized. This implies that the radiologist explicitly asks the permission to use the data during the exam. Hence, the vast majority of data theoretically available cannot actually be used because there lacks the appropriate authorization. Public databases accessible with restricted access exist nowadays, but most of them are still of a rather small size. Moreover, the lack of standardization from one database to another (due to hardware scanner differences or acquisition specifics) makes it hard to constitute a global base by aggregating data of different origin. In the era of big data, there remains fields like medical imaging for which little data is available and safely guarded. This makes it especially important to question the application of machine learning with little training data, which is one of the scopes of this thesis. Our experiment used 108 DXA scans from two different measurement devices. All data was anonymized beforehand, with the approval of the patient.

Another important ethical question is the introduction of undesired biases by algorithms. When relying on AI systems to help doctors making a treatment decision, it is necessary to understand where the system’s decision comes from and how it was made, to be sure that the patient will get the appropriate outcome. The problem is that nowadays, the most effective AIs give a decision out of a black-box model. This is the case of neural networks, in which we throw inputs and get back predictions that are hard to explain. If we cannot relate the decision to a set of input variables of significant importance, how can we be sure the system does not involve a learning bias which compromises its accordance to medical ethics? For instance, an algorithm that outputs the choice of a treatment according to several data (including social data, insurance, pay, etc.) could make a biased decision based on the patient’s wealth or racial origin if these variables are the most influential from the statistical point of view adopted by the machine. Especially, we can imagine that the algorithm is optimized with re-
spect to a given criterion (say, cost of treatment) which would not be the ethical criterion used by the practitioner. For these reasons, medical assistance from AIs needs to be controlled: which variables are we giving it access? How to ensure it does not introduce detrimental biases? In practice, this is hard to achieve, because machines tend to use all the data available if they make sense for the quantity they seek to optimize, even when we as humans would judge it unethical. Such an issue did not arise in our case: our machine learning application is restricted to the task of vertebra segmentation, which is why the only data we use are bone images acquired during DXA scans (and no other information on the patient). However, note that when designing a fracture classifier, one would have to be cautious about its application on patients outside the training database: vertebrae exhibit statistical differences in morphometry (size and aspect ratio) from one population to another, due for instance to age and ethnic origin. Although subtle, these implicit differences hidden in the images would probably need to be accounted for if we want the classifier to be as accurate as possible - which is why data always has to be compared to the correct reference population.
Chapter 1

Image quality analysis in DXA

In this section, we investigate the problem of DXA image quality. Such a wide topic entails different and independent questions, because global quality of an image results from various factors. As stated before, noise is an important problem in DXA imaging, and it will be at the center of this study. However, there are other interesting issues appearing in the processing chain of DXA images: artifacts of different nature (some of which are linked to the type of denoising technique employed), registration of multiple scans to reconstruct a global image, etc. An in-depth study of all the components of the DXA image processing chain is out of the scope of this thesis, so we will focus on the most important problem. However, before we do so, we will first briefly describe the processing chain to understand how the different images are generated and where artifacts arise. Once done, we will tackle in more details the question of denoising. We review a list of general image filtering algorithms that can be used for this purpose, and some more specific methods that can be found in medical imaging literature. Based on this, we select a few methods that we implement and test on our database. We finally compare our results with those from the current system.

1.1 DXA image processing chain

As explained in the general introduction, a DXA scan uses a beam of photons split around two levels of energy. Because attenuation level through the body is different for the two energies, measuring the photon counts per second after attenuation leads to two images: a low
energy count \((I_L)\) and a high energy count \((I_H)\) image. Before we can observe the bone structures as well as the tissue for Bone Mineral Density (BMD) measurement and fracture assessment, these images must undergo a series of transformation. Figure 1.1 gives an overview of the different steps.

**Figure 1.1: Image processing chain of the DXA system**

**Logarithm attenuation**

Because of the nature of the attenuation phenomenon, the Count Per Second images \(I_L\) and \(I_H\) defined in equations 4 and 5 depend exponentially on the bone and soft tissue densities, \(\sigma_B\) and \(\sigma_{ST}\). We linearize the system by taking the logarithm, and define the \(L_E\) and \(H_E\) images as:

\[
L_E = \log\left(\frac{I^0_L}{I_L}\right) \quad (1.1)
\]

\[
H_E = \log\left(\frac{I^0_H}{I_H}\right) \quad (1.2)
\]

in which the values with superscript zero stand for values before attenuation. The values after attenuation are always smaller than before attenuation, so the logarithms are positives.
Signal corrections

A series of corrections is applied to the $L_E$ and $H_E$ images. This is mostly to account for hardware issues: dead pixels caused by broken detectors, offset correction to compensate difference of calibration between detectors, pileup correction (when two low energy photons who hit a detector between two read-outs are mistaken for a single photon of high energy).

Basis material decomposition

The next step is called basis material decomposition and transforms low and high energy attenuations into areal density of two basis materials. The same way we can represent attenuation of any material as a weighted sum of attenuation coefficients for two interactions (photoelectric and Compton effects), we can also represent it as weighted sum of attenuations of two known materials, $M_1$ and $M_2$. Because the low and high energy x-rays are polychromatic in reality, the attenuation equations can be rewritten in an integral form, with a dependence in the attenuation of $M_1$ and $M_2$:

\[
I_L = \int_{E_{L,\text{min}}}^{E_{L,\text{max}}} I_0^L(E) \cdot \exp\left(-\left(\frac{\mu}{\rho}\right)_{M_1}(E) \cdot \sigma_{M_1} - \left(\frac{\mu}{\rho}\right)_{M_2}(E) \cdot \sigma_{M_2}\right) dE \quad (1.3)
\]

\[
I_H = \int_{E_{H,\text{min}}}^{E_{H,\text{max}}} I_0^H(E) \cdot \exp\left(-\left(\frac{\mu}{\rho}\right)_{M_1}(E) \cdot \sigma_{M_1} - \left(\frac{\mu}{\rho}\right)_{M_2}(E) \cdot \sigma_{M_2}\right) dE \quad (1.4)
\]

We want to invert this system and solve for $\sigma_{M_1}$ and $\sigma_{M_2}$, the areal densities for basis material 1 and 2. However, unlike in the monochromatic case, the system is not linear and does not have an analytic expression. Instead, we compute a numerical solution which takes the form of a polynomial of the two variables $L_E$ and $H_E$:

\[
\sigma_{M_1} = a_0 + a_1 L_E + a_2 H_E + a_3 L_E^2 + a_4 L_E H_E + a_5 H_E^2 + ... \quad (1.5)
\]

\[
\sigma_{M_2} = b_0 + b_1 L_E + b_2 H_E + b_3 L_E^2 + b_4 L_E H_E + b_5 H_E^2 + ... \quad (1.6)
\]
The coefficients $a$ and $b$ are determined by calibration. They depend on the energy spectrum and the chosen basis materials.

Without basis material decomposition, we could not always distinguish materials based on their attenuation. Indeed, one material can have a small atomic number and a high density and another one can have a high atomic number and a small density, resulting in a low attenuation for both of them. Thanks to the decomposition on known materials, we can recover the areal density of $M_1$ and $M_2$ and thus the density of bone and soft tissue by knowledge of their respective decomposition on this basis. The basis materials could in theory be any two materials, but in practice we take them to have similar attenuation properties as bone and soft tissue. We will now refer to these materials as Pseudo Bone Mineral ($PBM$) for $M_1$ and Pseudo Soft Tissue ($PST$) for $M_2$. Figure 1.2 shows an example of low and high (log) energy images and their corresponding basis material decomposition. Notice that $L_E$ and $H_E$ images have both bone and tissue elements, whereas the decomposed images have either bones ($PBM$) or organs ($PST$).

![Figure 1.2: Top: Low Energy ($L_E$) and High Energy ($H_E$). Bottom: Pseudo Bone Mineral (PBM) and Pseudo Soft Tissue (PST).](image-url)
Denoising

Once we get the images of basis materials density, we can filter them to remove noise. This filtering step could be applied before, on the low and high energy images, but it is often performed on the \( PBM \) and \( PST \) images to take advantage of the properties of noise on these images. It was shown by Kalender et al. [6] that the noise components of the \( PBM \) and \( PST \) images are anti-correlated. This property allows to estimate the noise on one of the basis images (\( PST \)) and add it to the complementary image (\( PBM \)) to denoise it. This algorithm called Anti-Correlated Noise Reduction will be described and studied in more details later on.

Registration

Because the whole body cannot be scanned at once, the photo-detectors are attached to a mechanical arm which moves over the patient to scan the whole body or area of interest. It follows a given path, as shown in the example figure 1.3.

Figure 1.3: Schematic view of the path of the detector over the scanned region

We obtain several overlapping sweeps, which are partial views of the final image. The registration process is the alignment of these sweeps to form a single, non overlapping image.
Bone Mineral and Soft Tissue densities

The last step is to compute the quantitative values for Bone Mineral Density and Soft Tissue Density. Both of them can be expressed as a linear combination of densities for the basis materials $PBM$ and $PST$, so this last transformation is a simple linear system:

$$\begin{bmatrix}
\text{bone} \\
\text{tissue}
\end{bmatrix} =\begin{bmatrix}
\alpha_1 & \alpha_2 \\
\beta_1 & \beta_2
\end{bmatrix}
\begin{bmatrix}
PBM \\
PST
\end{bmatrix}$$

(1.7)

where the $\alpha$ and the $\beta$ are the decomposition coefficients of bone and tissue respectively.
1.2 Image artifacts

In this section, we make a review of image artifacts encountered in our DXA database. In section 1.1, we have described the different components of the DXA processing chain. This will hopefully help us identifying the underlying cause of each artifact and give us clues on where to act on the image chain to improve the overall quality. Based on the observation of our DXA images, we have classified the main artifacts into four categories, that we will describe, explain and illustrate.

1.2.1 DXA datasets

Our DXA database contains images from two different datasets. They correspond to two measurement systems of GE, Prodigy and iDXA. Due to hardware specifics, these datasets exhibit major differences in terms of pixel resolution, as well as in contrast, noise and artifacts. We sum up some useful characteristics of the data in table 1.1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Acquisition system</th>
<th>Resolution (mm²)</th>
<th>Number of detectors</th>
<th>Number of images</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>iDXA</td>
<td>0.8×2.8</td>
<td>64</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>Prodigy</td>
<td>3×7</td>
<td>16</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 1.1: Dataset characteristics.

As an illustration of these characteristics, figure 1.4 shows an image from the iDXA device compared to an image of the same size from Prodigy. Note that the Prodigy image appears blurry, whereas in the iDXA image the outline of vertebrae and of the spine is well defined.

![Figure 1.4: Left: iDXA image, right: Prodigy image. Prodigy has lower resolution and appears blurry.](image-url)
1.2.2 List of artifacts

cross contamination

We have seen in 1.1 that denoising the basis material images \( PBM \) and \( PST \) uses the property of anti-correlation between their noises. This Anti-Correlated Noise Reduction algorithm (ACNR) assumes that we can extract the noise from \( PST \) to add it to \( PBM \) so that noises cancel out each other. However, in reality, the noise is extracted as a high-pass filtered version of \( PST \), which can be imperfect and thus also retain some high frequency signal such as edges or texture variations. This means the image \( PST_{\text{noise}} \) added to \( PBM \) to perform ACNR is likely to introduce parasite structures into \( PBM \), following the edges of the complementary image \( PST \). In practice, this results in dark lines corrupting the bone structures on the denoised \( PBM \). Because it is due to a transfer of structures from one image to another, this problem is referred to as cross contamination in [7], in which the author studies the denoising of computed tomography images. While we have not seen this term used in the DXA literature nor any mention of this issue, we also refer to it as cross contamination from now on. Note that by adding less of \( PST_{\text{noise}} \) to \( PBM \), one can limit the effect of cross contamination. Of course, this is at the cost of more noise in the final image, hence the ACNR method inherently suffers from the trade-off between noise reduction and cross contamination artifacts.

Misalignment of detectors columns

For the iDXA system, photo-detectors are positioned on a 32-by-2 array, with the second column being translated of half a detector’s length from the first one (see figure 1.5). This allows to double the resolution while keeping the same size of detectors. As a result, each sweep creates a sub-image with 64 rows by interpolating the second column’s pixels back onto the grid of the first column. However, although the detectors are fixed and the distance between the two columns remains the same at any time, the amount of translation needed to correct the shift depends on the object depth in the image. Thus, one can not simply translate of a given value all the pixels acquired with detectors of the second column. When the registration of the second column’s pixels onto the first is not perfect, this results in horizontal patterns as shown in figure 1.6. By taking only one out of two lines (either pix-
els from column 1 or pixels from column 2), we observe that the horizontal pattern disappears, which proves the origin is indeed in the mis-registration of the two columns of detectors (illustrated on a high-energy photon counts per second image, figure 1.6). Note that images acquired with the Prodigy system do not suffer from this artifact, as they use a single-column array of detectors.

Figure 1.5: Staggered columns of photo-detectors on iDXA. Imperfect registration of column 2 onto column 1 leads to horizontal stripes.
18 CHAPTER 1. IMAGE QUALITY ANALYSIS IN DXA

Figure 1.6: Artifact 2: horizontal pattern due to misalignment of the two columns of detectors for iDXA. The artifact is visible on the full sweep and causes discontinuities near edges every other line. The artifact disappears if we take only the even lines (detector column 1) or the odd lines (detector column 2). The images shown are high energy counts per second, for better visualization.

Non-uniformity of noise (photon starvation)

In areas where too few photons reach a detector, noise dominates the signal. This arises in dense bone areas such as along the spine, and results in visual inhomogeneity because of a stronger noise variance. This is known as photon starvation, because it originates in difference of attenuation of the X-ray beam.

Registration smear

As said in 1.1, detectors acquire partial views of the image, sweeps, that overlap with each other and can be registered to form the final image. In areas where two adjacent sweeps overlap, twice as much information is given so the sweeps are averaged in the final image. Averaging these parts also averages their noise component, so that in practice the resulting noise has lower variance in overlapping areas compared to non-overlapping areas. This results in visual marks with
delimitation of overlapping regions.

### 1.2.3 Ranking of image artifacts and perspectives of improvement

To rank these artifacts by severity, we gathered the database and wrote down for each image the impact of each artifact on the global quality. For instance, some artifacts are very localized whereas other impact the whole image. Some can reduce the visibility of important anatomical structures such as vertebrae or ribs and were judged more important. We also accounted for their frequency of apparition in the database. All the four artifacts described can be visualized in figure 1.7 on the final images (bone and tissue) after basis decomposition, denoising and registration of the sweeps.

![Figure 1.7: Left: review of the four artifacts on an iDXA image. 1: cross contamination, 2: horizontal lines from detectors misalignment, 3: noise inhomogeneity from photon starvation, 4: registration smears. Right: zoom on each artifact.](image)

It is easy to observe that the bone image is severely affected by cross contamination: dark lines on the vertebrae and parasite structures in the upper part of the image are all replicates of edges from the tissue image, introduced by the ACNR algorithm. The second artifact (horizontal lines) is also problematic and is more visible near edges,
which means it is amplified by cross contamination because the latter introduces fake edges. Photon starvation and registration smears are more subtle and concern mostly the tissue image. These are minor effects that cannot hinder the good detection of vertebrae, unlike cross contamination. Table 1.2 gives our ranking of the artifacts and which databases are concerned by each of them. For our application, we are mostly interested in improving quality of the bone image, so photon starvation and registration smears are less important. Because we cannot tackle all of them, we choose to focus on the most problematic, cross contamination. From now on, we will not deal with the three others and will instead dedicate Chapter 2 to the reduction of cross contamination artifacts by improvement of the Anti-Correlated Noise Reduction algorithm.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Artifact type</th>
<th>iDXA</th>
<th>Prodigy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cross contamination</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Horizontal pattern</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Photon starvation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Registration smear</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1.2: Artifacts in iDXA and Prodigy images.
Chapter 2
Adaptive Filtering for cross contamination Reduction

2.1 Review of denoising methods in medical imaging

The image quality analysis from Chapter 1 revealed that the major artifact was cross contamination, which occurs during Anti-Correlated Noise Reduction on the basis material images PBM and PST. Denoising is a crucial step in the process of acquiring good X-ray images that can be used for medical diagnosis purpose. In DXA images, high noise levels make it impossible to evaluate the bone mineral density or to visually assess fractures on unprocessed images. DXA remains a popular alternative for tomography or MRI because of the lower dose. However, this is at the cost of lower image quality and therefore the denoising operation is especially important. In this section, we will explain the main methods used in literature for the denoising operation on DXA or CT mainly, and then introduce the filter we will use.

2.1.1 Denoising methods for CT and DXA

Several authors have tackled the denoising task, acting at different levels of the processing chain described in 1.1. Denoising can be performed at the very beginning of the chain, on the raw counts per second images. At this stage, the noise can be decomposed into two components: electronic noise and quantum noise. Because electronic noise is negligible, most studies focus on the removal of quantum
Nowak and Baraniuk [8] showed that the latter follows a Poisson statistics, due to the random distribution of photons over the surface of detectors. This means that for a pixel at coordinates \((x, y)\) with true intensity signal \(\lambda\), the number of photons received by the detector is a random variable with probability law given by:

\[
p(I(x, y) = k) = \frac{\lambda^k e^{-\lambda}}{k!}
\]  

(2.1)

\(\lambda\) is the mean of the distribution as well as its variance, which means that the noise is signal dependent and varies with the local image features. This makes denoising of the quantum noise especially challenging in the spatial domain, and for this reason, most algorithms proposed operate on an image transform. For instance, in [9], Lu et al. use a multi-resolution wavelet approach. In the transformed domain, small wavelet coefficients (which carry most of the noise), are found with a thresholding method called covariance shrink and set to zero to filter out the noise. The denoised image is then recovered by computing the inverse transform.

In [10], Kwon et al. perform denoising on the low energy image only. The noise is separated into source and detector noises, which are reduced using Wiener filter and BayesShrink wavelet. M.A. Al-Antari [11] follows a similar approach by modeling source and detector noises, but this time for both the low and high energy images. They are empirically derived in phantom experiments, in which the source noise is found by counting photons at the detector with no object and the detector noise by counting them after covering the aperture of the X-ray source. The author uses a non local mean filter for both low and high energy, with smoothness parameters related to the noise power measured in the phantom experiments. However, he uses a single parameter for the whole image, making the assumption that the noise is stationary. We find this hypothesis highly restrictive in practice, as we have often observed spatially varying noise levels.

Some studies deal with the denoising part after the logarithm transformation, in the so-called Radon space. Although the raw photon counts follow a Poisson distribution, it was experimentally derived for CT data that noise approximately follows a Gaussian distribution in the Radon space (Wang et al. [12]). This allows to use a wide va-
riety of algorithms suited to the gaussian assumption. For example, Manduca et al. [13] use a bilateral filter coupled with a noise model on transformed data with a gaussian noise assumption to filter CT images. However, the authors in [12] also stated that the validity of the gaussian assumption decreases with the dose, which makes us question its applicability to our low dose DXA images.

Alternatively, noise can be reduced after basis decomposition. A popular denoising method is the Anti-Correlated Noise Reduction algorithm, briefly described in 1.1. Kalender et al. [14] expose an algorithm which takes advantage of the known property of anti-correlation between noises in basis material images to denoise one of the basis image by adding to it the estimated noise of the complementary basis image. Although it was originally developed for dual-energy computed tomography, the method is applicable for DXA and can be used to denoise the Pseudo Bone Mineral image using the noise of the Pseudo Soft Tissue, weighted by a factor which optimizes noise cancellation. Of course, we filter indirectly $PBM$ but still need to compute a low-pass filter for $PST$, to estimate the noise and separate it from the signal, so it may look like we have not gained anything. The justification for this indirect denoising is that the tissues typically have slower spatial variations than the bone structures, so they can be more easily separated from the noise component by a direct filtering. A direct low-pass filtering of the bone image would yield a risk of degrading edges of bones and could potentially lead to critical misinterpretations of the anatomical structures for fracture assessment. This algorithm is very effective in terms of noise suppression. In their original article, Kalender et al. already mention undesirable edge effects, similar to what we call cross contamination. Several variants of this algorithm have later been developed, differing in the way to compute the low-pass filter, ranging from a simple mean or median filter to more elaborated adaptive filters. Kalender’s ACNR algorithm can be summarized by the diagram of figure 2.1:
Figure 2.1: Workflow of the Anti-Correlated Noise Reduction algorithm. Noise from the tissue image is estimated by difference from PST and its low-pass filtered version, and later added with a proper weight to PBM.

2.1.2 Filters in image processing

Methods described in section 2.1.1 all make use of filters applied to images to separate signal from noise. For this reason, it is important to know their characteristics to decide if they are applicable in our study. This section reviews some common filtering techniques that can be used to denoise DXA images. Besides the noise reduction power, we are especially interested in the property of edge preservation, because we want to avoid cross contamination artifacts. Here, we simply describe their general behavior, and their relation to each other. A more precise mathematical formulation can be found in appendix, and examples of simple filters applied to our DXA images in section 2.1.3.

Linear filters

The most basic methods are linear filters. Such a filter can be expressed as the convolution between the input signal and a kernel function, called the impulse response of the filter. We can use them for denoising...
ing by taking a smoothing kernel, which outputs for a given pixel the average value in the neighborhood of this pixel. The kernel size sets the spatial extension of the neighborhood, and its coefficients set the weights of each pixel in the neighborhood. These coefficients are usually normalized so that they sum to one, ensuring the energy of the input signal is globally conserved. Two examples are the standard mean filter, which uses equal weights, and the gaussian filter, for which the weights follow a gaussian function of the distance to the center pixel. Linear filters are simple to implement and can usually be applied successfully to noise removal. In particular, Gaussian smoothing is efficient for Gaussian noise. However, these filters average pixels the same way everywhere in the image, so they tend to blur edges and other significant features that we might want to preserve.

**Median filter**

Another simple filter is the median filter. In this case, the input values in the neighborhood of a pixel are sorted and the filter outputs the median of these values. This filter is non linear, and can be better at preserving discontinuities. It is very efficient for the removal of salt-and-pepper noise, because unlike the mean, the median is not skewed by outliers. However, median filters tend to output piecewise constant functions, resulting in cartoon-like images appearing unrealistic.

**Bilateral filter**

The bilateral filter was first introduced in 1988 by Tomasi and Manduchi [15] as a nonlinear improvement of Gaussian filtering. Whereas a Gaussian filter only takes the geometric closeness into consideration for the weights computation, the bilateral filter adds a constraint on the photometric similarity. This is done by multiplying the spatial gaussian weight by another gaussian weight for difference of pixel intensities. This way, neighboring pixels with very different intensity values will not contribute much on the final pixel value after averaging over the neighborhood. In practice, this prevents edges from being blurred, giving the filter good edge-preserving properties. For this reason, the bilateral filter is a good candidate for our denoising application.
It has also been adapted in different versions with sophistications. For instance, the joint bilateral filter uses weights computed on another image, called the joint image. In a case of low contrast-to-noise ratio (high level of noise compared to edges and details to preserve), a bilateral filter would require a high intensity smoothness parameter to remove the noise, hence losing relevant image features. To avoid that, we can use a joint bilateral filter to compute weights on a slightly smoothed version of the image. It was originally introduced by Petschnigg, Agrawala and Hoppe while working on digital photography [16]. In a dark environment, a flash picture preserves much more details than a no-flash picture but the no-flash picture preserves the ambiance of the scene. Their idea was to combine information of both images to preserve details and ambiance at the same time. This was achieved by the joint bilateral filter, where the flash image was used as the joint image to enhance details of the no-flash image.

Another improvement on the original bilateral filter is its adaptive version, in which the spatial and intensity smoothness parameters are not fixed for the whole image and can be adjusted, based on local features (histogram of gradients, entropy, local noise standard deviation). This is useful if the noise is not uniform in the image, or if the level of details to be kept changes.

**Diffusion filter**

Diffusion filters are another important class of nonlinear edge-preserving filters. The idea of a filter generated by successive iterations of a diffusion process was first introduced by Perona and Malik in 1990 [17]. By analogy with the diffusion of heat inside a material, here the gray levels within the image are diffused as time increases, which results in smoothing. In their original article, the others explain the parallelism between a diffusion filter and the construction of a scale-space by convolution of the original image with gaussian kernels of increasing variance to produce coarser image resolutions. But while linear scale-spaces unavoidably lose features like edges when producing coarser views of the image, the diffusion filter suggested by Perona and Malik is non linear and looks for local image features in order to avoid inter-region smoothing. Nonlinearity is due to the spatially varying diffusivity coefficient, which is chosen as a function of the gradient in-
Note that although it has improperly been referred to as anisotropic diffusion by its authors, the Perona-Malik equation is in fact isotropic: it uses a scalar diffusivity coefficient. Following the work of Perona and Malik and the idea of non linear diffusion, Weickert develops another filter, this time non linear and anisotropic [18]. His idea is to compute a direction for smoothing, depending on local image characteristics. Like in the Perona-Malik model, the magnitude of the image gradient is computed, but instead of having one diffusivity coefficient, Weickert takes two coefficients: one for the direction of gradient and one for its orthogonal direction. This way, one can prevent diffusion across the edges while keeping diffusing along them (whereas the Perona-Malik diffusion simply stops at edges).

2.1.3 Illustration of filters on DXA images

Figures 2.2 and 2.3 show the behavior of a few filters on two different sweeps from the same image (iDXA dataset). The noise estimate is computed as the difference between the original PST and its filtered version, and added to PBM to perform the Anti-Correlated Noise Reduction, of which we also show the result.

On figure 2.2, the tissue image has homogeneous regions, with distinct separations in between. The level of noise is quite high, and as a result a gaussian filter with small standard deviation does not manage to remove it: instead it creates detrimental textures which are transferred to PBM. Conversely, a gaussian with large variance successfully removes the noise and does not introduce artificial texture, but it strongly blurs the edges delimiting organs in PST. This causes dark lines to appear on the denoised PBM image, which is a typical case of cross contamination. A joint bilateral filter with a low enough $\sigma_r$ manages to preserve sharp edges between organs, but it also preserves blobs within the same region when their intensity is different enough: this also results in texture artifacts on PBM (although less than for the gaussian). This already shows that a single choice of parameter is not enough to filter a whole sweep, hence the need for an adaptive bilateral filter. Finally, here anisotropic diffusion manages to preserve the edges and does not introduce structure artifacts.
Figure 2.2: The left column shows the PST original image and smoothing effects with different filters, and the right column shows the PBM image and its corresponding denoised version after adding the noise from PST.

Figure 2.3: Effect of different filters on an image during the ACNR algorithm. The same filters and parameters as on Figure 2.2 were used, but here they perform worse, hence showing that filters require adaptation.

On Figure 2.3, the tissue image is more heterogeneous. More precisely, the right part of the image has dense tissue and is quite homogeneous, which demands a heavy smoothing, while the mid-left part exhibits thin tissue structures to be preserved. The same observations can be made for the gaussian filter: a small kernel creates detrimental blob structures whereas a large kernel blurs too much the thin tissues.
The joint bilateral and the anisotropic diffusion do not perform very well either: they use the same parameter as for Figure 2.2 which here are not well suited (too much blurring for the anisotropic diffusion because of too many iterations or wrong level of anisotropy).

### 2.1.4 Reference denoising algorithm in enCORE software

As a image quality reference to our study, we very briefly describe the denoising scheme used in GE’s proprietary software for densitometry, enCORE. Denoising is performed on the tissue image using several simple filters with different scales. Tissue noise is added to the bone image in an adaptive manner to perform ACNR. After that, an additional denoising is performed on the bone image directly to remove more noise. For confidentiality reasons, we cannot give here more information about the specificities of the algorithm. Denoised images obtained with enCORE will be used for comparison in section 2.3.
2.2 Methods for Anti-Correlated Noise Reduction

In this section, we focus on the denoising of \( PBM \) images using Kalender’s ACNR algorithm. We remind that a direct filtering would degrade the resolution (blurring), which is why we instead cancel the noise with the phase-opposed noise from \( PST \). \( PST \) is easier to filter directly because tissues have slower intensity variations than bone structures. Because the noise is estimated by the difference from a smoothed image, the algorithm boils down to finding the best low pass filter for \( PST \), in the sense of maximal noise reduction and minimal cross contamination by edges or parasite filter-specific structures. On the basis of the observations from 2.1.3, we have highlighted two scales of variability in the tissue images:

- **inter-sweep variability**: tissues from different sweeps are different. The lower torso (below the diaphragm) contains large organs and few small structures, unlike the upper torso where thin objects have to be preserved by smoothing.

- **intra-sweep variability**: within the same sweep, we sometimes find homogeneous regions as well as heterogeneous ones displaying thin structures (especially for the upper torso).

For these reasons, the filters have to be made adaptive. Throughout this whole section, we detail our choices of filters and their parameters for denoising. Instead of using multiple simple filters combined in various ways along with post-processing steps like in enCORE software (described briefly in section 2.1.4), we design our own version of the ACNR algorithm using a single edge-preserving filter to prevent cross contamination, along with an adequate choice of parameters that we motivate, and we do not use any additional post-processing. Our methods are applied to an adaptive bilateral filter and independently to anisotropic diffusion, to obtain two filters that we will compare to enCORE. Although such filters are frequently seen in literature, their use is often described when working with stationary noise. In our case, high non-uniformity in noise as well as different degrees of contamination force us to design an effective strategy of parameter adaptation. To this end, we also describe the metrics we develop and use to measure local noise and cross contamination. We will often illustrate our
choices on the two sweeps already presented in section 2.1.3, denoted sweep A and sweep B, which contain enough qualitative variations to be representative of our data (see Figure 2.4).

Figure 2.4: Example bone (PBM) and tissue (PST) images before denoising: (a) Sweep A - PBM; (b) Sweep A - PST; (c) Sweep B - PBM; and, (d) Sweep B - PST.

2.2.1 Metrics

The two main quantities we want to monitor are noise and cross contamination.

Noise

The noise reduction in the PBM image is related to the quality of filtering of PST. For this reason, we can compute a noise level metric for the filtered PST image by looking at the ratio between the mean signal value ($PST_{LP}$) and the standard deviation of the noise ($PST_{HP}$). However, we known that noise is not uniform (even within the same sweep) so it is inaccurate to compute a global standard deviation. Instead, we define it by:
32 CHAPTER 2. ADAPTIVE FILTERING FOR CROSS CONTAMINATION REDUCTION

\[
\sigma(PST_{HP}) = \sqrt{\mathbb{E}(PST_{noise}^2) - (\mathbb{E}(PST_{noise}))^2}
\]  
(2.2)

The expected value can be thought of as a local average, computed by convolution with \(G_s\), a gaussian of scale \(s\) (which is defined \textit{a priori}). Hence, we get:

\[
\sigma(PST_{HP}) = \sqrt{G_s * (PST_{noise}^2) - (G_s * (PST_{noise}))^2}
\]  
(2.3)

This gives us a local noise metric \(N\):

\[
N = \frac{PST_{LP}}{\sigma(PST_{HP})}
\]  
(2.4)

which is then averaged over all pixels to get the global noise metric. This quantity has to be minimized, because we want to maximize the standard deviation of \(PST_{HP}\) so that as much as possible of the noise is removed. Figure 2.5 shows heat maps representing the local noise metric values for sweeps A and B, along with histograms showing the overall distribution of noise pixels.

![Figure 2.5: Local noise metric and noise distribution for Sweep A (top) and Sweep B (bottom).](image)
Although cross contamination is very easy to identify visually in terms of texture degradation or transfer of fake edges from PST, it is difficult to define a quantitative measure which is consistent with our intuition. A possible way of doing is to quantify the presence of low frequencies in the spectrum of the estimated noise, PST_{HP}. If the noise was perfectly estimated, there would be no correlation between adjacent noise pixels. However, when cross contamination occurs, edges are present in the noise image, creating spatial correlations between noise pixels, and thus the noise power spectrum is concentrated in the low frequencies. Figure 2.6 shows an example of the frequency power distribution in noise in case of little or heavy smoothing, resulting in very different levels of cross contamination for the same image.

Based on these considerations, we choose to define cross contamination $\chi$ as:
\[
\sum_{f_x, f_y} \omega_{f_x, f_y} |\mathcal{F}(PST_{HP})(f_x, f_y)|^2 \\
\sum_{f_x, f_y} \omega_{f_x, f_y}
\]

(2.5)

where \(\omega_{f_x, f_y}\) is the weight for spatial frequency \((f_x, f_y)\) and \(\mathcal{F}\) denotes the Fourier transform, computed with the fast Fourier transform algorithm. Because we want to penalize low frequencies, we choose the following weights:

\[
\omega_{f_x, f_y} = \frac{1}{1 + |f_x| + |f_y|}
\]

(2.6)

### 2.2.2 Adaptive bilateral filtering

In this section, we explain and motivate our choices to design a bilateral filter adapted to our problem. Because of the specifics of our problem (very noisy images and sensible phenomenon of cross contamination), we need to be careful regarding the choice of the parameters. Indeed, the results of a bilateral filter are very sensitive to the choice of the intensity smoothness parameter \(\sigma_r\) especially.

**Joint bilateral filter: influence of \(\sigma_r\) on noise and cross contamination**

Remember that a bilateral filter averages pixels which are close enough and similar enough in intensity. In our case, the contrast-to-noise ratio is low: the difference due to noise between adjacent regions is lower than the the contrast at edges between two regions. Therefore, to eliminate noise, we need to set \(\sigma_r\) very high, which in turn will undesirably blur edges. For this reason, we use instead a joint bilateral filter (see appendix, equation A.5). This means that the weights are computed on a median-filtered image rather than directly on the original image. This way, noisy pixels have a high weight and contribute to averaging the image within homogeneous regions (noise removal), while opposite from an edge, weights are small so the signal discontinuity remains sharp (edge preservation). Now that we have defined metrics for noise and cross contamination, we can investigate the role of \(\sigma_r\) and verify the interest of the joint bilateral filter.

On figure 2.7, we see that for both the standard and joint bilateral filters, increasing \(\sigma_r\) decreases noise but increases contamination.
However, for a given $\sigma_r$, the joint bilateral has a lower noise level and less contamination. In fact, even for very low values of $\sigma_r$, it manages to strongly reduce the noise, whereas the standard bilateral has much higher noise values (the noise plot on figure 2.7 is in log scale). This means that for a given noise level, the joint bilateral achieves lower contamination levels or conversely, if we fix a threshold on the amount of contamination, we get a much better denoising. Note also that for high values of $\sigma_r$, both filters tend to behave similarly, like a simple gaussian filter).

The joint image is computed as a median filtered version of the original $PST$ image, with a small kernel (3x3). Using a bigger kernel brings no advantage and only degrades the quality of the edges.

Figure 2.7: Influence of $\sigma_r$ on noise (left) and cross contamination (right), for a standard and a joint bilateral filter.

**Adapting the Joint Bilateral Filter parameters through local noise estimation**

The joint bilateral filter solves some issues of the standard bilateral. However, as already mentioned in 2.1.3, using the same $\sigma_r$ parameter does not allow to solve correctly the problem in every situations, because of inter and intra sweep variability. For this reason, we design an adaptive joint bilateral filter which automatically uses a local value of its parameter $\sigma_r$. $\sigma_s$ is set to a small constant value, as increasing it does not reduce the noise but only creates too much cross contamination and slows down the computation.

Our adaptive filter is entirely determined by the choice of $\sigma_r(x)$,
where \( x \) denotes any pixel location in the image. As we have seen, we need to adjust \( \sigma_r \) in the different image regions, to cope with different levels of noise. To that end, we relate \( \sigma_r(x) \) to the local noise standard deviation \( \eta(x) \). Like in 3.2, \( \eta(x) \) is computed using convolutions with a gaussian \( G_s \) by:

\[
\eta(x) = \sqrt{G_s \ast (a(x)^2) - (G_s \ast a(x))^2}
\]  

(2.7)

where \( a(x) \) is an a priori estimate of the noise (difference from \( PST \) and a small median filtered image). To get \( \sigma_r(x) \), we want to cast \( \eta(x) \) to a suitable range of values via an affine mapping.

However, there are two issues as such: first, \( \eta(x) \) contains extreme values that we want to exclude before the mapping. These outliers are located along the spine axis, where the noise standard deviation is too high because of photon starvation (see figure 2.8, point A). We set a threshold on the histogram of \( \eta(x) \) such that the 5% pixels with highest values are treated as outliers. If we did not exclude them, these pixels would have too high a \( \sigma_r(x) \) value and edges near the spine would be blurred. The second reason is that although this strategy ensures that homogeneous regions with different noise levels get different \( \sigma_r \), it does not ensure that the edge between the two regions get assigned a sufficiently small \( \sigma_r \) and be preserved (see figure 2.8, point B). We solve this by multiplying \( \sigma_r(x) \) by \( d(x) \), a weight related to the norm of the gradient of a highly smoothed image (so only strong edges are preserved):

\[
d(x) = 1 - \frac{\| \nabla (G_s \ast PST(x)) \|}{\max(\| \nabla (G_s \ast PST(x)) \|)}
\]

(2.8)

d(\( x \)) equals 1 in homogeneous regions and 0 at the strongest edges. As a result, the map \( \eta(x) \) is modified mostly near edges, where its value is strongly reduced to preserve sharp discontinuities in intensity. Finally, we obtain the desired map \( \sigma_r(x) \) with an affine relationship:

\[
\sigma_r(x) = ad(x)\eta(x) + b
\]

(2.9)

In this equation, coefficients \( a \) and \( b \) are pre-determined by sampling homogeneous regions with different noise standard deviations and finding manually an adequate \( \sigma_r \) for each of them.
CHAPTER 2. ADAPTIVE FILTERING FOR CROSS CONTAMINATION REDUCTION

Figure 2.8: From top to bottom: \( PST \) image, \( \eta(x) \) (local noise standard deviation), \( d(x) \) (gradient norm factor) and \( \sigma_r(x) \). The final map limits high values near the spine (point A) and very low values near strong edges (point B).

2.2.3 Anisotropic diffusion

Choice of the diffusion tensor

As an alternative to the bilateral filter, we implement an anisotropic diffusion filter. We base our filter on Weickert’s coherence enhancing diffusion (CED, see appendix, equations A.10 and A.11). Using this filter is motivated by the fact that the problem of cross contamination might be solved by using a directional smoothing to preserve thin edges in the \( PST_{LP} \) image, while smoothing along them. However, CED is not directly suited to our problem and needs adjustment: in CED, we set the diffusion strength along the gradient direction to be always very small (\( \lambda_1 = \alpha = 0.001 \)) and the diffusion strength in the orthogonal direction \( \lambda_2 \) to vary with the coherence \( \kappa \). This would lead to strong smoothing along sharp edges and no smoothing in any direction in homogeneous regions (where \( \kappa \) is small), which is not the desired behavior. Instead, we want the smoothing to be strong and isotropic in homogeneous regions (where there are few details to be preserved) and anisotropic in regions with edges. To this end, we make the following choice for the eigenvalues of the diffusion matrix:
\[ \lambda_1(\kappa) = \alpha + (1 - \alpha)e^{-\kappa} \]  

(2.10)

with \( C \) the value which tunes the sensitivity to edges, and \( \lambda_2 = 1 \). Hence, smoothing orthogonal to gradient is always strong. Near edges, \( \kappa \) is high and smoothing in the gradient direction decreases, creating anisotropy. Figure 2.9 illustrates the difference between CED and our filter.

![Diagram showing smoothing behavior between Coherence Enhancing Diffusion (left) and our filter (right), near edges and within homogeneous areas. Ellipses represent the local smoothing behavior and the arrows are the values \( \lambda_1 \) and \( \lambda_2 \).](image)

Figure 2.9: Schematic comparison of smoothing behavior between Coherence Enhancing Diffusion (left) and our filter (right), near edges and within homogeneous areas. Ellipses represent the local smoothing behavior and the arrows are the values \( \lambda_1 \) and \( \lambda_2 \).

**Choice of the sensitivity to edges**

However, we do not wish \( C \) to be constant, because we want the filter to have different behaviors according to the intensity level in the \( PST \) image, for a fixed level of anisotropy given by \( \kappa \). For instance, in high tissue density areas, we want the smoothing to be nearly isotropic and to disregard edges, but in low tissue density areas, we want to preserve as many edges as possible (highly anisotropic behavior) because these areas are the ones introducing most cross contamination. Like for the map \( \sigma_r(x) \) of the adaptive bilateral filter, we modulate \( C \) by the local noise standard deviation. Figure 2.10 shows the comparison
between the cases of anisotropic diffusion with a constant or adaptive coefficient $C$.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure.png}
\caption{Anisotropic diffusion: $\lambda_1$ values and corresponding diffused image, in the case of a constant coefficient $C$ (top) or adaptive (bottom).}
\end{figure}

The $\lambda_1$ maps on Figure 2.10 give the value of the diffusion coefficient in the gradient direction (black corresponds to $\lambda_1 = 0$, meaning no diffusion along the gradient, and white corresponds to $\lambda_1 = 1$, meaning fully isotropic diffusion). Because of strong gradients everywhere in the image, using a constant $C$ value gives approximately the same anisotropic behavior everywhere. As a result, the brighter area in the right half of the PST image also undergoes a highly anisotropic filtering. This leads to undesirable vortices and artificial anisotropic structures in the diffused image. These parasite structures are then transferred to the denoised PBM because of the Anti-Correlated Noise Reduction, yielding another form of cross contamination. In contrast, using a variable $C$, weighted by the smoothed signal value, allows to create a filter which is truly anisotropic in the dark areas with thin tissue structures, yet isotropic in the brighter areas of dense tissue. This reduces the apparition of cross contamination due to artificially created anisotropic structures.
cross contamination as a diffusion stopping criterion

The diffusion is performed independently sweep by sweep, before registration (the registration process is facilitated by denoising). We integrate the partial derivative equation of anisotropic diffusion with an explicit scheme, and accordingly set a small diffusion time step. For each sweep, we diffuse until a stopping criterion is met. Because we try to limit the cross contamination in Anti-Correlated Noise Reduction, we relate this stopping criterion to a measure of cross contamination, quantified by the frequency distribution in the estimated noise of PST. More specifically, at each diffusion time step $t$, we compute $I_t$, the current diffused image, with $I_0 = PST$ and $I_{t,noise} = PST - I_t$. Then, we compute the Fourier transform of the noise and sum its coefficients between a given frequency threshold. The reason for this is that along diffusion, more and more edges are unavoidably blurred, introducing low frequency content in $I_{t,noise}$. Thus, limiting the amount of low frequencies in the noise spectrum should limit cross contamination. Equivalently, this measure of contamination can be thought of as a measure of whiteness of noise: ideally, quantum noise is white and so is the noise on the basis material image after the $PBM/PST$ decomposition. Colored noise means spatial correlations and thus probable cross contamination.

It can easily be observed that this measure of cross contamination is monotonically increasing throughout diffusion, whereas noise level in the diffused image is monotonically decreasing. Hence, the chosen contamination threshold balances the relative importance of noise and cross contamination.

2.2.4 Weighting of the tissue noise component

Once the noise component $PST_{noise}$ has been extracted by means of an appropriate filter, it is weighted and added to the $PBM$ image, according to Kalender’s equation:

$$PBM_{Kalender} = PBM + \omega . PST_{noise}$$  \hspace{1cm} (2.11)

We need to determine the weight $\omega$. If $\omega$ is too small, the term introduced from $PST$ will not be enough to cancel the noise in $PBM$. If it is too large, we end up adding noise in the image. Ideally, we want the
denoised bone image $PBM_{Kalender}$ to be completely decorrelated from $PST_{noise}$, that is:

$$\text{Cov}(PBM + \omega \cdot PST_{noise}, PST_{noise}) = 0 \quad (2.12)$$

Expanding the covariance gives a linear equation in $\omega$, which is solved for

$$\omega = -\frac{\text{Cov}(PBM, PST_{noise})}{\text{Var}(PST_{noise})} \quad (2.13)$$

This gives us an analytical formula to compute automatically the weight for ACNR. By computing the Signal-to-Noise Ratio of the resulting $PBM_{Kalender}$, We can check empirically that the weight wound analytically corresponds to a maximum SNR (figure 2.11).

![Figure 2.11: Signal-to-Noise Ratio of $PBM_{Kalender}$ and covariance($PBM_{Kalender}, PST_{noise}$) for different values of the weight $\omega$.](image)

**2.2.5 Mode of evaluation**

Although we previously defined a quantitative metric for cross contamination for early stopping of diffusion, we do not use it a posterior measure of denoising quality. The reason for this is that we found out it did not always reflect the visual aspect of the denoised image.
Because the phenomenon of contamination is difficult to quantify, we choose to evaluate our filters with an observer experiment. In this study, an external observer is shown two images side by side. One of them is an image denoised with the algorithm from GE’s software (enCORE), and the other is the same image denoised using one of our filters (adaptive bilateral filter or anisotropic diffusion). The observer does not know which one is which, and is asked to tell which one is better for several criteria. We use a three-levels scale (better, similar and worse). The observer is asked to answer for 20 images of the dataset, and the experiment is repeated with different observers (four in total) for the same images, but in random order (to compensate the fact that observers are learning during the experiment and improve their classification).

The criteria are contamination artifacts on vertebrae, contamination artifacts in the background (ribs and area with no bone), noise level and sharpness. We split the contamination measure into two criteria for vertebrae and background because we observed major differences of quality between these areas. We include sharpness because of the common trade-off between noise level and blurring, decreasing the spatial resolution. The observers first receive a short preliminary training with examples for each criterion. During the experiment, they are shown full images and can zoom in to inspect details. The windowing is fixed and equal for the two images to be compared. By averaging the results for every images, each observer has a grade between -1 and 1 for every criterion. 1 (resp. -1) indicates that he considered our filter to be better (resp. worse) on all images for this criterion. Scores in between can indicate different repartitions of preferences: a score of 0 could for instance indicate that all images were considered comparable to enCORE, or that half of them were considered better but the other half was considered worse. This still indicates a useful trend when averaged over the dataset.
2.3 Results

2.3.1 Observer experiment

We report the results of our observer experiment on figure 2.12 with the mean and standard deviation (to indicate the level of agreement between observers). For contamination on vertebrae, which was our main focus, both our filters outperform the current system, and the adaptive bilateral filter received a higher score than anisotropic diffusion (0.83 versus 0.66). Moreover, the standard deviations are low in both cases (0.13 and 0.15), indicating a strong agreement between readers. For contamination in the background, both our filters are comparable and graded slightly worse than enCORE (mean -0.12 and -0.15) but the standard deviations are quite high (0.49 and 0.39). This means that the observer’s appreciation for this criterion is much more subjective.

Regarding noise, our filters were considered slightly better than enCORE (0.18 and 0.30). This can appear surprising because enCORE performs an additional filtering of the bone image after ACNR to further reduce noise, whereas we did not. In reality, our denoised images exhibited a higher pixel-based noise (originating in quantum noise) but had less colored noise coming from the denoising, which could give a better appearance and texture. However, the differences measured are hard to perceive in most cases and more than one out of two times the noise was perceived as equal by the reader. Finally, the sharpness was judged slightly worse in average for our filters (-0.27, -0.26). Again, this is surprising because we would expect our filters to have a better spatial resolution due to no additional filtering. In fact, spatial resolution is indeed equivalent or slightly better for our filters but the contrast was smaller near edges of vertebrae (due to no edge-enhancement post-processing). This was interpreted as a sign of smaller sharpness by some readers, whereas some other only considered the overall resolution. In consequence, the agreement between observers is really low for sharpness (standard deviations of 0.75 and 0.68), and this criterion should not really be considered as relevant.
2.3.2 Qualitative results

Typical cases of cross contamination appearing as dark lines on vertebrae are shown on figure 2.13. On these three cases, cross contamination is heavy on enCORE’s denoised images and strongly reduced by our filters. This greatly improves the visibility of vertebrae in most cases. Note that contamination does not completely disappear in our images, and can still sometimes be seen, although being reduced (case 3). Figure 2.14 shows two additional examples in the background. Case 4 gives an example of ribs area with little contamination for enCORE and our filters. The noise level and texture quality are comparable for the three processings, which shows that our methods do not introduce additional drawbacks. Case 5 gives an example of ribs area with heavy cross contamination. Here, our filters do not manage to properly reduce the artifacts compared to enCORE, and the visibility of ribs is low. This case demonstrates that our processings manage to handle cross contamination on vertebrae very well but not as well in
the background areas. This is due to the difficulty to find adaptive parameters that solve the issue for very different tissue images.

Figure 2.13: Comparison between enCORE and our filters (adaptive bilateral filter and anisotropic diffusion). On the three cases, our filters manage to reduce greatly cross contamination artifacts on vertebrae.

Figure 2.14: Comparison between enCORE and our filters (adaptive bilateral filter and anisotropic diffusion) in background areas. In most cases, contamination levels are similar among our filters and enCORE. Case 4: low contamination, case 5: high contamination.
2.4 Discussion

This chapter described our study of noise reduction in DXA images. Among the diversity of methods existing in the medical imaging literature, few are dedicated to DXA, whereas plenty of articles deal with other modalities (tomography, MRI). Methods commonly described are not always applicable with the same success to DXA, because of the significantly higher noise levels due to the low radiation dose. This is the main reason why direct denoising is harder and why we investigated and implemented Anti-Correlated Noise Reduction instead. We developed a strategy for an adaptive choice of parameters for edge-preserving filters to limit the apparition of artifacts observed in enCORE’s images and compared our results to the software.

As expected, the observer experiment confirmed that our adaptive filters managed to strongly decrease the level of cross contamination on vertebrae. This is a step towards global improvement of DXA image quality, as it can provide the radiologist clearer images to interpret. This can also be useful in the context of automatic vertebra segmentation because these artifacts could possibly hamper the good detection of vertebrae boundaries. We have seen that noise and texture quality provided by our filters were quite equivalent to enCORE, meaning there is no major drawback to our method. Here, we focused solely on the Anti-Correlated Noise Reduction step. For the comparison to be as fair as possible, a few additional processing steps should have been applied (like a small additional denoising). However, these subtle refinements are independent of the main method exposed here regarding ACNR and incorporating them would not change the conclusions we can draw from the use of our filters.

The experiment revealed that the adaptive bilateral filter and anisotropic diffusion had overall comparable performances. Although their mathematical formulation is different, they are both edge-preserving filters and thus have the same potential to prevent cross contamination, so this result is not really surprising. However, we expected anisotropic diffusion to handle better contamination of thin line-like elements in the background (lung structures in the upper part of the soft tissue image) due to its ability to smooth differently in two orthogonal directions. This was not observed experimentally: bilateral
filter and anisotropic diffusion performed similarly for contamination in the background and did not improve the results over enCORE. This suggests that there is still some work to be done regarding the choice of the parameters. These areas are most difficult to handle, and finding a single set of parameters for all sweeps in the whole image is a hard task. An additional difficulty is to find a relevant metric for contamination that suits the visual intuition. It is clear that in general, heavier smoothing means blurred edges hence contamination, yet not enough smoothing also generates undesired texture effects. Several metrics were tried, but none of them accounted for all the phenomena observed. With a better metric, a more accurate choice of parameters could perhaps be found through gradient descent of a cost function balancing the trade-off between noise and contamination.

Compared to the bilateral filter, anisotropic diffusion is naturally described in an iterative framework. This allows to refine the map of parameters at each time step and should in turn give more accurate gradients and smoothing. However, the diffusion filter has more parameters than the bilateral and proved to be harder to tune. Plus, although setting a threshold linked to a measure of contamination is a reasonable idea for a diffusion stopping criterion, this did not improve the results over the bilateral filter because of our imperfect contamination measure. In our implementation of filters, both took approximately the same time (0.1 second per sweep) so this was not a decisive criterion. However, we would expect the bilateral filter to be faster in an optimized implementation. For all these reasons, we tend to prefer the use of the bilateral filter which is easier to tune and gives comparable results.
Chapter 3

Vertebra segmentation in DXA images

In chapter 2, we investigated the question of image quality improvement through the denoising operation. This allowed to build sharp images that can be used by the radiologist for visual inspection or as input of an automated diagnosis system. In the second half of the thesis, we will now study the problem of vertebrae segmentation in DXA images. As explained in the overall introduction, a DXA exam serves two general purposes: it is used to derive the bone mineral density when looking for signs of osteoporosis and to detect bone fractures. Segmentation is of great interest for both problematics. First, BMD is usually computed as a mean value over regions of interest located on the lumbar vertebrae, and thus an accurate segmentation of these vertebrae will yield a good numerical evaluation of the BMD. Second, the vertebral shape outlined by the segmentation gives valuable information for fracture assessment (especially concavity measures of the upper and lower endplates). For these reasons, vertebrae segmentation is a critical operation which might heavily affect the medical diagnosis. In the following, we will start by presenting a literature review of segmentation techniques, with a special emphasis on recent deep learning approaches, which have been found to give state-of-the-art results. We will then explain our method, based on fully convolutional neural networks and apply it to our vertebrae dataset. Finally, we will evaluate the results in terms of common segmentation metrics.

For the sake of clarity, let us first begin with a review of the termi-
nology of classical problems in computer vision:

- Image classification is the problem of attributing a label to an image, corresponding to one specific class among a predefined set. The classification is often (but not necessarily) done according to which object is present in the scene. An example of a classification problem would be to infer whether an image contains a cat or a dog. In such a problem, we only assess the presence of an object but we do not give any localization information.

- Object localization is the problem of locating one or several objects, which could belong to different classes. However, we often mean by localization that we output a bounding box containing the object, and not its precise contour. Indeed, it is not always necessary to outline an object’s shape when we are only interested in a rough localization. Note that object localization can be thought of as a classification problem for sub-images in the original image.

- Object segmentation is the problem of delimiting the exact shape of one or several objects in an image. It is the last step towards coarse to fine inference, because here we want to make a pixel-wise classification.

As we will see, many segmentation methods are built on or derived from classification techniques, because classification has initially been addressed before segmentation in computer vision.

### 3.1 Literature review

Segmentation is a crucial problem in computer vision, and is especially sensitive when applied to medical imaging because it is used to infer the medical condition of a patient. Poor segmentations in a framework of automatic or assisted medical diagnosis will usually lead to wrong conclusions. Several authors have tackled the issue of segmentation and have come up with various algorithms based on different paradigms (thresholding methods, graph theory). We first present the main classical methods and then expand on deep learning techniques applied to the segmentation problem.
3.1.1 Common segmentation methods

Threshold-based methods

One of the most basic set of methods to perform segmentation is to use thresholds on gray level (or color) intensity. In its simplest setting, it consists in separating the object of interest from the background by comparison with a threshold. Methods differ in the way they compute the threshold. One of the most famous techniques is Otsu’s thresholding [19], in which the optimal global threshold is found to minimize the intra-class variance, or equivalently, to maximize the inter-class variance. Other methods, such as Ridler and Calvard’s [20], find the threshold in an iterative manner, similar to K-means clustering. Their main drawback is that the segmentation result is sensitive to the initialization. More sophisticated thresholding methods are adaptive, so they compute different values for the threshold based on local image features. Most of these methods can be extended to multi-class problems by using more than one threshold. The implicit assumption in all thresholding methods is that because objects are usually smooth and separated by sharp discontinuities, nearby pixels belonging to the same object will have similar intensity. For these reasons, these methods often fail for objects with different illumination conditions. Also, they do not incorporate prior knowledge about the object’s shape for instance, so they cannot distinguish the object from another adjacent object with similar intensity. For instance, in our problem, the whole spine has similar intensity than vertebrae and it would be hard to separate vertebrae only using a threshold-based segmentation.

Edge-based methods

Another simple way to segment images is to use edges to localize boundaries between regions. Edges can be found as maxima of the gradient (Prewitt or Sobel methods [21]) or zero-crossings of the Laplacian. These very simple methods can be improved by incorporating other steps like noise reduction, non-maxima suppression, edge linking and gradient thresholding (Canny edge detection [22]). However, these algorithms also suffer from similar drawbacks as threshold-based methods: edges are not always good indicators of true objects boundaries, as objects with different semantic meaning are not necessarily separated by an edge. In our case of vertebra segmentation,
there is no edge between a vertebra and the rest of the spine. This strongly restricts the applicability of these methods to a few and easy specific cases.

**Graph-based methods**

Image segmentation can be performed using a graph formulation, where the nodes of the graph are the pixels and the weights of the edges measure similarity between nodes. Hence, segmentation can be viewed as an energy minimization problem, for which many results exist in graph theory. Segmentation is obtained by a cut in the graph, which partitions it into two disjoint subsets of nodes by removing edges in between. The best segmentation with respect to a given energy function determining values of the weights is the one that minimizes the cost of the cut in the graph. This is the approach taken by Wu and Leahy [23]. This method suffered from the problem of very imbalanced segmentations due to the fact that the minimum cut preferred cutting very small sets of nodes from the graph. This was improved by Shi and Malik [24] with the normalized cut algorithm. Egger et al. [25] use a graph-cut method to segment vertebrae in MRI slices. They achieve a 0.91 mean Dice coefficient, but their method requires user input in the form of a seed point within each vertebra, so it is only semi-automatic.

**Active shape models**

Active shape models are a type of learning method in which a shape is deformed to fit an instance of an object. The set of possible shapes is constrained by a statistical model learned on labeled training examples. It was initially developed by Cootes et al. [26]. Each training example is defined by $n$ landmark points, so that for 2-D data (images) the object can be represented as a point in a $2 \times n$ feature space. In this space, all examples are aligned to the mean shape by scaling, rotation and translation, and the variability for each landmark defines an allowable shape domain, represented as a $2n$-dimensional ellipsoid. Then, a reduced number of independent modes of variation is found by principal component analysis. To find new objects in a search image, the landmarks defining it are iteratively updated to fit best the data while staying in the allowable shape domain, until convergence. This requires a rough guess on landmarks position to initialize the
fitting procedure. Compared to the previous methods, active shape models make use of prior knowledge on the object of interest in the image, so it is not fooled as easily by fake edges or lightning variations. However, it requires that different instances of the object do not vary too much in appearance, so that a useful statistical model can be built. Active shape models have been used for vertebra segmentation in X-ray images by Benjelloun et al. [27]. They model a vertebral body with 20 landmarks, so that each vertebra lies in a 40-D space. The authors report that the results are heavily dependent on good initialization of the landmarks for the iterative procedure. For this reason, their initialization is semi-automatic: the user provides two points per vertebra, and the rest of the landmarks is found using Harris corner detector points and a set of filters. They developed two statistical models, one for the vertebra and one (of really high dimension) for the whole spine. The conclusion of their experiments is that the spine model had worse results than the vertebra model, suggesting that automatic segmentation of all vertebrae on a full image is a hard task for active shape models. Other results of vertebrae segmentation using active shape models are reported in [28] in terms of Dice coefficient, as a benchmark for comparison with deep neural networks.

3.1.2 Other work on vertebrae segmentation or localization

Ghosh et al. [29] perform segmentation of lumbar vertebrae on 3D CT data. Their framework consists in a “handcrafted” method with several steps: they first locate intervertebral discs using an expectation-maximization algorithm, and get a rough localization of the vertebral skeleton by morphological processing. The obtained image is used to compute gradients, and a hough transform is then applied to retrieve geometrical lines giving the upper and lower vertebral endplates. Finally, a polynomial fit on the location of the discs is used to infer the vertebrae centerline, whose intersection with the endplates give major boundary points for each vertebra. They report a 15.5% false positive rate and a 5.1% false negative rate on lumbar vertebrae in average, and the authors show that their methods sometimes gave cases of complete failure, while it worked in average.

In [30], Chu et al. use a learning approach for the segmentation
of vertebrae in MRI and CT volumes. They use random forest regression to compute a probability map of centers of vertebral bodies. This map is spatially regularized with a hidden Markov model to obtain regions of interest for the segmentation. Finally, a random forest classification computes the voxel-wise likelihood of belonging to a vertebra within the ROIs, and combined with the prior obtained from a training set, this allows them to get the posterior probability for segmentation. They report a 0.887 Dice for MRI and a 0.910 Dice for CT.

### 3.1.3 Segmentation in enCORE software

In enCORE, the shape of a vertebra is defined with six points: four corners and two middle points on the upper and lower endplates. These points are used to compute ratios of height to identify fractures based on Genant’s classification. The points are found through a multi-step analysis on the denoised bone image with enhanced edges. Figure 3.1 gives an example of positioning of landmarks on vertebrae in enCORE. Although in many cases 6 points are sufficient to capture the shape of a vertebra, there cases for which this geometrical representation does not account much for the curvature of the lower and higher endplates, resulting in inaccurate positioning of the markers impacting the computation of the height ratios. Based on these observations, we decided to develop a fully automated method for vertebra segmentation that could possibly give better shape descriptors and more accurate positioning of the landmarks.
Figure 3.1: Two examples of interest points on the outline of vertebrae in enCORE software.

### 3.1.4 Convolutional Neural Networks for classification

With the increasing number of successes in deep learning since 2012, neural networks have been more and more often used to solve various machine learning problems. The field of image processing is no exception, and tasks like classification, localization and segmentation have been extensively solved with one category of networks, Convolutional Neural Networks (CNN). Although we are interested in segmentation, it is worth describing first the structure of CNNs for image classification, which was the first task tackled by deep learning in computer vision (see ImageNet Challenge, a classification problem with 1000 classes and over one million images [31]).

**Network architecture**

Unlike other machine learning algorithms such as Support Vector Machines who process features selected and extracted beforehand, CNNs find the classification features themselves. Hence, the architecture of a CNN for classification is in two parts: the first part deals with feature learning, and the second uses these features to classify the input images (see figure 3.2).
In the first part (feature learning), a set of filters compute different convolutions with the input image. This is followed by a nonlinear activation function, \( \sigma \). Often in convolutional networks, this function is a Rectified Linear Unit, defined by \( \text{ReLU}(x) = \max(x, 0) \), but other functions can be used (sigmoid, hyperbolic tangent, etc). The third step is a max-pooling layer, which computes a spatial maximum and whose effect is to down-sample the image to process for the subsequent layers. These basis bricks (convolution, nonlinearity and max-pooling) can be stacked onto each other to build deeper networks. At each max-pooling, the resolution of the image is degraded because of down-sampling, but the next convolution layers will process a bigger corresponding area of the original image, called its receptive field, and thus compute filters at larger scales. During learning, the network learns appropriate convolutions by tuning the parameters of each kernel so that it becomes able to classify correctly the training examples. Because of the structure of the network, the first layer has a small receptive field: it acts on small parts of the original image and learns low-level features (at the scale of a few pixels). On top of that, the next layers build more meaningful image items (edges, corners and texture) and so on. Eventually, at its last layer, the network has a good high-level representation of the object it has learnt to recognize. For this reason, CNNs are able to recognize objects by building a hierarchical representation made of large-scale features created from smaller-scale features.

In the second part of the network (classification), a standard neural network with fully-connected layers is used. Its input is the output of the last convolutional layer, flattened into a 1D array. The last layer transforms its outputs \((z_1, z_2, \ldots, z_M)^T\) into a vector of probabilities for each class \(p = (p_1, p_2, \ldots, p_M)^T\), defined by:

\[
p_i = \frac{e^{z_i}}{\sum_{i=1}^{M} e^{z_i}}
\]

(3.1)

This activation is called a softmax classifier, and compute probabilities that remain differentiable so we can apply the back-propagation algorithm. The layer outputs \(z\) are often regarded as log-probabilities, so that adding neurons outputs results in multiplying probabilities. By exponentiation and normalization, the softmax classifier thus outputs
"regular" probabilities.

Figure 3.2: Architecture of a Convolutional Neural Network for classification. The convolutional layers learn discriminative features, while the other fully connected layers and the softmax classify the image based on these features.

**Learning and hyper-parameters**

The softmax layer gives a distribution of probabilities among possible classes. Using a loss function $L$ (commonly cross-entropy, see 3.2), these probabilities are compared to a ground truth vector, giving the real class for the input image. By backpropagation, we compute the gradient of the loss with respect to each neuron’s weight in every layer, and update the corresponding weight $w$:

$$\Delta w_{i+1} = w_{i+1} - w_i = -\eta \frac{\partial L_i}{\partial w}$$  \hspace{1cm} (3.2)

where $i$ is the current iteration, $L_i$ the current value of the loss and $\eta$ a parameter called learning rate. This is repeated iteratively over the training examples until convergence of the loss function.

Although the network’s parameters are tuned automatically by gradient descent of the loss function, there remains a few "hyper-parameters" to be chosen carefully beforehand and which can impact significantly the results. Here are the most important and their effect:

- **Learning rate**: the step length $\eta$ in the gradient’s direction, used to update the network’s weights. A high learning rate means faster learning but also possibly instabilities because of gradient oscillations. In practice, the learning rate is also decayed along
• **Momentum:** because the loss function is usually not convex, we use momentum to avoid getting stuck in a local minimum. The weights are modified according to the formula:

\[
\Delta w_{i+1} = -\eta \frac{\partial L_i}{\partial w} + \alpha \Delta w_i
\]  

(3.3)

where \(\alpha\) is set between 0 and 1 and prevents the weight update to change too quickly. In other, this avoids oscillations by limiting variations of the weights from one iteration to another.

• **Batch size:** when the training dataset is large, one does not evaluate the loss directly by summing over all training samples. Instead, samples are presented randomly by batches of a given size, and the gradient of the training set is approximated by the gradient over a batch of examples, and backpropagation of the error occurs after each pass of a batch. One does not either want to present a single example for the computation of the loss function because it would lead to noisy gradients.

• **Network depth:** the depth of the network is an essential feature for the learning process. It has a great impact on the system’s performances because each additional layer gives one more non-linearity. More nonlinearities mean the possibility for the network to learn more sophisticated decision boundaries in the parameters space, which are often needed for difficult tasks. Recent work often use deep architectures, such as VGGNet (Simonyan and Zisserman [32]) and ResNet (He, Zhang, Ren and Sun [33]), which used up to 19 and 152 trainable layers respectively for the ImageNet challenge. Although depth allows more powerful representation, it also adds some constraints. Such networks have many parameters and can thus overfit the training data, by learning noise and data specifics instead of the underlying patterns, preventing a good generalization to unseen data. Moreover, deep networks have been found to be harder to train, due to the problem of vanishing gradients (when the error signal backpropagated towards the input gets too small through the network’s layers because of weight multiplication, leading to zero
gradient and layers not being trained). This problem has been known for a long time for convolutional and recurrent networks (Bengio, Simard and Frasconi [34]), but can be alleviated by the use of modern GPUs as well as several training tricks (layer-wise pre-training, rectified linear units with a 0 or 1 gradient to induce sparse representations of the data as in Glorot, Bordes and Bengio [35], instead of sigmoid activations).

- **Dropout**: one of the most commonly used regularization processes to prevent overfitting. At each iteration, the network randomly disables a proportion of its connections, hence "dropping out" the concerned neurons (Srivastava et al. [36]). For this current iteration, only the enabled connections can backpropagate an error signal and be trained. This gives the network robustness by forcing it to adapt to learn with sparser networks and prevents neurons to become overly specialized in some specific patterns. In practice, dropout is commonly used to reduce overfitting for deep neural networks. However, it is known to work well for fully connected layers but the results are mitigated for convolutional layers, although some studies have been made regarding its use in these situations (Wu and Gu [37]).

### 3.1.5 Fully Convolutional Networks for segmentation (FCN)

With the previous architecture, we have seen how to build a convolutional neural network for classification. Such a network needs inputs of a fixed given size, for the output size of the last convolutional layer to match the number of neurons in the fully connected layers. This structure can be adapted to solve segmentation tasks by intelligently up-sampling the network’s output to the size of the input. This is the original approach used by Long, Shelhamer and Darrell [38], in which the fully connected layers are replaced by up-sampling convolutions, resulting in a so-called fully convolutional network.

**U-Net architecture**

A classification network only resolves information about the type of data through the down-sampling process, at the cost of a loss of localization. To output a segmentation, i.e. a pixel-wise classification,
we need the network to up-sample the high-level features of the last convolutional layer until we obtain a probability map for each class, whose spatial dimensions equal those of the input image. Up-sampling is performed by interpolation. In terms of neural network layer, this can be achieved by a transposed convolution (sometimes referred to as deconvolution): the same way a standard convolution can perform down-sampling with an integer stride \( f \), a transposed convolution can perform up-sampling by using a fractional stride \( \frac{1}{f} \). We build a FCN by replacing the fully connected layers by deconvolution layers, and the original size is recovered if we have as many up-samplings as max-pooling in the first part of the network. These layers are of course trainable and are followed by a nonlinearity, yielding learned nonlinear up-samplings instead of simple bilinear interpolations for instance.

As such, the FCN would have very poor localization properties (especially in case of a very deep network where many up-samplings are needed) and would output a coarse segmentation map. The idea is to combine information from the deconvolution layers, which contain high-level feature representations but with bad localization, with information from the corresponding convolution layer in the first part of the network, which on the contrary has low-level feature representations but good object localization. This results in a symmetrical U-shaped architecture, where features from the down-sampling paths are copied and stacked onto the up-sampling corresponding layer (skip connections). Finally, as in the case of classification, the U-Net has a softmax activation at its very end to output the probability of each class, for any pixel in the image.

Note that because we got rid of the fully connected layers which impose a fixed input size, the FCN can process any arbitrary image, for training as well as for testing. This removes a huge limitation of most classification network which can only process one predetermined size of images.

U-Nets trained end-to-end or with transfer learning achieve very good results for segmentation, and have been successfully used for medical imaging (Ronneberger, Fischer and Brox [39]). The following diagram sums up the key parts of the U-Net FCN architecture:
In this example, the image to segment has dimensions 600x400, and the initial feature dimension (number of convolutional filters) is set to 32. There are 3 max-poolings in total in the down-sampling path, so 3 up-sampling 2x2 convolutions (deconvolutions) in the up-sampling path. Each time a down-sampling is performed, the number of filters is doubled. In this case, after the third max-pooling operation the image has a 75x50 resolution and 256 filters. The skip connections in gray show how information from the down-sampling path is copied to the up-sampling path. Note that zero-padding for convolutions is used in the case where the image original dimensions is not a multiple of two.

Ultimately, a 1x1 convolution maps the number of features to the number of classes, followed by a softmax activation which outputs the probability that each of the 600x400 pixels belong to any of the \( M \) defined classes. For each pixel, the class with highest probability is then chosen as the winner class.
3.2 Classification and quality metrics

3.2.1 Cross-entropy

Cross-entropy is a measure in information theory between probability distributions. For a discrete variable $x$ and distributions $P$ and $Q$, it is defined by $H(P, Q) = -\sum_x q(x) \log(p(x))$. Cross-entropy is the most commonly used loss function for optimization of neural networks on classification tasks, by taking for $P$ the probability vector output by the network’s softmax layer and for $Q$ the vector of true labels (0 everywhere except at the index of the true class). This definition can also be applied for segmentation because segmentation is pixel-wise classification. The cross-entropy for a whole image is then the average pixel-wise cross-entropy. For a two-class problem such as vertebra segmentation, $P = (p, 1 - p)^T$ and we can express the cross-entropy loss as:

$$H(P, Q) = -\frac{1}{N} \sum_{i=1}^{N} q_i \log(p_i)$$

(3.4)

where $N$ is the number of pixels, $p_i$ the vertebra probability for pixel $i$ and $q_i$ the corresponding binary ground truth value, $1 \leq i \leq N$.

In conjunction with the probabilistic point of view of the softmax classifier, minimizing the cross-entropy amounts to minimizing the negative log-likelihood of the true class which is the same as doing a maximum likelihood estimation.

When dealing with imbalanced classes like often in medical imaging (large background areas and small objects of interest), it is necessary to artificially balance the classes by weighting the training examples. The weighting can be incorporated in the cross-entropy loss function by assigning each class a factor inversely proportional to its prevalence in the training set (for example). Without weighting, the network would be biased towards the most frequent class and would predict it more often than it should.
3.2.2 Dice coefficient

Another metric often used in segmentation to replace cross-entropy is the Dice coefficient (or simply Dice). The Dice is a measure of overlap between two binary images $P$ and $Q$, defined by:

$$\text{Dice}(P, Q) = \frac{2|P \cap Q|}{|P| + |Q|} = \frac{2\sum_{i=1}^{N} p_i q_i}{\sum_{i=1}^{N} p_i + q_i}$$

(3.5)

where $N$ is the number of pixels and $p_i$ and $q_i$ are the binary values for pixel $i$, $1 \leq i \leq N$.

For binary segmentation, this can be turned into a loss function by taking for $p_i$ the softmax probability of the vertebra class and for $q_i$ the binary value of the ground truth segmentation mask, and minimizing the quantity:

$$DL(P, Q) = 1 - \text{Dice}(P, Q)$$

(3.6)

The advantage of Dice against cross-entropy is that it does not require weighting.

3.2.3 Other common metrics

Some other criteria are not loss functions but usually seen as evaluation metrics at test time for binary classification or segmentation tasks. If $P$ is the binary prediction (thresholded softmax probability) and $Q$ the ground truth segmentation, we have:

- Sensitivity (True Positive Rate):
  $$TPR = \frac{|P \cap Q|}{|Q|}$$
  (3.7)

- Specificity (True Negative Rate):
  $$TNR = \frac{|\bar{P} \cap \bar{Q}|}{|\bar{Q}|}$$
  (3.8)

- False Positive Rate:
  $$FPR = 1 - TNR$$
  (3.9)

- False Negative Rate:
  $$FNR = 1 - TPR$$
  (3.10)
3.3 Data acquisition

In our study, we consider two datasets of bone images, acquired with two different measurement devices from GE Healthcare: iDXA and Prodigy. We refer the reader to section 1.2.1 for a summary of these datasets characteristics. For information, each image contains in average ten fully visible vertebrae. All images do not necessarily have the same number of vertebrae, but they contain both thoracic and lumbar vertebrae (no cervical). No segmentation was available initially, so we had to generate our own ground truth masks to be used for training and numerical evaluation of results. Because the process of manual segmentation is very long, we only did it for the iDXA dataset, which will thus be our only source of training and quantitative results. The Prodigy dataset will serve for additional comments about the network’s ability to generalize to images from a different device.

3.3.1 Ground truth annotation

Using the image edition software Gimp, we manually labeled the 44 iDXA images by choosing landmark points on the vertebrae boundaries. A closed curve is automatically generated for each vertebra by linear interpolation between landmark points, and this defines a binary mask (see figure 3.4). Note that segmentation is an ill-posed problem: because images are the 2D projection of a 3D volume, we often find multiple contours for each edge to locate. For this reason, there is no such thing as true segmentation. However, what we can do is being as consistent as possible for the choice of edges, so the network is fed with the same boundary characteristics for each training patch. We choose to systematically take as the boundary the inner edge, which is usually the brightest. While our segmentation might differ from a radiologist’s, what we are trying to achieve here is to see whether a network can learn our segmentation - even though it may not be optimal from a clinical point of view. If we were to use our network to classify fractures based on the segmentation, we would have to make sure that the segmentation provided is good enough and use ground truth annotated by a specialist, but this is not the point here.
3.3.2 Data preprocessing

Although a fully convolutional network can process images of arbitrary size, the GPU used for training receives data by batches which must have the same dimensions. For convenience, we generate our training set from the iDXA images by sampling patches of fixed dimensions. We choose a size of 160x128 pixels, such that a patch centered on a vertebra contains it all (in iDXA resolution). Patches are sampled anywhere in the images, to contain the object of interest (whole vertebra or parts of adjacent vertebrae) as well as non-objects (ribs, spine and soft tissue areas). Patches are selected automatically, by random sampling of coordinates defining their center.

Images are not processed apart from denoising (described in the first part of the thesis) and normalization (subtraction of the training set’s mean value and division by its standard deviation). In some experiments, images are also equalized to investigate the effect of contrast on the resulting segmentation.
3.3.3 Data augmentation

We artificially increase the size of the training set by randomly applying transformations on an image before extracting each patch. We consider the following transformations:

- **Rotations**: each patch is rotated randomly of an angle between \(-15^\circ\) and \(+15^\circ\). These limits roughly correspond to the observed angles of vertebrae along the spine (due to its natural curvature).

- **Scaling**: each patch is scaled at random with a factor between 0.8 and 1.2. We consider two independent scaling parameters for the two dimensions of the image, so as to allow not only variation of size but also of aspect ratio.

We did not apply horizontal flipping, because vertebrae are always on the right of the spine in our images (systematic procedure of patient positioning). If it was not the case, we would have to include flipping as an additional transformation. We do not either include elastic deformations. Although they would add a lot of variability to possible vertebrae shapes, it is risky to include them for a system whose ultimate goal is fracture analysis.

Figure 3.5 shows examples of patches in presence of vertebrae or ribs, with diverse rotation angles and scaling factors.

![Examples of patches extracted from images (top) and corresponding ground truth segmentation mask (bottom).](image-url)

Figure 3.5: Examples of patches extracted from images (top) and corresponding ground truth segmentation mask (bottom).
3.4 Methods

3.4.1 Network architecture

We use the Fully Convolutional Network architecture with skip connections between the corresponding convolutional and deconvolutional layers (U-Net), as described in 3.1.5. The contracting path is a succession of blocks, each of them containing, in order, convolution, activation, convolution, activation, max-pooling. Symmetrically, the basis block in the expanding path contains deconvolution (concatenated with features from the same level convolution by skip connection), convolution, activation, convolution, activation. A final 1x1 convolution with softmax activation outputs the map of pixel-wise vertebra probabilities. We use small convolution kernels (3x3) with padding and 2x2 max-poolings with stride 2 (which thus operate a factor 2 down-sampling). Each convolution layer is followed by a ReLU activation and a batch normalization layer (see 3.4.2). The advantage of ReLU over sigmoid or tanH is that it is non saturating, so the gradient, which is always either 0 or 1, does not vanish through the succession of layers.

We denote by UNet-n the architecture with n blocks in the contracting path. Hence, it has n down-samplings, for a minimal resolution of $\frac{1}{2^n}$ times the original image resolution, along each dimension. Table 3.1 shows the number of layers (convolutions, deconvolutions and max-poolings) and the total number of trainable parameters for different depths of U-Nets.

<table>
<thead>
<tr>
<th>Network</th>
<th>Layers</th>
<th>Total parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet-2</td>
<td>15</td>
<td>538850</td>
</tr>
<tr>
<td>UNet-3</td>
<td>21</td>
<td>998050</td>
</tr>
<tr>
<td>UNet-4</td>
<td>27</td>
<td>7771330</td>
</tr>
</tbody>
</table>

Table 3.1: Number of parameters for different network depth

**Remark:** following the approach of Simonyan and Zisserman for VGGNet in [32], we choose to use stacks of two 3x3 convolutional layers between max-poolings, instead of for example one 5x5 convolutional layer. As explained in their paper, two 3x3 convolutional layers
with stride 1 and no down-sampling in between result in an equivalent receptive field of 5x5, just like a single 5x5 convolutional layer. So this stack of two smaller filters has the same contextual view (in terms of pixels from the preceding layer) but with fewer parameters \(2(3^2C^2) = 18C^2\) instead of \(5^2C^2 = 25C^2\) where \(C\) is the number of channels) and also benefits from one more nonlinearity, which allows to learn more complex decision functions. This is interpreted in [32] as a regularization of larger kernels (like 5x5 or 7x7) which must learn a decomposition in the form of stacked smaller kernels.

### 3.4.2 Training specifics

**Hyperparameters**

Our networks are trained with stochastic gradient descent on the Dice loss, presented in 3.2. We use a high momentum \((\alpha = 0.9)\) and a decayed learning rate. We use a rather small batch size (16 image patches) due to GPU memory limitations. As we did not experience overfitting, we do not use dropout or weight decay regularization. The training is done on a Nvidia GTX 690, and stopped when the validation loss does not improve any more.

**Batch normalization**

It is well known that the training process of a neural network is facilitated if all data follow the same statistics, and this is why we preprocess them so that each batch has zero mean and unit variance as input of the network. However, even if the batches flowing through the input layer are preprocessed this way, their statistical distribution may vary in the next layers, due to weight multiplications and activations. Throughout the learning process, the network’s parameters change within a layer and for this reason the distribution of the next layer’s inputs changes also. This phenomenon, referred to as internal covariate shift by Ioffe and Szegedy in [40], makes the training of the subsequent layers harder. As a result, the training of the whole network is slowed down considerably. The solution presented in [40] is to normalize batches after each layer inside the network. The intuition is that Without batch normalization, the network has to dedicate some of its learning power to adjust to a wide range of numerical values,
whereas with batch normalization it only focuses on learning the patterns on statistically similar data.

During the remaining of this thesis, we therefore use batch normalization (BN) systematically as it speeds up the training. As an example, figure 3.6 shows the influence of using batch normalization on the learning curves. The network used is UNet4, with the same learning rate (0.01), batch size (16 patches) and data (training patches taken from 36 iDXA images, without data augmentation).

![Figure 3.6: Dice loss for training and validation sets, with and without batch normalization.](image)

Figure 3.6 shows that using BN does not impact the final performances of the trained network: the validation Dice loss reaches approximately 3% in both cases. However, convergence is achieved much faster in the case of BN: 9 epochs are enough to reach a 5% Dice loss on validation, whereas it takes around 40 epochs to get the same without BN. In practice, for our data, the learning process can be stopped at a third of the time it would take without BN, which is of great practi-
cal interest when running many simulations. We also verify that this property is true for different network sizes so that it can be always be used to achieve the same performances in less time.

Figure 3.6 also shows that with BN, the training loss is still slowly decreasing from 20 to 140 iterations, but the validation loss plateaus. This indicates the network would fit better and better the training data but with no performance gain as generalization to unseen data is concerned, so we can stop the training.

### 3.4.3 Test methodology

**Tests on iDXA**

This database is the basis of our main experiments. We investigate mostly the influence of network’s depth and amount of training data on the Dice, computed on several test images. Whereas the network is trained on patches, at test time its input is directly the whole image (average size of 1500 x 300 pixels). For each experiment, we cross-validate the results by repeating the training 10 times with different partitions of the data into training and test sets, and ensure that each image is tested or trained an equal number of times. This provides mean and standard deviation values for our metrics. We report the results in terms of sensitivity, specificity and Dice. We choose not to report the pixel-wise accuracy, which is not an indicative metric for imbalanced datasets. Although the same can be argued for sensitivity and specificity, we report them because they are often used for segmentation in literature. However, they remain indicative only and our main evaluation metric is the Dice coefficient.

**Tests on Prodigy**

We also test our networks on the Prodigy images, which have never been trained on. This is to assess the network’s ability to generalize to another database, with different characteristics. Indeed, it is important to know if one can use a cross-platform network or if each different scanning device needs its own network. Because we do not have ground truth for Prodigy, we will only present qualitative results by showing test images.
Remember that Prodigy and iDXA have two different resolutions. Because the network learns filters at a given scale, it is not resolution invariant so we must rescale the Prodigy images at test time to match iDXA’s resolution. We solve this by up-sampling the Prodigy images at test time to match iDXA’s resolution, so that the network actually compute inference on images of the same size that it has been trained on. The up-sampling factor is 4 along the row axis, and 2 along the column axis. It is anisotropic because it depends on the pixel size for the row axis, but also on the scanner’s speed for the column axis (motion of the mechanical arm stretches shrinks pixels in this direction). This factor is very large, hence the bilinear interpolation used for up-sampling introduces a heavy blurring effect. For this reason, we introduce a blur in the training (iDXA) images by filtering each patch with a gaussian kernel. Along with rotations and scalings already discussed in 3.3.3, this additional data augmentation artificially adds robustness with respect to blurring. The diagram on figure 3.7 sums up the up-sampling and blurring operations at test and training time respectively.

![Diagram showing the process of predicting segmentation for Prodigy images with a network trained on iDXA.]

Figure 3.7: Predicting segmentation for Prodigy images with a network trained on iDXA.
3.5 Results

3.5.1 Performance evolution with respect to depth

Here, the amount of training data is fixed. We use 36 iDXA images for training, for a total of 3600 patches. Several network configurations were tried and compared, all based on the U-Net architecture but differing by their depths. The test set contains the 8 remaining iDXA images. The mean results and corresponding standard deviations for sensitivity, specificity and Dice coefficient are reported in Table 3.2.

<table>
<thead>
<tr>
<th>Depth</th>
<th>Training time (min)</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Dice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>St. dev.</td>
<td>Mean</td>
</tr>
<tr>
<td>UNet-1</td>
<td>33</td>
<td>0.867</td>
<td>0.024</td>
<td>0.917</td>
</tr>
<tr>
<td>UNet-2</td>
<td>39</td>
<td>0.895</td>
<td>0.017</td>
<td>0.968</td>
</tr>
<tr>
<td>UNet-3</td>
<td>47</td>
<td>0.941</td>
<td>0.019</td>
<td>0.986</td>
</tr>
<tr>
<td>UNet-4</td>
<td>61</td>
<td>0.949</td>
<td>0.012</td>
<td>0.988</td>
</tr>
<tr>
<td>UNet-5</td>
<td>97</td>
<td>0.950</td>
<td></td>
<td>0.988</td>
</tr>
</tbody>
</table>

Table 3.2: Test performance for different network depths

The results from Table 3.2 show that the quality of the segmentation highly depends on the network’s depth. Shallow network (UNet-1 and UNet-2) do not perform well and have low mean Dice coefficient over the test set. Deeper networks give much better segmentation results, but at some point the performances saturate and increasing the depth does not lead to better results. For instance, UNet-5 exhibits results comparable to UNet-4 on the test set, although it manages to get a smaller training loss. This suggests that at this point, adding more layers only leads the network to overfitting the training data, with no benefit on its ability to generalize. For this reason, and the fact that deeper networks take longer to train, we do not investigate deeper networks than UNet-5, and stick to UNet-4 as our best model. Note that we did not report the standard deviation for UNet-5, as it was only cross-validated on three different experiments. However, it did follow the trend of very low variation between models from different data and it is to be expected that the standard deviation would be very close to the one obtained with UNet-4.
Figure 3.8 gives an example of predicted segmentation with different depths. The prediction is made on a test set image, and after convergence of the training. As we can see, UNet-1 and UNet-2 give poor segmentation results. The predictions of vertebra points tend to cluster around actual vertebrae, but there remains a lot of false positives for UNet-1 (on the spine, which the network has trouble to distinguish from vertebrae) and even in soft tissue areas. These networks also exhibit too many false negatives, leading to holes in the vertebrae. Globally, their predictions seem to lack spatial coherence. In contrast, UNet-3 and mostly UNet-4 give much more accurate segmentations. Beyond the greater number of parameters, the reason for these results is in the increased size of the receptive field allowed by more convolutions and max-poolings. A wider spatial context helps the network to make spatially consistent predictions.

![Figure 3.8: From left to right: original image and predictions from networks with increasing depths. Predicted segmentations from U-Net5 look identical as U-Net 4.](image)

### 3.5.2 Performance evolution with respect to data

From now on and until the end of this thesis, we do not vary depth any more and use UNet-4. Here, we investigate the relationship between the number of data used for training and the evaluation metrics. Because the right amount of data depends on the model’s complexity
and the inherent difficulty of the task, there is no general rule to determine \textit{a priori} how much data is needed. Among the 44 iDXA images of our dataset, we trained networks on \( N \) images, for different values of \( N \). For each \( N \), we tested the network on 8 of the \( 44 - N \) remaining images, and cross-validated the results by repeating the experiment 10 times after shuffling the data. We used 8 images only and not all of the \( 44 - N \) so that the variance do not be skewed by the number of test images. The training set was each time constituted by sampling 100 patches from each of the \( N \) images, with random rotations and scalings. Figure 3.9 shows the evolution of the mean sensitivity, fall-out(1 - specificity) and Dice coefficient with respect to \( N \). Error bars represent one standard deviation. The same network (UNet-4) was used in all experiments.

![Figure 3.9: Sensitivity (orange), fall-out (green) and Dice (blue) for different number of training images. Vertical bars are equal to the standard deviation of metrics, computed on the 10 folds cross-validation.](image)

These curves show that very few images are needed to obtain good results on the iDXA dataset, provided enough patches are sampled from each image. Note that here the fall-out rate is almost constant and really low, due to heavy class imbalance (which is why the Dice is a better performance indicator). Whereas we could expect the Dice to be steadily increasing with the number of images, its mean value saturates around 0.95 quickly. For instance, from \( N = 18 \) to \( N = 36 \), the Dice only increases of 0.01 whereas we doubled the number of im-
ages. This is not very surprising as all the images look quite similar, so a few of them are enough to be representative of the whole dataset and the network only needs a few cases to grasp relevant segmentation features. However, one should keep in mind that our dataset did not contain severe fractures, which are harder to segment and would probably skew the test Dice. If we were to include cases of fractures, we could expect more diversity of shapes and thus a lower mean Dice and a higher standard deviation. Here, the standard deviation is really low (less than 0.02 for $N \geq 10$ images). Even though we did not make the same experiments with different network depths, we expect the qualitative trend to be the same, with shallower networks having very low performances for little training data.

### 3.5.3 Example results on iDXA

Figure 3.10 gives an example of segmentation results on a test image from the iDXA database. Overall, the prediction is accurate for each vertebra, and the total error seems equally distributed among all of them. In a lot of cases, the predicted segmentation accurately follows the natural curvature of the vertebra endplates, which indicates that our method has a good shape description capability. Note that the two adjacent vertebrae on top of the image are not separated in this case, and form a single connex component after thresholding the network’s predicted probability map. Other common issues are small holes inside a vertebra (false negatives) or small clusters outside (false positives). Hopefully, they can easily be distinguished by their size and removed by morphological post-processing before labeling connex components into actual vertebrae.
Figure 3.10: Example of segmentation results for an iDXA image from the test set (Dice = 0.948). From left to right: Original image, ground truth labels, predicted segmentation, pixelwise error type.

### 3.5.4 Generalization to the Prodigy database

Figure 3.11 gives qualitative results of the segmentation on up-sampled Prodigy images. Obviously, the results are not as good as on iDXA. More specifically, the segmentation often fails in the upper part of the image (vertebrae above T8). We argue that this is because these vertebrae usually have smaller intervertebral distances so they are harder to separate. This, combined with Prodigy’s rather low resolution, leads to blurry and low-contrasted vertebral edges that the network does not identify as a boundary. In contrast, for the same intervertebral distances one would see sharper edges on an iDXA image, due to higher resolution, which is why iDXA’s predictions are better. However, the segmentation quality is much better in the mid-lower part of the images. This can be because lumbar vertebrae generally have bigger intervertebral distances than upper thoracic vertebrae, so they are much
easier to separate. For these cases, qualitative results look consistent with the ones obtained on iDXA.

3.5.5 Influence of cross contamination on segmentation

The phenomenon of cross contamination, examined in chapter 2, can create parasite dark lines across vertebrae. This could be harmful for the segmentation process, because the high contrast area introduced could be mistaken for the outline of a vertebra, causing the failure of the task. In practice, this does not impact often segmentation. This is not a surprise, because the neural network has been trained on images with contamination so it has learned to overcome this problem. Moreover, because of the architecture of the U-Net, it relies on local as well as more global features and the deep layers see a wide context with a whole vertebra. However, it does fail on a few marginal cases.
these cases, denoising the image with our method (adaptive bilateral filter) leads the same network to recover a good segmentation, thanks to the reduced cross contamination. Figure 3.12 gives an example of this phenomenon.

![Figure 3.12](image)

Figure 3.12: Top: Comparison of segmentation produced for an image from the current system (top) and the same image denoised with our adaptive bilateral filter. In both cases, the same network was used (not trained on this image).

### 3.5.6 Comparison with the current segmentation system

Here, we compare the results of our chosen network, U-Net 4, to those produced by GE’s software enCORE (a six points per vertebra segmentation), on iDXA images. To get an objective numerical comparison between enCORE and U-Net 4 would require an external ground truth segmentation, performed by a radiologist. Remember that we created the ground truth ourselves by manual annotation. If for some reason our ground truth is imperfect from a radiological point of view, the neural network, which has been trained on it, will be biased. So even though U-Net 4 may score better than enCORE with respect to this particular ground truth, we could not logically draw conclusions for
its radiological interest. Hence, because we did not have an external ground truth, we chose not to evaluate numerically U-Net 4 against enCORE. This being said, we can still compare them qualitatively on images and comment on the differences. Again, every segmentation we provide comes from a test set image.

**Improvement on vertebra outline localization**

In many cases, the line segments provided by enCORE to delimit vertebrae are too coarse and our network finds a more descriptive shape. Figure 3.13 shows a comparison between the geometric segments obtained with enCORE and our segmentation. The differences in shape and landmarks localization, although minor, would impact the computation of intervertebral distances used for fracture assessment.

![CASE 1](image1.png) ![CASE 2](image2.png)

**Figure 3.13:** In both cases, U-Net 4 manages to follow accurately the upper and lower vertebral endplates. In comparison, the line segments given by enCORE do not follow the curvature of the vertebra outline very well.

**Improvement on vertebra detection and labeling errors**

Out of our 44 iDXA images, we reported three cases of failure of vertebra detection from enCORE’s segmentation tool. In these three cases, a vertebra was missed, and the subsequent labeling was skewed (shift of labels and corresponding information for vertebral fracture assessment). In these three cases, our network recovers the missed vertebra. Figure 3.14 shows such examples.
Figure 3.14: Two cases of mislabeling from enCORE. Each time, U-Net 4 corrects the detection error. Case 1: L1 is missed and the vertebra labels are shifted for every vertebra above L1. Case 2: L5 is missed and all the vertebra labels are shifted (labeling is performed bottom-up).

Segmentation failure case

In rare cases, the network fails to segment accurately a vertebra. This typically happens when another bone is overlaid upon the spine, resulting in a high-contrast area within the same vertebra because of the perceived difference in areal bone density. This creates an edge inside the vertebra where the vertebral outline is mistakenly set. Note that for the two cases present in our database, the segmentation from enCORE also fails. Hopefully, such cases are not frequent because they result in bad positioning of the patient on the DXA table (for instance, an arm is visible in the scan).

For these cases where the predicted shape is irrelevant, we manage to solve the problem with an alternate network, trained on a modified loss function. The idea is that by explicitly adding a constraint on the predicted edges within the loss, we might recover better the actual vertebra boundaries. We build this shape-constrained loss from the Dice by adding a new term proportional to the Dice between predicted boundaries and ground truth boundaries, in order to penalize heavily shape errors. The boundary, which is where the spatial gradient of probability is maximal, is located by a threshold on the probability gradient given by the last layer of our network (softmax). Since the threshold must remain differentiable for back-propagation, we com-
pute a soft threshold with a sigmoid. Mathematically, this modified loss is defined by:

\[
DL_{\text{gradient}}(P, Q) = DL(P, Q) + \lambda DL(s(\|\nabla P\|), s(\|\nabla Q\|))
\]  

(3.11)

where \(DL\) denotes the Dice loss defined in section 3.2.2, \(P\) and \(Q\) are respectively the network’s output and the ground truth, \(s\) the sigmoid function and \(\lambda\) the parameter which tunes the relative importance of the two terms. This approach is close to the one taken in [28], dealing with cervical fractures.

Figure 3.15 shows an example of case in which enCORE and U-Net 4 fail, and the corresponding segmentation given by U-Net 4 trained on the modified loss.

Figure 3.15: Bone overlaid on vertebra, causing a segmentation failure for enCORE and U-Net 4 (vertebra T7). In comparison, the network trained on the modified loss recovers more accurately the right boundary.
3.6 Discussion

Our best model, U-Net 4, showed good performances overall, with a 0.940 mean Dice and high consistency (standard deviation 0.011 between different cross-validation folds). In comparison, [28] reports a 0.938 mean Dice with a similar network. However, their dataset is made of cervical vertebrae with many cases of fractures, whereas ours had thoracic and lumbar vertebrae and no fractures, so any comparison is indicative only. Still, using the same training data as for their neural networks, they report a 0.883 mean Dice for their best performing active shape models, suggesting a strong superiority of neural networks on this task. The authors also report a slight augmentation of the Dice by the use of a shape-constrained training with a modified loss function (0.944). We did not observe any improvement on the Dice with our modified loss, although it increased the segmentation results on specific failure cases.

Even if we did not test our models on cases of heavy fractures by lack of data, it remains the ultimate long-term goal of such a study on segmentation, so constraining a priori the segmentation to a set of shapes learned on healthy vertebrae might be a dangerous idea, as we may recover regular shapes at test time for fractured vertebrae instead of their real boundaries.

Although Dice or cross-entropy remain gold standards for segmentation, modifying the loss function to suit a specific problem’s needs is still an interesting idea. More precisely, it might be interesting to design a loss function that explicitly uses softmax probabilities of neighboring pixels to improve spatial coherence of the prediction. Indeed, even though deep networks have a large receptive field (4 down samplings for U-Net 4), we still find inconsistencies in the predicted segmentation map, such as holes of very few pixels within a vertebra. A common technique is to add a post-processing step with a conditional random field for instance, to diminish these inconsistencies. However, this splits the optimization problem into two distinct tasks and adds parameters, and incorporating it in a single neural network framework might offer better results.
Conclusions

Our study on Dual Energy X-ray Absorptiometry imaging included several steps. A careful analysis of our database allowed us to highlight several artifacts that degraded the quality of our DXA images at different degrees (Chapter 1). We chose to focus on the most visible one, cross contamination, which had its origin in the Anti-Correlated Noise Reduction (ACNR), due to poor filtering behavior near edges of the tissue image. The ACNR algorithm indirectly denoises the bone image by filtering the tissue image, which is a useful alternative to direct methods in a high noise scenario like DXA.

Chapter 2 was entirely dedicated to the improvement of the ACNR using edge-preserving filters and the choice of adaptive parameters. Our two main denoising alternatives, bilateral filtering and anisotropic diffusion, had very comparable behaviors in the end. Both managed to greatly reduce the noise while preventing the apparition of cross contamination on vertebrae. All observers taking part in our experiment confirmed with a high level of confidence that our algorithm reduced the number and severity of these artifacts, which validates the use of such filters. This answers the question of the interest and applicability of these filters in the specific context of ACNR for DXA imaging. However, there remains evidences of cross contamination in the background of the images that we did not manage to remove. Because of high variability of noise and image tissue structures, adapting parameters to leverage the trade-off between noise, contamination and texture introduced by filtering proved to be a very difficult task. Another difficulty was the choice of a suitable metric to measure cross contamination, and this reflected in the behavior of anisotropic diffusion when using it as a stopping criterion. Because the two filters acted similarly, we concluded that the bilateral filter remained a better choice as it is often easier to tune than diffusion, although the iterative nature of the
latter allows to refine parameters step-by-step.

In Chapter 3, we experimented on a fully automatic method for vertebrae segmentation in DXA, in an attempt to improve the rather coarse current six-points model. To this end, we implemented convolutional neural networks in a U-Net architecture, suited to the problem of pixel-wise prediction. Experimentations on depth showed that shallow networks could not produce useful segmentation results, while deeper nets like U-Net 4 provided accurate predictions, due to more parameters and a larger receptive field. We had limited training data (44 iDXA images with ground truth annotated by ourselves) but obtained a 0.94 mean Dice by using 36 images only, which showed that for our specific problem, training end-to-end was possible and that there was no need for transfer learning. For U-Net 4, we investigated the variation of metrics with respect to the amount of training data: this revealed that the performances quickly saturated while adding more images (+ 0.01 Dice only from 18 to 36 training images). Overall, very few training data was needed in our specific case, provided we used data augmentation and sampled enough patches per image. The reason is that our task is a two-classes only prediction task, with objects whose appearance do not vary much between instances (vertebrae are always seen in the same plane and have similar shapes). The next step in this study would be to test the model on cases with heavy fractures to see whether it needs to be retrained with such cases. Because fracture cases often lack in medical databases, it would also be interesting to see if one can efficiently simulate fractured vertebrae to use for training by data augmentation (for instance by applying elastic transformations to patches of healthy vertebrae).

We also tested our network on another database, Prodigy. Indeed, in a real world scenario, it would be interesting to have a cross-platform system which does not need retraining the model with specific data to account for variability between different acquisition devices. Qualitative results demonstrated that similar performances were obtained for the lumbar vertebrae, after up-sampling the low resolution Prodigy images to match the scale of iDXA. This could be applied the same way to other images with different resolutions. However, we encountered limitations of this method due to the intrinsic low resolution of Prodigy for thoracic vertebrae, whose boundaries are blurry and ill-
defined and could not be identified accurately by the network. This application remains constrained to dual energy systems only, for which a bone image with no organs is available. For tomography or MRI, the network would probably be fooled in dense tissue areas and need specific training.

We observed some cases of segmentation failure due to cross contamination for Prodigy. In these cases, artifacts in the form of dark lines superimposed on vertebrae modified the predicted vertebra outline or introduced an non-existent concavity, leading to under-evaluated vertebra heights. For these cases, using our bilateral filter or anisotropic diffusion greatly removed the artifacts and an accurate segmentation could be recovered.

To conclude, the qualitative comparison with GE’s software enCORE showed promising results: neural networks demonstrate the ability to account for slight curvatures of the vertebrae horizontal endplates, whereas the geometric lines computed with enCORE sometimes lose relevant information. Because fracture assessment is made on the basis of height ratios at three points of the vertebra endplates in Genant’s classification, this could possibly modify the diagnosis in some cases.
Bibliography


Appendix: description of filters

Linear filters

Let us denote by \( x \) a pixel in image \( I \), \( y \) a nearby pixel within the neighborhood \( \Omega \) (centered around \( x \)) and \( H \) is the kernel function of the filter. Then, the equation of the linear filter is:

\[
I_{\text{filtered}}(x) = (I * H)(x) = \sum_{y \in \Omega} I(x - y)H(y) \tag{A.1}
\]

For a 3x3 neighborhood, the kernel for a mean filter is given by:

\[
H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{A.2}
\]

For a Gaussian filter, the impulse response is a gaussian function:

\[
H(y) = G_\sigma(y) = e^{-\frac{\|y\|^2}{2\sigma^2}} \tag{A.3}
\]

Here, \( \sigma \) is the standard deviation of the Gaussian. The larger \( \sigma \) gets, the more smoothing will be done in the output image, resulting in a blurry effect. The Gaussian function does not have a finite support, but in practice we use a truncated Gaussian in the range \([-3\sigma; 3\sigma] \).

Both these filters’ kernels are matrices which can be expressed as the outer product of two vectors. As a result, the 2D convolution can be split up in two 1D convolutions: we say that the kernel is separable. This is computationally convenient for large images and large kernels, because the complexity of the filter reduces to \( O(2N) \) instead of \( O(N^2) \) for a kernel of size \( N \).
Bilateral filter

With the same notations, the bilateral filter equations read:

$$I_{filtered}(x) = \frac{\sum_{y \in \Omega} I(y).G_{\sigma_s}(\|x - y\|)G_{\sigma_r}(I(x) - I(y))}{\sum_{y \in \Omega} G_{\sigma_s}(\|x - y\|)G_{\sigma_r}(I(x) - I(y))} \quad (A.4)$$

It depends on two user-defined parameters: $\sigma_s$ and $\sigma_r$, standard deviations of the spatial and intensity gaussians respectively. Since we have to compute the intensity differences for each pixel $y$ in $\Omega$ for each pixel $x$, the brute implementation of a bilateral filter can be quite time-consuming. Hence, this neighborhood is often reduced to a few times $\sigma_s$.

If $\sigma_r$ is small, the intensity gaussian will be very narrow and will quickly reduce weights as $I(y)$ diverges from $I(x)$. At the opposite, if $\sigma_r$ is high, the intensity gaussian will not be selective and the bilateral filter will tend to behave like a simple gaussian filter. Although $\sigma_s$ is often taken quite small and does not impact too much the results, the choice of $\sigma_r$ is of crucial importance, as it determines the level of contrast that we want to preserve. There is no universal strategy for the choice of this parameter, but it is often suitable to tune $\sigma_r$ starting from the noise’s standard deviation if it can be accessed.

Similarly, the equation for the joint bilateral filter reads:

$$I_{filtered}(x) = \frac{\sum_{y \in \Omega} I(y).G_{\sigma_s}(\|x - y\|)G_{\sigma_r}(J(x) - J(y))}{\sum_{y \in \Omega} G_{\sigma_s}(\|x - y\|)G_{\sigma_r}(J(x) - J(y))} \quad (A.5)$$

with $J$ the joint image used to compute the weights. In our denoising application, the joint image is a gaussian or median-filtered version of the image to denoise.

For the adaptive bilateral filter, the equation remains the same, but with $\sigma_s(x)$ and $\sigma_r(x)$, functions of the spatial position in the image.
Nonlinear Isotropic Diffusion (Perona-Malik Diffusion)

The equation of the Perona-Malik diffusion is as follows:

\[
\frac{\partial I}{\partial t} = \text{div}(D(x, t).\nabla I) = \nabla D.\nabla I + D(x, t).\Delta I
\]  

(A.6)

with \(\text{div}\) the divergence operator, \(\nabla\) the gradient, \(\Delta\) the Laplacian and \(I\) the original image on which the diffusion starts. The diffusivity coefficient is usually chosen as:

\[
D(\|\nabla I\|) = e^{-\left(\frac{\|\nabla I\|}{C}\right)^2}
\]  

(A.7)

where the parameter \(C\) is a constant which tunes the sensitivity to edges.

This diffusion equation has several nice properties. First, because diffusion is a conservative process, the average value of gray levels is conserved within the image. Second, it does not introduce any new maxima or minima that did not exist in the original image.

In practice, the diffusion equation cannot be used as such: noise in the original image distorts the computation of gradients, and noisy pixels are interpreted as edges which stops the diffusion nearby. This makes the diffusion process instable, but we can address the problem using a regularized version of the equation, in which the local diffusivity coefficient is computed on a smoothed image with less noise. The regularized partial derivative equation of the Perona-Malik model thus becomes:

\[
\frac{\partial I}{\partial t} = \text{div}(D(\|\nabla I_\sigma\|).\nabla I)
\]  

(A.8)

where \(I_\sigma = I * G_\sigma\) denotes the smoothed image resulting from the convolution of \(I\) with the Gaussian function of standard deviation \(\sigma\). Of course, regularization brings robustness to noise but at the unavoidable cost of some blurring, which is why the parameter \(\sigma\) has to be tuned carefully to solve the trade-off between noise level and blur. A common strategy is to estimate the standard deviation of the noise in the image and to set \(\sigma\) equal to it.
Nonlinear Anisotropic Diffusion (Weickert Diffusion)

In Weickert’s model, the diffusivity is no longer a scalar but a matrix, built on each image point from the so-called structure tensor which gathers local gradient information. The structure tensor is defined by:

\[ J_{\rho}(\nabla I_\sigma) = G_\rho \ast \left( \nabla I_\sigma \nabla I_\sigma^T \right) \quad (A.9) \]

In the above equation, \( (\nabla I_\sigma \nabla I_\sigma^T) \) is the 2x2 Hessian matrix computed on the considered pixel of image \( I \) after smoothing the gradients with a Gaussian \( G_\sigma \) for regularization (like in the Perona-Malik model). The Hessian matrix is convolved element-wise with a Gaussian \( G_\rho \), which has for effect to average the gradient orientations over a scale \( \rho \).

The eigenvector of \( J_\rho \) with the largest eigenvalue gives an average yet accurate direction of smoothed gradients in \( I \), and the corresponding eigenvalue \( \mu_1 \) gives the intensity variation along the gradient direction. Since the matrix is symmetric, its second eigenvector is orthogonal to the first one, and the same way the smallest eigenvalue \( \mu_2 \) indicates the intensity variation in the direction orthogonal to gradient. Both eigenvalues are positive, and they are relatively close to each other in uniform regions but different near strong edges. This allows Weickert to define a measure of local coherence, \( \kappa = (\mu_1 - \mu_2)^2 \), which he uses to compute the diffusivity matrix. This matrix is set to have the same eigenvectors as the structure tensor, but with modified eigenvalues \( \lambda_1 \) and \( \lambda_2 \). So the diffusion matrix can be written:

\[ D = U_\theta \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U_{-\theta} \quad (A.10) \]

with \( U_\theta \) the rotation matrix defining the gradient orientation \( \theta \).

Different choices for \( \lambda_1 \) and \( \lambda_2 \) give different types of anisotropic filters, depending on the application. One can cite the Edge-Enhancing Diffusion (EED) and the Coherence-Enhancing Diffusion (CED). For Coherence-Enhancing Diffusion [41], the eigenvalues of the diffusivity matrix are given by \( \lambda_1 = \alpha \) and:

\[ \lambda_2(\kappa) = \begin{cases} 
\alpha, & \text{if } \kappa = 0 \\
\alpha + (1 - \alpha)e^{-\frac{\kappa}{\kappa}}, & \text{otherwise}
\end{cases} \quad (A.11) \]
\( \alpha \) is a small parameter typically set to \( \alpha = 0.001 \) to prevent diffusion across edges (it must remain positive so that the matrix be well-conditioned). Close to edges, the coherence \( \kappa \) is high so the exponential term approaches 1 and \( \lambda_2 \) approaches 1 as well, leading to a strong diffusion along the structure. Away from edges, \( \kappa \) is small so the exponential term approaches 0 and \( \lambda_2 \) approaches \( \alpha \), thus reducing the diffusion.

Coherence-Enhancing Diffusion has been widely used to enhance thin flow-like structures without degradation of edges. A famous application of CED is edge linking in noisy images of fingerprints. The filter manages to smooth along the fingerprints lines and fill the gaps, without blurring between lines.