

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE**  
**SCIENTIFIQUE**  
**UNIVERSITE MOHAMED BOUDIAF - M'SILA**

FACULTE : SCIENCE ET TECHNOLOGIE

DEPARTEMENT : ELECTRONIQUE

N° : .....



FILIERE : ELECTRONIQUE

OPTION : ELECTRONIQUE DES  
SYSTEMES EMBARQUES

*MEMOIRE PRESENTE POUR L'OBTENTION*

*DU DIPLOME DE MASTER ACADEMIQUE*

**Par:**

*Asli khalil  
Haouas Mounir*

**THEME**

**Traitement d'image médicale par python  
sur la carte FPGA**

**Soutenu devant le jury composé de :**

Dr. AHMED BOUCHELGHEM

Université M<sup>ed</sup> Boudiaf –M'sila Président

Dr. ABDELOUAHAB BENHAMADOUCHE

Université M<sup>ed</sup> Boudiaf –M'sila Rapporteur

Dr. LAHOUAOUI LALAOUI

Université M<sup>ed</sup> Boudiaf –M'sila Examineur

**Année universitaire : 2021 /2022**



## REMERCIEMENT



*Nous voudrions d'abord remercier le dieu le plus puissant de nous avoir donné la force de faire cet humble travail de licence*

*Nous remercions sincèrement notre promoteur.*

**M. ABDELOUAHAB BENHAMADOUCHE**

*est en charge du projet.*

*Nous tenons également à remercier les professeurs qui nous ont aidés pour la partie programmation.*

*Je tiens à remercier les membres du jury pour l'honneur qu'ils nous ont fait en acceptant la tâche de juger ce travail.*

*Nous tenons particulièrement à exprimer nos sincères remerciements à nos parents qui nous ont soutenus tout au long de ce travail.*

*Enfin, nous n'aimons pas non plus oublier tous mes amis pour leur aide et leurs conseils.*



DEDICACE



Nous dédions ce modeste travail à :  
Nos mères, sources de tendresse et d'amours  
pour leurs soutiens tout le long de notre vie  
scolaire.

Nos pères, qui nous ont toujours soutenus et  
qui ont fait tout possible pour nous aider.

Nos frères et nos soeurs, que nous aimons  
beaucoup.

Nos grandes familles.

Nos chers amis et enseignants.

Tous ceux qui ont collaboré de près ou de loin  
à la réalisation de ce travail.

## Résumé

Les images sont de plus en plus utilisées dans de nombreuses disciplines (télédétection, médecine...etc.) car elles fournissent des informations précieuses sur les scènes ou les objets filmés. En médecine, depuis quelques années, les images sont devenues des éléments indispensables à l'établissement d'un bon diagnostic.

Quelquefois, il apparaît qu'une image brute ne fournit pas tous les détails nécessaires à une bonne analyse, elle doit être retouchée de diverses manières : c'est le traitement des images .

Afin d'accélérer le travail des radiologues dans l'analyse des images, nous avons recours au calcul parallèle. Les FPGA sont des circuits électroniques parallèles qui permettent aujourd'hui de développer des applications de plus en plus performantes en vitesse d'exécution et gourmandes en ressources matérielles. Dans le cadre de notre travail de recherche, nous mettons en œuvre, sur FPGA, quelques algorithmes de traitement d'images par le langage de programmation que nous utilisons dans notre étude est python

**mots clés :** traitement d'image . FPGA.PYNQ.

## Abstract

Images are increasingly used in many disciplines (remote sensing, medicine, etc.) because they provide valuable information on the scenes or objects filmed. In medicine, for several years now, images have become essential elements in establishing a good diagnosis.

Sometimes, it appears that a raw image does not provide all the details necessary for a good analysis, it must be retouched in various ways: this is image processing.

In order to speed up the work of radiologists in image analysis, we use parallel computing. FPGAs are parallel electronic circuits that today make it possible to develop increasingly high-performance applications in terms of execution speed and greedy in terms of hardware resources. As part of our research work, we implement, on FPGA, some image processing algorithms by the programming language we use in our study is python

**keywords:** image processing. FPGA.PYNQ.

## الملخص

يتم استخدام الصور بشكل متزايد في العديد من التخصصات (الاستشعار عن بعد ، والطب ، وما إلى ذلك) لأنها توفر معلومات قيمة عن المشاهد أو الأشياء التي تم تصويرها. في الطب ، منذ عدة سنوات حتى الآن ، أصبحت الصور عناصر أساسية في إنشاء تشخيص جيد.

في بعض الأحيان ، يبدو أن الصورة الأولية لا توفر كل التفاصيل اللازمة لتحليل جيد ، يجب تنقيحها بطرق مختلفة: هذه معالجة للصور.

من أجل تسريع عمل اختصاصي الأشعة في تحليل الصور ، نستخدم الحوسبة المتوازية. FPGA عبارة عن دوائر إلكترونية متوازية تجعل من الممكن اليوم تطوير تطبيقات عالية الأداء بشكل متزايد من حيث سرعة التنفيذ والجشع من حيث موارد الأجهزة. كجزء من عملنا البحثي ، نفذ ، على FPGA ، بعض خوارزميات معالجة الصور بواسطة لغة البرمجة التي

نستخدمها في دراستنا هي Python

**الكلمات الرئيسية:** معالجة الصور. FPGA.PYNQ.

## Table des matières

### Chapitre 1 : Généralités sur le traitement d'images et le traitement d'images médicales

1	Introduction.....	2
2	Traitement d'image.....	2
2.1	Définition du traitement d'image.....	2
2.2	Les images numériques.....	3
2.3	Les caractéristiques d'une image numérique.....	3
2.3.1	Le pixel.....	3
2.3.2	Codage d'une image.....	4
2.3.3	La taille d'une image.....	6
2.3.4	Résolution.....	7
2.3.5	Bruit.....	7
2.3.6	La texture.....	8
2.3.7	Histogramme.....	8
2.3.8	Luminance.....	8
2.3.9	Contraste.....	8
3	Imagerie médicale.....	9
3.1	Définition de l'imagerie médicale.....	9
3.2	Types d'imagerie médicale.....	10
3.3	Radiographie.....	11
3.4	Le scanner.....	12
3.4.1	Le principe général du scanner à rayon x.....	13
3.5	L'échographie.....	13
3.6	L'Imagerie par résonance magnétique (IRM).....	14
4	Architecture générale d'un système de traitement d'images.....	15
4.1	Acquisition.....	16
4.2	Pré-traitement et post-traitement.....	16
4.3	Le traitement numérique.....	16
4.4	Visualisation, transmission et stockage.....	17
5	Conclusion.....	17

### Chapitre 2 : Généralités sur les cartes FPGA PYNQ

1	Introduction.....	18
2	FPGA.....	18
2.1	Domaines d'application.....	19
2.2	Les circuits FPGA.....	20
2.2.1	Technologie FPGA Xilinx.....	20
2.2.2	Les cellules logiques d'un FPGA.....	21
2.3	Flot de conception classique des circuits FPGA.....	21
2.3.1	L'entrée de la conception.....	22
2.3.2	La synthèse.....	23
2.3.3	L'implémentation de la conception.....	23
3	Les SOC FPGA.....	23
3.1	Les FPGA Xilinx Zynq-7000.....	24
3.1.1	Le PYNQ.....	25
3.1.2	L'Architecture PYNQ.....	25
4	Conclusion.....	26

### Chapitre 3 : Application au traitement d'image médicale avec Python

1	Introduction.....	27
2	Présentation du langage de programmation Python.....	27
2.1	Techniques de production des programmes.....	27
3	Programmer avec python.....	28
3.1	Installation de Python .....	28
3.2	Lancement de l'interpréteur Python 3.10.4.....	29
3.3	Environnement de développement intégré (IDE) : PyCharm.....	30
3.4	L'outil de programmation Anaconda .....	31
3.4.1	L'interface graphique de Anaconda Navigator.....	31
3.4.2	Jupyter .....	32
3.4.3	Les avantages de Jupyter.....	33
4	Application au traitement d'image TDM thoracique .....	33
4.1	Les type de segmentation d'image médicale .....	33
4.2	Introduction aux image TDM.....	34
4.2.1	Intensités CT et unités Hounsfield.....	35
4.2.2	Visualisation des données TDM .....	37
4.2.3	Base de données utilisée .....	37
4.3	Visualisation des données de scanner TDM : niveau et fenêtrage .....	38
4.4	Application pour le calcul de la surface des poumons sur des images TDM ....	39
4.4.1	Etape 1 : binarisation des images et seuillage des intensités. ....	39
4.4.2	Etape 2 : Recherche de contours.....	41
4.4.3	Etape 3 : Trouver la zone pulmonaire à partir d'un ensemble des contours possibles .....	41
4.4.4	Etape 4 : Extraction du masque binaire à partir du contour des poumons. ....	41
4.4.5	Etape 5 : trouver la dimension de pixel pour le calcul de la surface en mm <sup>2</sup> ...	42
5	Conclusion .....	45

### Chapitre 4 : Implémentation et résultats expérimentaux

1	Introduction.....	36
2	Utilisation de la carte Pynq-Z2 .....	36
2.1	Prérequis d'utilisation .....	36
2.2	Configuration de carte.....	37
2.3	Mise sous tension de la carte Pynq-Z2.....	37
2.4	Connexion réseau .....	38
2.4.1	Ethernet.....	38
2.4.2	Connecter la carte à un ordinateur .....	38
2.4.3	Connexion à un routeur réseau .....	38
2.4.4	Connexion à Jupyter Notebook .....	39
2.5	Exemples des notebooks.....	39
3	Conclusion .....	40

## Liste des figures :

### Chapitre 1

Fig.1 Représentation numérique d'une image.....	3
Fig.2 Représentation des pixels dans une image. ....	4
Fig.3 Décomposition d'une image en 3 couches.....	5
Fig.4 Représentation d'une image bruitée.....	7
Fig.5 Histogramme et palette associés à une image.....	8
Fig.6 Image médicale - représentation 3d.....	9
Fig.7 L'imagerie médicale par modalité et par axe thérapeutiq .....	10
Fig.8 Principe de la radiologie .....	11
Fig.9 Formation de l'image en rayon X en mode analogique (A) et numérique (B).....	12
Fig.10 Appareil de radiologie de type scanner .....	12
Fig.11 Schéma de principe général de scanner. ....	13
Fig.12 Eléments fonctionnels d'un appareil d'échographie médical .....	14
Fig.13 Eléments de l'imagerie par résonance magnétique .....	15
Fig.14 Schéma synoptique d'un système de traitement d'image .....	16

### Chapitre 2

Fig.1 exemple de version carte FPGA.....	18
Fig.2 Exemples de circuit FPGA .....	19
Fig.3 Architecture de principe d'un FPGA SRAM.....	20
Fig.4 Architecture de Principe d'un FPGA SRAM .....	21
Fig.5 structure d'un CLB. ....	21
Fig.6 Flux de conception RTL pour FPGA.....	22
Fig.7 Processus de démarrage Xilinx Zynq-7000 .....	25
Fig.8 L'architecture PYNQ.....	26

### Chapitre 3

Fig.1 Chaîne de compilation.....	28
Fig.2 Chaîne d'interprétation .....	28
Fig.3 Interprétation du bytecode compilé .....	28
Fig.4 Site web <a href="http://www.python.org/downloads/">http://www.python.org/downloads/</a> .....	29
Fig.5 Interface de IDLE (Python GUI).....	29
Fig.6 interface de PyCharm .....	30
Fig.7 interface anaconda .....	31
Fig.8 interface Jupyter .....	32
Fig.9 Exemple d'image d'une TDM d'un crâne humain .....	35

Fig.10 Echelle de Hounsfield .....	36
Fig.11 Fenêtrage et mise à niveau de l'échelle Hounsfield.....	37
Fig.12 Application du fenêtrage et mise à niveau avec l'échelle de Hounsfield sur 3 échantillons d'image de scanner thoracique.....	39
Fig.13 Application de binarisation d'image.....	40
Fig.14 recherche des contours et les contours de poumons et l'affichage des masques et des calculs la surface.....	44

#### **Chapitre 4**

Fig.1. Schéma représentative de la carte Pynq-Z2.....	48
Fig.2. Fenêtre du navigateur avec Jupyter après connexion à la carte.....	50
Fig.3. Dossier des exemples de notebook.....	51

#### **Liste des tableaux**

Tableau 1 Profondeur du codage .....	6
Tableau 2 Exemple des différentes tailles que peut avoir une image .....	7
Tableau 3 valeurs HU approximatives pour les tissus couramment trouvés sur les tomodensitogrammes.....	36

# Introduction générale

Les images sont de plus en plus utilisées dans de nombreuses disciplines (télédétection, médecine, etc.) car elles fournissent des informations précieuses sur les scènes ou les objets filmés. Le traitement d'images est un terme employé pour décrire des opérations effectuées sur des images, dans le but d'accomplir un certain objectif. Il est devenu fortement lié à l'objectif limité de modification des images tels que : corriger les erreurs présentées pendant l'acquisition ou la transmission ; ou augmenter pour surmonter la faiblesse du système visuel humain. Donc, la discipline du traitement d'images peut être brièvement récapitulée comme étant concerné par un processus qui prend une image entrée et produit une image modifiée comme résultat

En médecine, depuis quelques années, les images sont devenues des éléments indispensables à l'établissement d'un bon diagnostic. L'imagerie médicale est l'un des domaines de la médecine qui est plus progressé dans les dernières décennies.

Il existe actuellement diverses techniques médicales pour examiner le corps, analyser les organes internes et distinguer les processus pathologiques sans intervention chirurgicale. Les images médicales montrent des anomalies du corps humain que le médecin ne peut pas voir et facilitent ainsi le diagnostic. Le médecin peut également contacter des spécialistes dans des centres éloignés en envoyant des photos pour aider à diagnostiquer et à traiter le patient. Ces images sont généralement unidimensionnelles (1D), bidimensionnelles (2D), tridimensionnelles (3D) ou quadridimensionnelles (4D = 3D + temps).

Dans ce travail nous allons utiliser la puissance de calcul des FPGA ainsi que la puissance du langage Python qui dispose d'un ensemble important de bibliothèque de traitement d'image. Dans le cadre de notre travail, nous mettons en œuvre, sur FPGA, quelques programmes de traitement d'images qui permettront d'extraire de informations utiles à partir d'images de scanner thoracique.

Ce mémoire est organisé en 4 chapitres :

- Le premier chapitre présentera le traitement d'image et l'imagerie médicale.
- Le deuxième chapitre présentera des généralités sur les circuits FPGA en particulier sur les cartes FPGA de type Xilinx Zynq-7000.
- Dans le troisième chapitre nous allons décrire le programme Python qui permettra de traité des images de scanner thoracique.
- Dans le dernier chapitre, nous allons présenter les résultats du traitement des images sur FPGA.
- Nous terminerons ce mémoire par une conclusion générale.

# Chapitre 1

## Généralités sur le traitement d'images et le traitement d'images médicales

# 1 Introduction

Le traitement d'image est une discipline relativement jeune en pleine expansion, donnant ainsi lieu chaque année à une multitude de travaux académiques, dans tous les domaines technologiques, industriels, médical etc. Par traitement d'images, on désigne l'ensemble des opérations appliquées sur les images numériques transformant une image en une autre, ou en une primitive formelle, ceci afin de tirer des informations utiles des images traitées.

Ce chapitre a pour objectif de faire une introduction au domaine du traitement d'images et surtout au traitement d'images médicales. Dans ce chapitre nous allons ainsi voir les différentes techniques d'acquisition des images médicales, ainsi que leurs objectifs et que leur domaine d'utilisation.

## 2 Traitement d'image

### 2.1 Définition du traitement d'image

Le traitement d'images désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées. [1]

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant :

- ✓ La résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions,
- ✓ Les réglages optiques utilisés qui déterminent par exemple la netteté de l'image,
- ✓ Les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées,
- ✓ Le bruit de la chaîne de transmission d'image.

Quelques exemples d'informations qu'il est possible d'obtenir d'une image numérique :

- ✓ La luminance moyenne
- ✓ Le contraste moyen
- ✓ La couleur prédominante
- ✓ Le taux d'acuité moyen (précis ou flou)
- ✓ Le taux d'uniformité des couleurs
- ✓ La présence ou l'absence de certains objets

## 2.2 Les images numériques

Le mot "image" découlant du latin " imago" désigne la représentation visuelle d'un objet par différents supports (dessin, photographie, peinture...). L'image d'un objet peut alors être interprétée comme un ensemble de points sur face plane. [2]

Une image numérique est représentée sous forme matricielle (Figure I.1) qui regroupe un ensemble de pixels. Pour représenter (afficher) une image réelle par un outil informatique (ordinateur...) l'image a besoin d'une numérisation, Chaque pixel ayant une couleur spécifique qui sera codée par un nombre binaire. On désigne donc sous le terme d'image numérique toute image acquise, créée, traitée ou stockée sous forme binaire. Le stockage de l'image numérique ainsi obtenue peut s'effectuer sous différents formats (jpeg, bmp, tiff, png, gif...).[1]

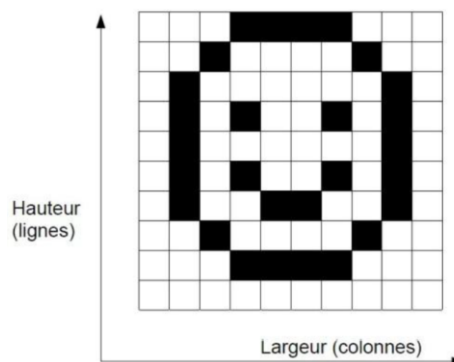


Fig.1 Représentation numérique d'une image.

## 2.3 Les caractéristiques d'une image numérique

### 2.3.1 Le pixel

Le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels

d'affichage ou d'impression. L'image peut être affichée comme un groupe de pixels tel que montré sur la figure ci-dessous :

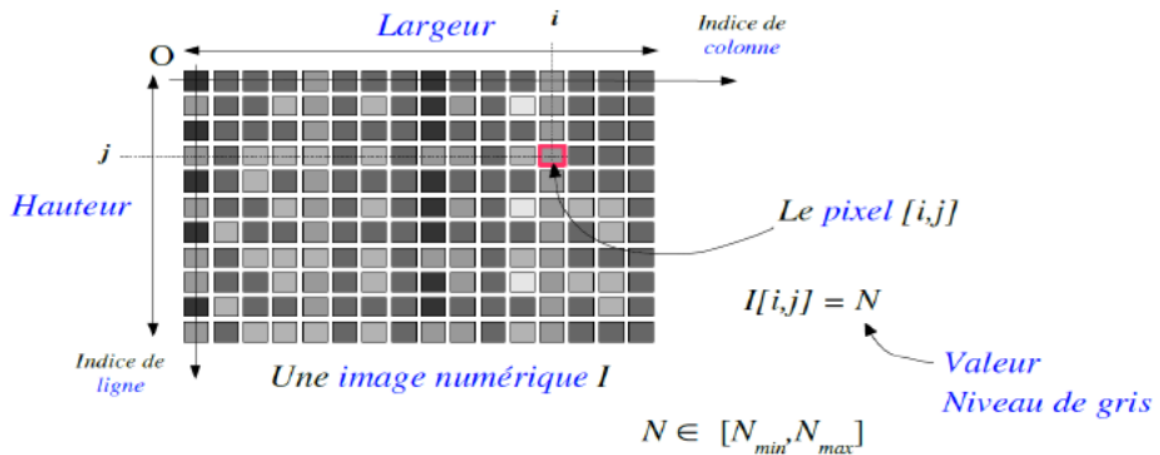


Fig.2 Représentation des pixels dans une image.

Une image numérique représentée par une matrice de dimension  $(m \times n)$  d'éléments matériel qui sont des pixels. La valeur de chaque pixel représente une couleur considérée dans l'image.

### 2.3.2 Codage d'une image

Les images fournies par les appareils photo sont généralement en couleur. Une image en couleur est constituée de trois couches : une couche rouge (R), une couche verte (V), une couche bleue (B).

Soit  $N_x$  le nombre de colonnes de l'image et  $N_y$  le nombre de lignes. Le nombre de pixels total est  $N = N_x \times N_y$ . Chaque couche est une matrice comportant  $N_y$  lignes et  $N_x$  colonnes. Le plus souvent, cette matrice contient des entiers codés sur 8 bits (les valeurs vont de 0 à 255). Pour l'image en couleur complète, il y a donc 24 bits par pixels, à multiplier par le nombre de pixels pour obtenir l'occupation totale en mémoire. Chaque couche peut être vue comme une image en niveaux de gris.[3]

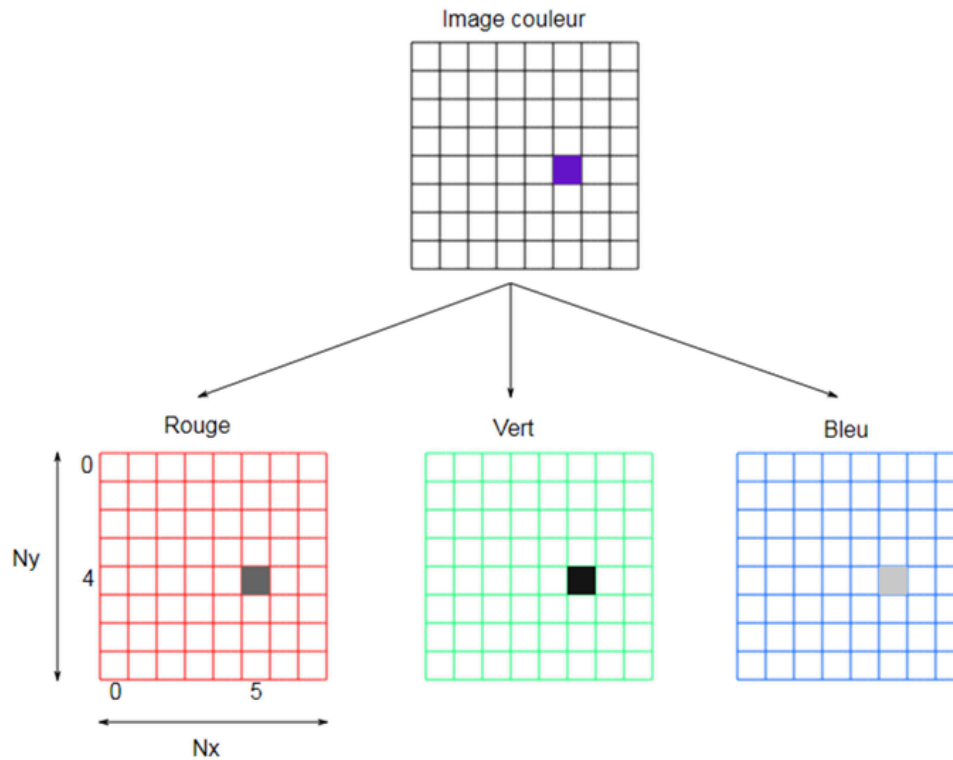


Fig.3 Décomposition d'une image en 3 couches.

On voit sur la figure le pixel (5,4) dont les niveaux de gris des trois couches sont (100,20,200), ce qui donne une couleur violette. Par convention, on notera en premier l'indice qui repère les colonnes, conformément au repérage des coordonnées d'un point sur un plan (x,y). On remarque néanmoins que l'origine se trouve sur le coin supérieur gauche de l'image. Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur). (Voir tableau 1)

**Tableau 1.** Profondeur du codage

Niveau	Couleurs
1 bit	2 (noir et blanc)
4 bits	16 couleurs : 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes
8 bits	256 couleurs ou nuances de gris : 256 dégradés de gris allant du noir au blanc ou bien 256 couleurs différentes
16 bits	65 536 couleurs : 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes
24 bits ou vraies couleurs	16 777 216 couleurs : Cette représentation permet de représenter une image en définissant chacune des composantes (RGB, pour rouge, vert et bleu). Chaque pixel est représenté par un entier comportant les trois composantes, chacune codée sur un octet, c'est-à-dire au total 24 bits
30 bits	1 073 741 824 couleurs

### 2.3.3 La taille d'une image

La taille d'une image numérique peut se définir par :

- ✓ Sa définition en pixels (ex : 640x480 pixels)
- ✓ Ses dimensions en pouces (Ex : 12")
- ✓ Sa résolution en dpi ou ppp. (Ex : 300dpi)

Ces 3 informations sont liées. Si vous en connaissez 2, vous pouvez calculer la troisième à partir des formules suivantes [4] :

- ✓ Résolution = définition / dimension
- ✓ Définition = résolution x dimension
- ✓ Dimension = résolution / définition

Voici quelques exemples (en considérant que l'image n'est pas compressée) :

**Tableau 2.** Exemple des différentes tailles que peut avoir une image

<b>Définition de l'image</b>	<b>Noir et blanc (1 bit / pixel)</b>	<b>256 couleurs ou nuances de gris (8 bits / pixel)</b>	<b>65000 couleurs (16 bits / pixel)</b>	<b>Couleurs réelles 16 000 000 couleurs (24 bits / pixel)</b>
<b>320x200</b>	7.8 Ko	62.5 Ko	125 Ko	187.5 Ko
<b>640x480</b>	37.5 Ko	300 Ko	600 Ko	900 Ko
<b>800x600</b>	58.6 Ko	468.7 Ko	937.5 Ko	1.4 Mo
<b>1024x768</b>	96 Ko	768 Ko	1.5 Mo	2.3 Mo

### 2.3.4 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution.[5]

### 2.3.5 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. [6]



*Fig.4 Représentation d'une image bruitée*

### 2.3.6 La texture

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels [7]

### 2.3.7 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant [5]

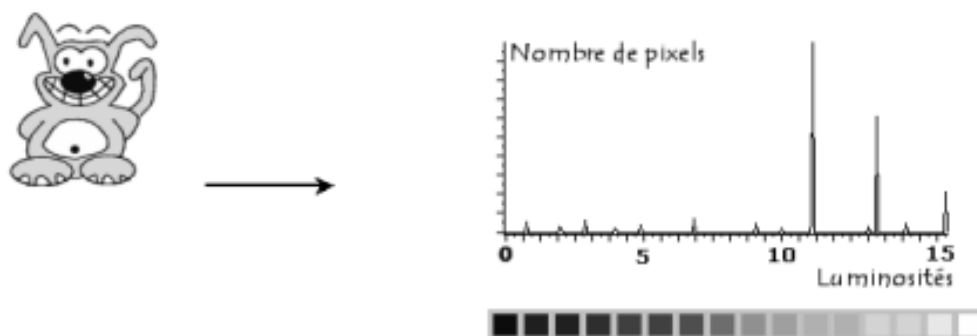


Fig.5 Histogramme et palette associés à une image

### 2.3.8 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain.

### 2.3.9 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des

luminances de deux zones d'images. Si  $L_1$  et  $L_2$  sont les degrés de luminosité respectivement de deux zones voisines  $A_1$  et  $A_2$  d'une image, le contraste  $C$  est défini par le rapport : [6]

$$c = \frac{l_1 + l_2}{A_1 + A_2}$$

### 3 Imagerie médicale

#### 3.1 Définition de l'imagerie médicale

L'imagerie médicale constitue aujourd'hui un outil extraordinaire d'aide au diagnostic pour le médecin. Les modalités d'acquisition de l'information 'image' sont nombreuses et la quantité d'images manipulées s'accroît. La radiographie par rayon X constitue la part la plus importante du marché, l'imagerie ultrasonore, rapide, non invasive et peu onéreuse, gagne du terrain. Cependant, l'Imagerie par Résonance Magnétique (IRM), la tomographie de même que la radiologie RX ont un avantage sur l'imagerie ultrasonore pour la qualité des images.

Une image médicale peut être définie comme la représentation des variations spatiales et temporelles d'une ou de plusieurs propriétés physiques d'un sujet sous investigation. On parle d'images statiques lorsque la variation temporelle de la propriété n'est pas prise en considération, autrement on parle d'images dynamiques.[7]

L'image médicale est une représentation discrète d'une partie du corps qui est décrite par une matrice à 3 dimensions, comme la montre la figure

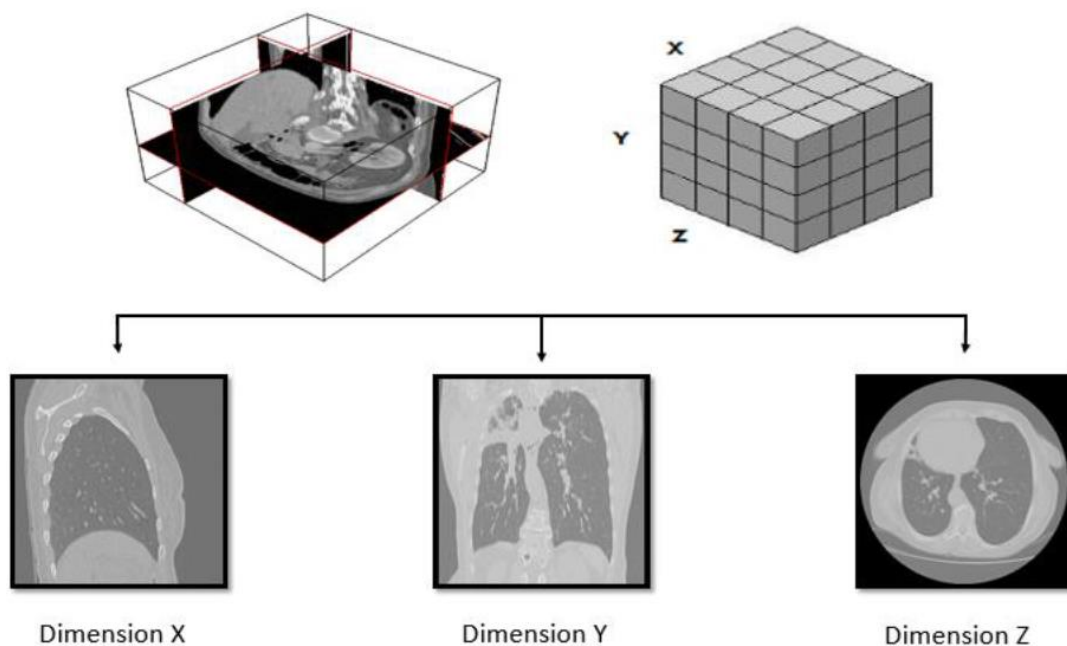


Fig.6 Image médicale - représentation 3d

### 3.2 Types d'imagerie médicale

Aujourd'hui, l'imagerie médicale se construit principalement autour de six éléments technologiques :

- Les traceurs (agents de contrastes et radiopharmaceutiques) ;
- La génération du signal ;
- Le traitement du signal ;
- La visualisation de l'image ;
- L'archivage, la communication et le stockage des images ;
- La sécurité et le confort du patient.

L'arsenal de techniques, toutes complémentaires, autorise une exploration ciblée pour différentes finalités :

- **L'imagerie structurelle** recueille des informations sur la morphologie des organes, tissus ou cellules, comme la taille, le volume, la localisation, ou encore la forme d'une éventuelle lésion, de la zone étudiée (IRM, TDM, RX...) ;
- **L'imagerie fonctionnelle** encore appelée imagerie métabolique et moléculaire, vise à étudier le fonctionnement et l'activité physiologique des organes, tissus ou cellules ;
- **L'imagerie interventionnelle** correspond à l'ensemble des actes médicaux réalisés sous le contrôle de dispositifs d'imagerie, comme le repérage, le guidage, le contrôle optimal du geste médical ; [8]

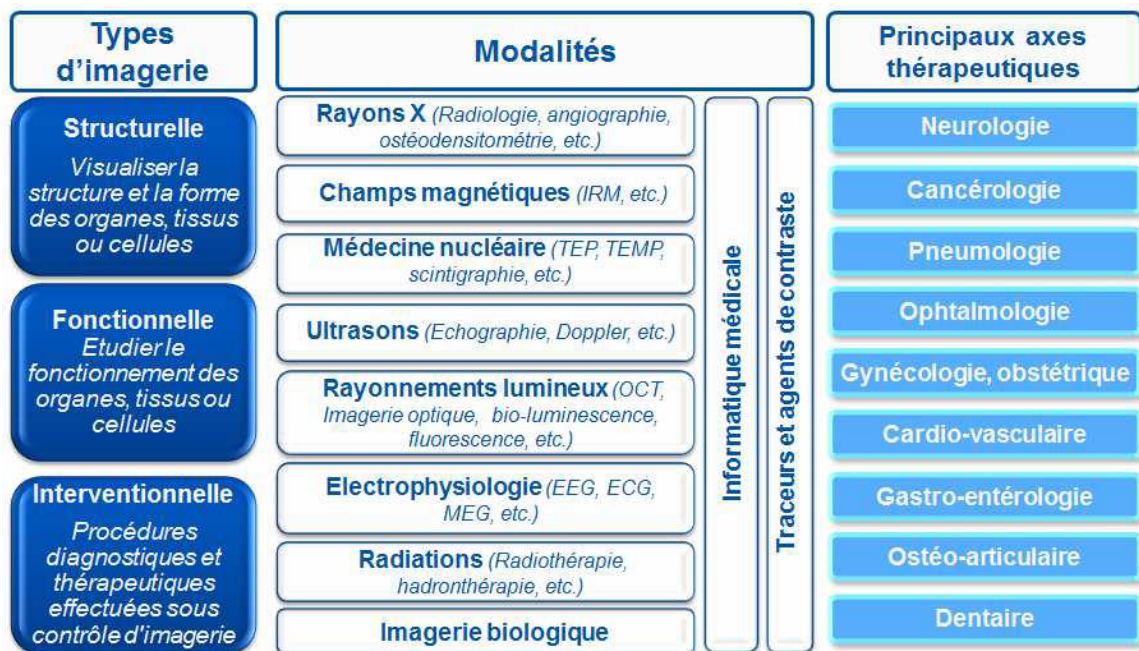
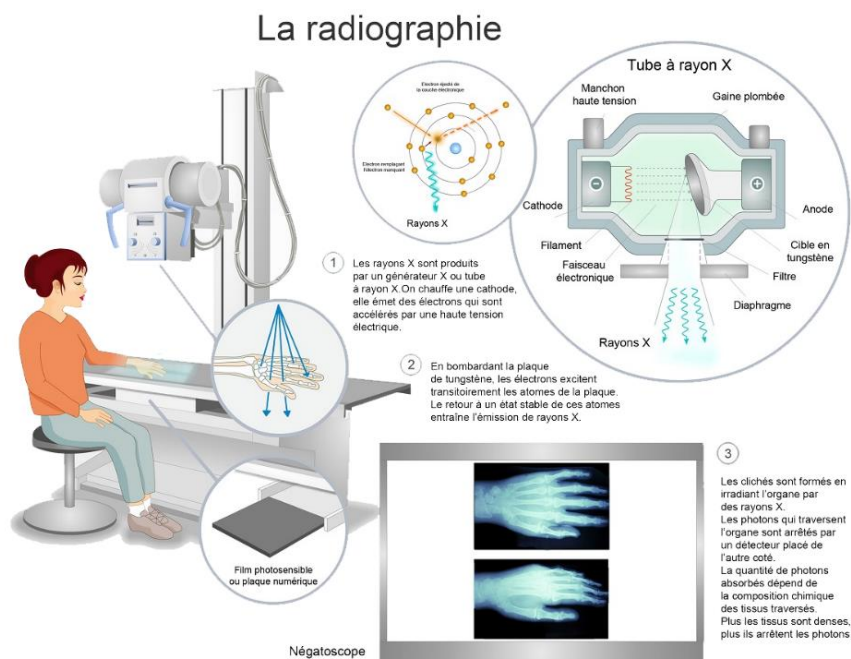


Fig.7 L'imagerie médicale par modalité et par axe thérapeutique

### 3.3 Radiographie

L'appareil utilisé, se compose d'un bras articulé muni d'un tube à rayons x et d'une plaque contenant un film radiographique (voir Fig.8).

Le principe de la radiologie est une impression photographique par les rayons x sur un film argentique (plus précisément une cassette contenant un film sensible à la lumière visible et recouverte généralement d'un ou deux écrans renforçateurs qui s'illuminent lorsqu'ils sont exposés aux rayons x). Les rayons x peuvent traverser la matière, ce qui explique que la quantité de rayons x reçue dans les différentes zones du film dépend de l'absorption des rayons x par les différents tissus traversés soit leur densité. Les tissus de faible densité comme la peau les muscles et les organes contenant de l'air, apparaissent plus foncés parce qu'ils absorbent moins les rayons x, il se laisse facilement traverser quant aux tissus beaucoup plus denses, comme les os, ils absorbent davantage les rayons x et apparaissent blancs sur la radio. la radiographie est en somme, la photographie au moyen des rayons x. [9]



*Fig.8 Principe de la radiologie*

La formation de l'image radiologique à rayon x (RX) se fait schématiquement en quatre étapes au sein d'une chaîne radiologique (voir fig.9.A) qui comporte la production des RX, la formation de l'image radiante, la modulation du faisceau de RX par la traversée de l'objet, la détection de l'image radiante recueillie sur un détecteur analogique ou numérique et enfin la présentation de l'image définitive [10].

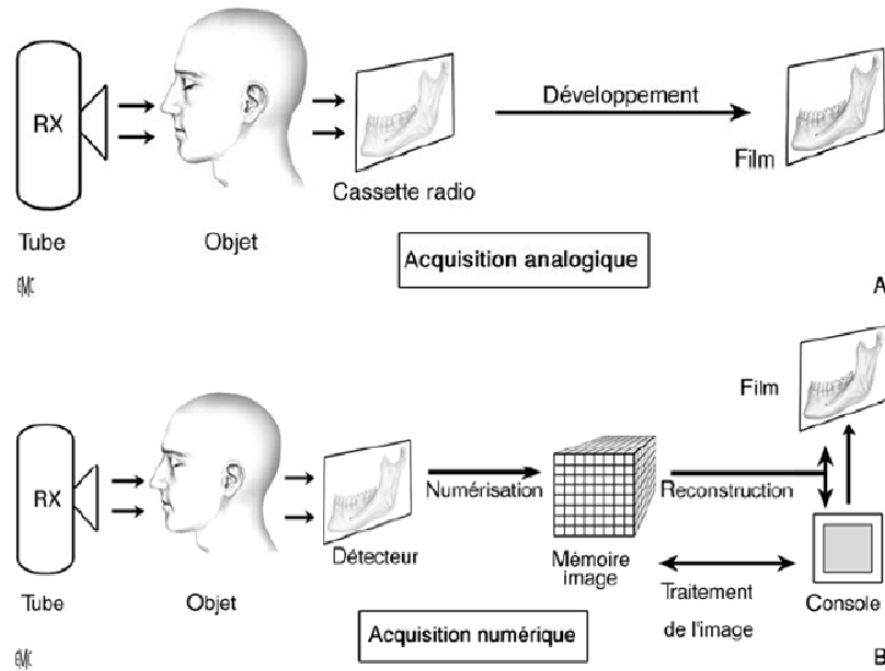


Fig.9 Formation de l'image en rayon X en mode analogique (A) et numérique (B)

Dans les équipements les plus récents la numérisation de l'image est nécessaire pour la récupération d'image de bonne qualité et pour la sauvegarde et la restitution sans pertes de ces images (voir Fig.9.B)

### 3.4 Le scanner

Le scanner x appelé aussi tomodensitométrie est une technique d'imagerie médicale qui permet d'étudier différentes parties du corps humain, comme notamment le cerveau, la cage thoracique, l'abdomen ou bien encore les os. Le scanner permet de rechercher des anomalies qui ne sont pas visibles sur des radiographies de bases ou encore sur l'échographie [9]

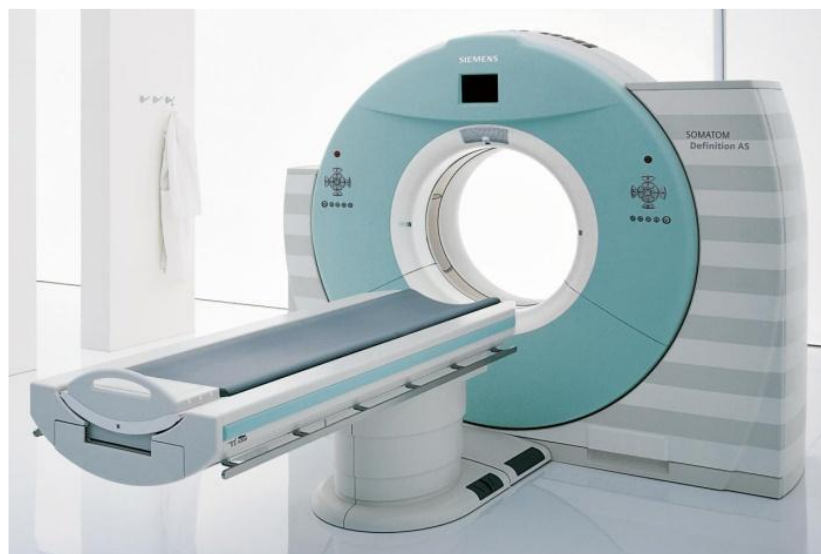


Fig.10 Appareil de radiologie de type scanner

### 3.4.1 Le principe général du scanner à rayon x

Une source de rayon x externe irradie le patient. Dans un scanner, les rayons x sont captés et mesurés par des détecteurs situés de l'autre côté du patient (voir Fig.8). La source et les détecteurs tournent autour du patient. La table se déplace très lentement à une vitesse constante.

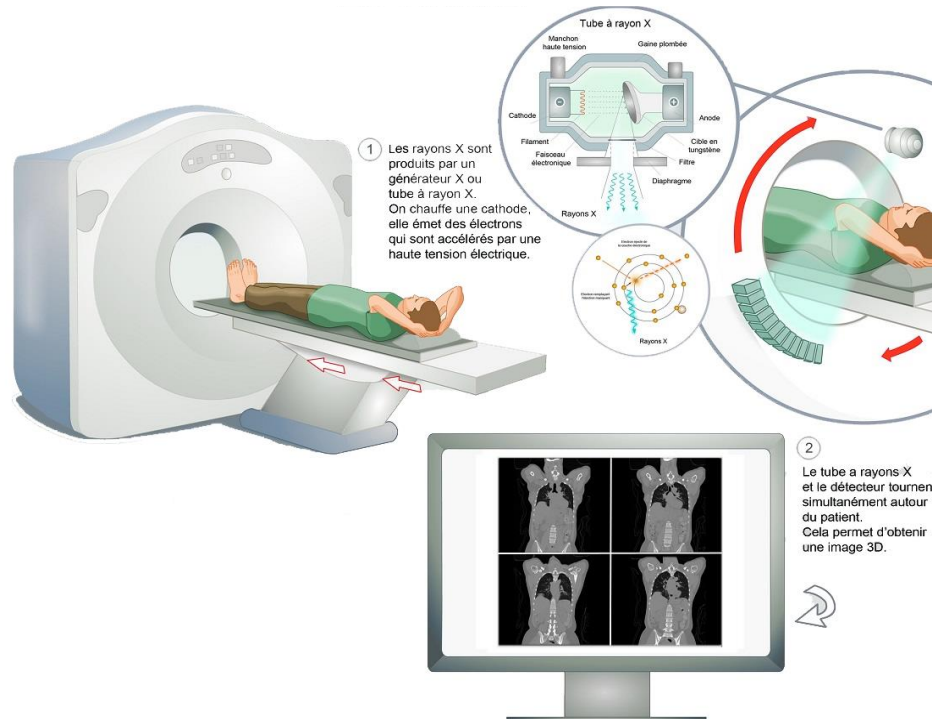


Fig.8 Schéma de principe général de scanner.

## 3.5 L'échographie

L'échographie est une technique d'exploration de l'intérieur du corps basée sur les ultrasons. Elle étudie les organes pleins de l'abdomen, le cœur et tous les organes non masqués par lui et elle a révolutionné la surveillance de la grossesse. L'élément de base de l'échographie est une céramique piézoélectrique [11]

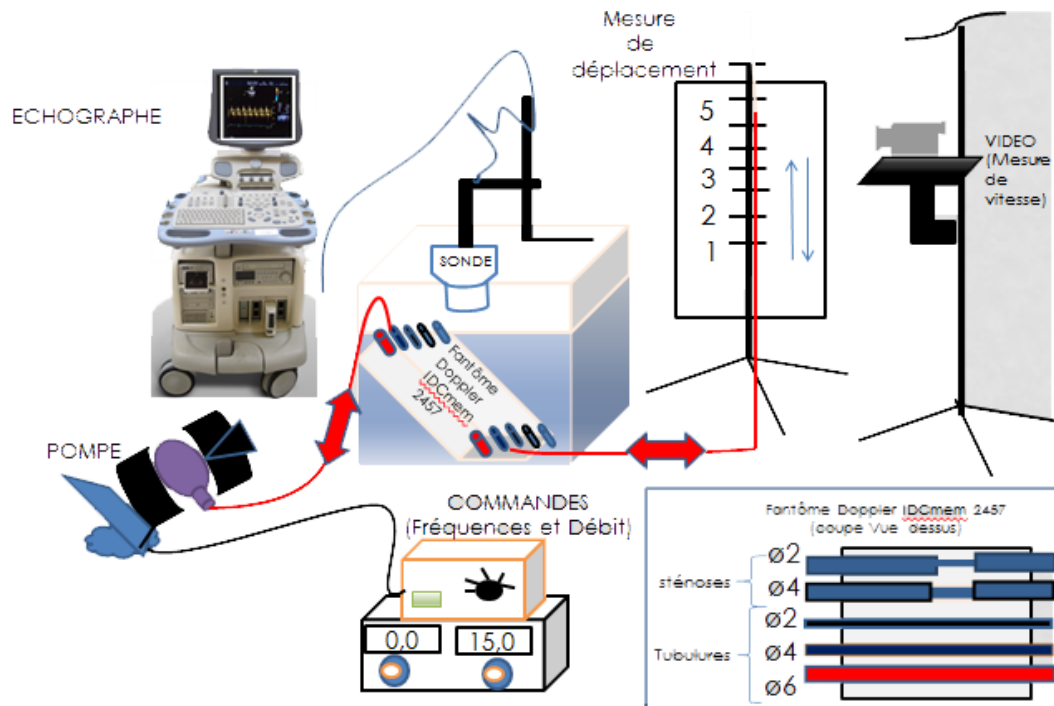


Fig.9 Eléments fonctionnels d'un appareil d'échographie médical

### 3.6 L'Imagerie par résonance magnétique (IRM)

L'IRM est une technique d'imagerie médicale non invasive largement répandue dans les milieux hospitaliers. Elle donne accès à des images en deux ou trois dimensions de l'intérieur du corps avec de bonnes résolutions spatiale et temporelle. Ces images concernent les tissus mous, essentiellement le système nerveux, les poumons, les muscles ou le cœur. L'IRM permet de donner des informations sur l'anatomie mais aussi sur la fonction de ces organes. Elle repose sur le principe de résonance magnétique nucléaire. [12]

À l'aide d'un champ magnétique, d'ondes radio et d'ordinateurs qui génèrent des images des tissus internes, l'imagerie par résonance magnétique comporte un avantage par rapport à d'autres techniques d'imagerie (voir Fig.10). L'IRM permet aux médecins d'établir un diagnostic et de proposer des traitements efficaces sans faire appel à la radiation ou à du matériel radioactif. [12]

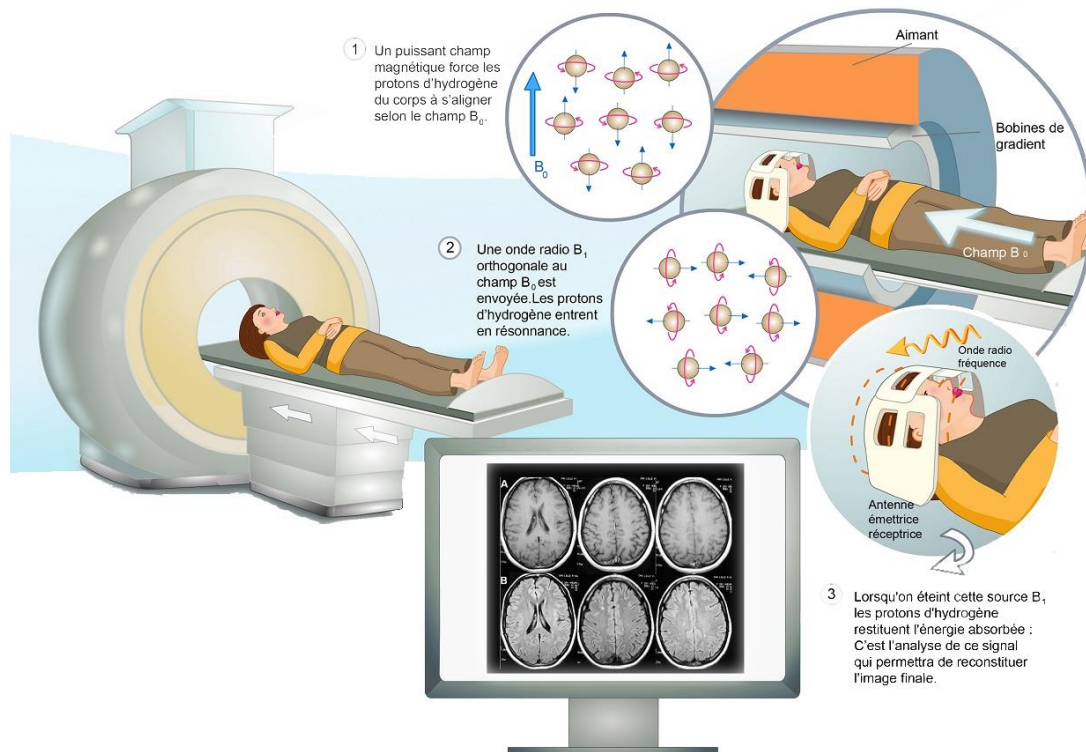


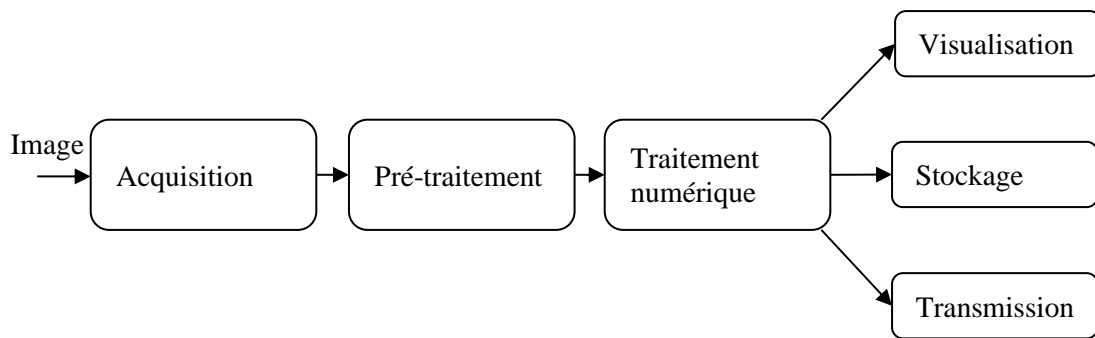
Fig.10 Eléments de l'imagerie par résonance magnétique

L'intensité recueillie pour un élément de volume (voxel) dépend de la concentration de l'eau à l'endroit considéré et on obtient ainsi une image tridimensionnelle de la répartition de l'eau dans le corps d'un patient. Selon la méthode utilisée, le contraste entre deux voxels peut être augmenté si les temps de relaxation des spins nucléaires (décrivant le retour à l'équilibre des noyaux après l'excitation) diffèrent dans les deux zones. Il est donc possible d'observer des altérations des tissus (telles que des tumeurs) grâce aux différences de densité et de relaxation de l'eau [13].

Après l'obtention des images du patient ainsi que toutes les informations associées, il nécessite un format identique selon une norme permettant l'interconnexion et l'interaction « interopérabilité » des équipements et le transfert des données. Une des normes la plus utilisée par la plupart des fabricants de matériel d'imagerie médicale c'est la norme DICOM. [14]

#### 4 Architecture générale d'un système de traitement d'images

Le traitement d'image désigne une discipline de l'informatique et de mathématique appliquées, qui étudie les images numériques et leurs transformations dans le but d'améliorer leur qualité ou d'extraire des informations pertinentes. Un système de traitement d'image est composé de plusieurs parties, la figure 11 montre les différentes étapes de traitement. [15]



**Figure I.6:** Schéma synoptique d'un système de traitement d'image

## 4.1 Acquisition

L'acquisition d'une image est l'opération qui permet le passage de l'information réelle à une représentation numérique. Elle est réalisée en 3 étapes :

- La transformation du signal optique en un signal analogique (électrique),
- L'échantillonnage qui consiste à multiplier le signal analogique par une série d'impulsion unité dans le but d'avoir des échantillons du signal,
- La quantification qui est une traduction des échantillons en valeurs numériques selon une règle de codage choisi.

## 4.2 Pré-traitement et post-traitement

Le prétraitement concerne les images acquises tandis que le post-traitement concerne les images traitées. Dans les deux étapes, les images ne peuvent échapper aux effets de dégradation due essentiellement aux phénomènes physiques.

L'opération de prétraitement a pour but principal la correction de l'imperfection du système d'acquisition, alors que l'opération post-traitement vise surtout à maitre en forme pour éventuellement une visualisation, transmission ou un stockage.

Les techniques de prétraitement les plus utilisées reposent sur le filtrage numérique et la modification de l'histogramme.

## 4.3 Le traitement numérique

Le traitement numérique d'image est l'ensemble des méthodes qui permettent de décrire quantitativement le contenu d'une image. On distingue deux types de traitement, les traitements de bas niveau et les traitements de haut niveau. Cette distinction est liée au contenu sémantique des entités traitées et extraites de l'image.

Les traitements de bas niveau opèrent en général sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire la liaison avec la réalité. Ils opèrent plutôt sur des données de nature numérique. Les traitements de haut niveau s'appliquent à des entiers de nature symbolique associés à une représentation de la réalité extraite de l'image ils sont relatifs à l'interprétation.

#### **4.4 Visualisation, transmission et stockage**

La visualisation est une opération qui permet de transformer le signal numérique qui est la matrice image en un signal analogique visible par l'œil de l'observateur à travers un dispositif d'affichage.

L'image peut être transmise vers une station d'archivage ou de traitement qui peut être différente de la station d'acquisition. Les images sont archivées sur des supports destinés à cet effet comme par exemple les bandes magnétiques, les disques durs...etc.

### **5 Conclusion**

Dans le chapitre, nous avons présenté des généralités sur le traitement d'image numérique, et sur l'imagerie médicales, nous avons ainsi fait une rétrospective rapide sur les techniques de radiologie tel que le scanner, l'IRM...etc.

## Chapitre 2

# Généralités sur les cartes FPGA PYNQ

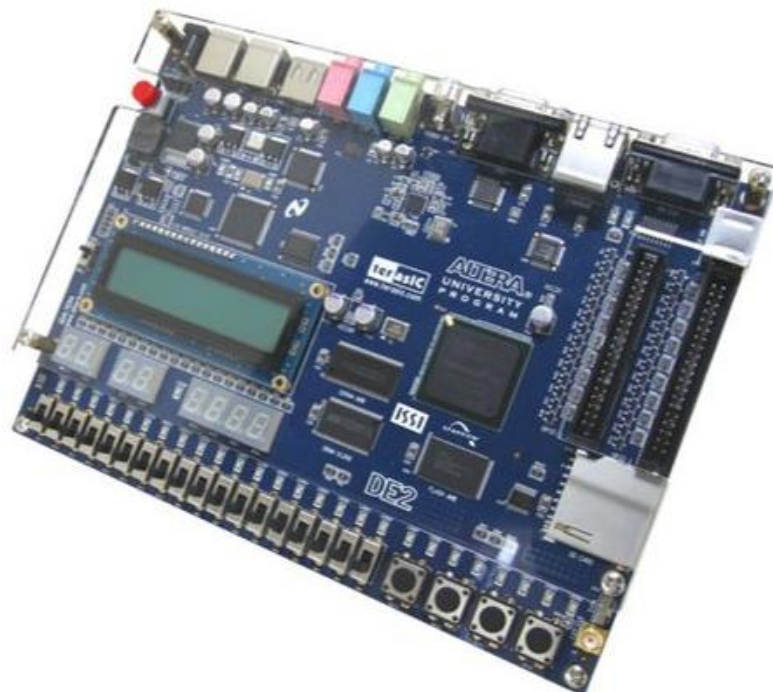
## 1 Introduction

Le traitement des images numériques est un domaine caractérisé par le besoin de nombreux travaux expérimentaux visant à établir la viabilité des solutions proposées. On désigne par le traitement d'images numériques l'ensemble des techniques permettant de transformer une image numérique en une nouvelle image ou même combiner plusieurs images entre elles.

Dans ce chapitre nous allons présenter les outils matériels dont nous disposons pour effectuer le traitement d'images médicales escompté. Nous utiliserons à cet effet une carte FPGA de type Xilinx Zynq 7000, pour cette raison nous allons passer en revue des généralités sur les circuits FPGA classique, ainsi que sur les circuits FPGA intégrant un processeur ARM.

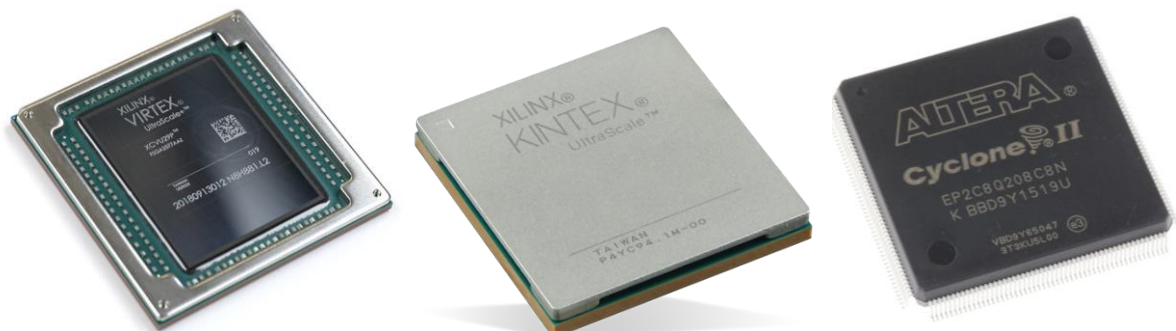
## 2 FPGA

Un FPGA (pour Field-Programmable Gate Array) est un circuit intégré composé d'un nombre d'élément logiques programmables reliés entre eux grâce à une matrice de routage qui elle-même programmable. Cette structure permet au FPGA d'émuler n'importe quel circuit, à la seule condition que celui-ci ne soit pas trop gros pour ne pas épuiser les ressources logiques et le routage du circuit FPGA. [17]



*Fig.1 Exemple de carte FPGA (Altera DE2 Board)*

Cependant, cette souplesse de programmable se paye au niveau des performances circuit. Bien que les FPGA soient gravés avec la même technologie que les processeurs généralistes cadencés à quelques gigahertz. [18]



*Fig.2 Exemples de circuit FPGA*

La fréquence d'horloge des FPGA ne dépasse pas des centaines de mégahertz pour les modèles les plus récents. Néanmoins, la grande profiter au mieux du parallélisme à grain fin des circuits implémentés, ce qui compense largement les relativement faibles fréquences atteignables.

## 2.1 Domaines d'application

Les cellules du circuit FPGA étant indépendantes les unes des autres, elles peuvent très bien effectuer leurs calculs en parallèle. Le FPGA permet ainsi d'optimiser très finement des applications en jouant sur le parallélisme des traitements. On retrouve par conséquent l'utilisation de ce type de cartes dans de nombreux domaines d'applications très calculatoires comme :

- ✓ Le traitement image
- ✓ Le traitement du signal
- ✓ La commande numérique
- ✓ Les télécommunication
- ✓ La cryptographie
- ✓ La bio-informatique

Ou encore plus généralement l'accélération de calculs scientifiques. [19]

## 2.2 Les circuits FPGA

### 2.2.1 Technologie FPGA Xilinx

Les circuits FPGA, inventés en 1985 (par la société Xilinx), sont les plus populaires actuellement. Cela est en partie dû à leur grande capacité d'intégration inégalée dans le domaine des circuits programmable sur site. [20]

Chaque ilot est composé généralement de quatre, huit ou seize cellules logiques, ainsi que d'un routeur (switch box) programmable réalisant la connexion entre les entrées et sorties des cellules logiques et la matrice de routage. Cette matrice de routage couvre elle aussi toute la surface du FPGA, sous forme de pistes horizontales et verticales. Chaque cellule d'un FPGA utilisant la technologie SRAM peut être vu comme une cellule standard de taille moyenne.

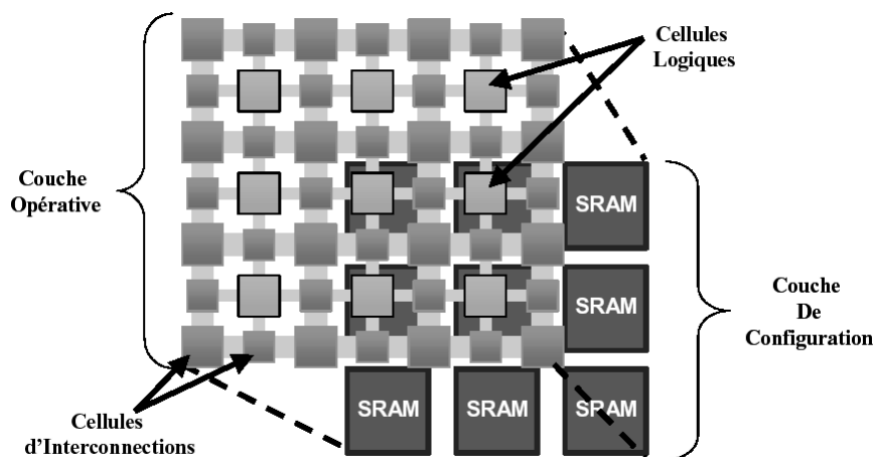


Fig.3 Architecture de principe d'un FPGA SRAM.

Certaines de ces pistes relient des ilots voisins, alors que d'autres traversent tout le FPGA pour permettre de relier deux ilots distants l'un de l'autre. Les switch boxes situés au niveau de chaque ilot réalisent les connexions nécessaires entre ces différents fils.

De plus, des blocs de mémoire dédiés (RAM blocks) sont répartis sur la surface du FPGA. Connectés à la matrice de routage, ces blocs offrent un espace mémoire confortable utilisable par la logique programmable voisine.

Enfin, pour permettre au FPGA de communiquer avec le monde extérieur, des cellules d'entrée/sortie (IO blocks) sont disposées sur tout le pourtour du circuit et elles aussi reliées à la matrice de routage. La Figure 4 présente sur la gauche une vue simplifiée de la surface d'un FPGA type, sur lequel on peut distinguer les cellules d'entrée/sortie, les blocs de mémoire, ainsi que les ilots et la matrice de routage. Un de ces ilots est représenté plus en détails sur la droite.

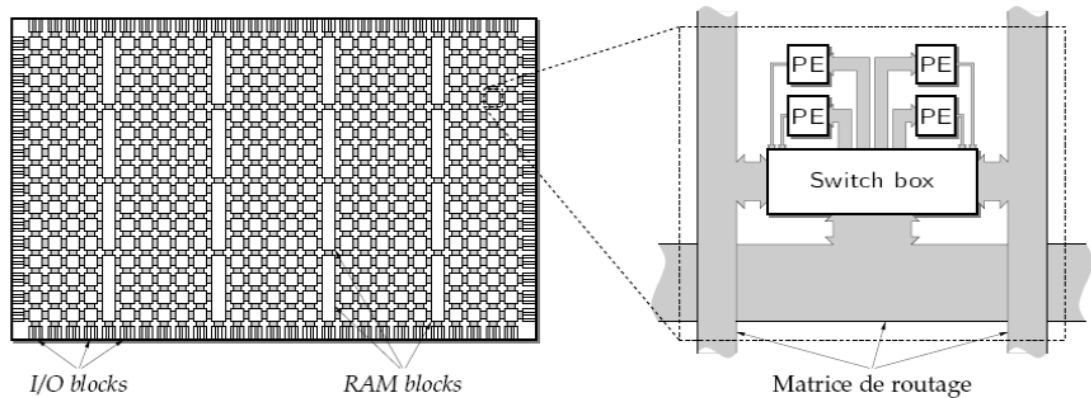


Fig.4 Architecture de Principe d'un FPGA SRAM

### 2.2.2 Les cellules logiques d'un FPGA

Consistent en général simplement en une partie de traitement et d'une partie de mémorisation. La partie traitement est assurée par une table (Look-Up) avec n bits d'adresse et 1 bit de sortie. Quant à la partie de mémorisation, c'est un registre contrôlé par un signal horloge.

[21]

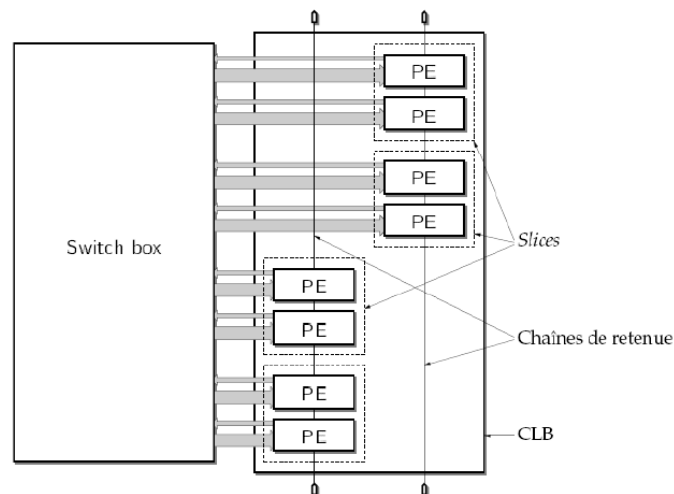


Fig.5 Sstructure d'un CLB.

### 2.3 Flot de conception classique des circuits FPGA

Le flot de conception est les étapes qu'on doit suivre pour produire l'implémentation de l'application ou le bitstream qui est employé pour configurer le FPGA. Une plate-forme efficace de l'FPGA, ne mènera pas à l'implémentation optimale de l'application sans un ensemble

efficace d'outils pour projeter l'application sur les ressources du dispositif, vérifie sa fonctionnalité et sa contrainte du temps, et en fin, produire le bitstream pour la configuration.

Une conception typique pour FPGA implique quatre étapes, qui sont montrées sur la figure III.1, et ils sont décrits brièvement ci-dessous.

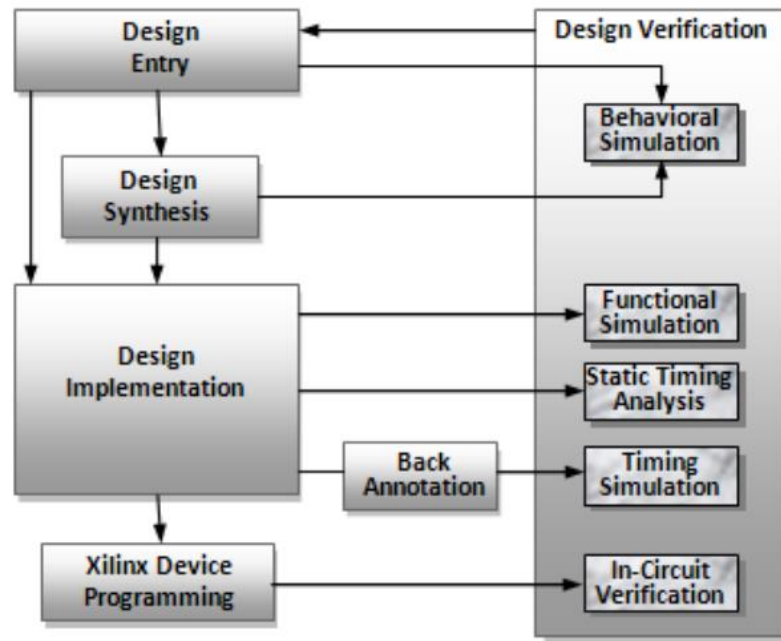


Fig.6 Flot de conception typique des circuits FPGA

### 2.3.1 L'entrée de la conception

L'entrée de la conception est la première étape, elle permet de décrire le fonctionnement ou la structure du circuit visé. Une série d'outils sont disponibles pour accomplir cette étape. Quelques concepteurs utilisent l'entrée schématique tandis que d'autres préfèrent de spécifier leur conception en utilisant un langage de description matériel tel que le langage VHDL ou Verilog. D'autres préfèrent de mélanger les deux méthodes dans la même conception. Les entrées au flot de conception incluent typiquement les spécifications HDL de la conception ou le schéma, les contraintes de conception, et le dispositif FPGA cible.

- La première entrée est la description RTL de la conception. Si des spécifications plus élevées (comportementale) sont employées alors un outil de synthèse comportementale doit être utilisé pour convertir la description comportementale en spécifications RTL.
- Les contraintes de conception incluent typiquement la fréquence d'opération désirées, les retards d'acheminement entre les broches d'entrée et celles de sorties,

entre les broches d'entrée et les bascules, et entre les bascules et les broches de sortie. Cependant, il faut noter que les contraintes de conception peuvent inclure des contraintes de localisation physique, qui spécifient que certains éléments logiques soient placés à certains endroits.

- La troisième entrée de conception est le choix du dispositif FPGA cible.

### 2.3.2 La synthèse

Il y a deux types de synthèse matérielle, à savoir, la synthèse RTL, qui crée une netlist générique de niveau porte de la description RTL, la synthèse comportementale qui crée une description RTL à partir d'une représentation algorithmique. Dans la synthèse RTL, la description de circuit est convertie en netlist de portes logiques génériques. Ceci implique l'optimisation logique qui est une opération indépendante de technologie qui simplifie la fonction logique de la conception sans utilisation des informations spécifique à la technologie.

### 2.3.3 L'implémentation de la conception

Après que la conception soit écrite et synthétisée, elle est prête pour l'implémentation sur le dispositif FPGA cible. Le mappage technologique permet le mappage des portes génériques de la conception dans les composants de la bibliothèque logique du dispositif cible. Alors le logiciel groupe les primitifs dans des blocs logiques. Après, le logiciel d'implémentation cherche le meilleur placement pour placer les blocs logiques parmi toutes les possibilités. Le but primaire des outils de placement est de réduire la quantité de ressources de routage exigées et de maximiser la performance de système.

Quand le processus de placement et routage est fini, le logiciel crée le fichier bitstream employé pour configurer l'FPGA cible. Dans de grandes ou complexes applications, le logiciel peut ne pas pouvoir placer et router la conception avec succès. Les outils permettent au concepteur d'essayer différentes options ou d'exécuter plus d'itération afin d'essayer d'obtenir une conception plein-router. En outre, quelques fournisseurs fournissent des outils floor-planning pour aider la disposition physique des composant du circuit.

## 3 Les SOC FPGA

Les dispositifs SOC FPGA intègrent à la fois les architectures processeur et FPGA dans un seul dispositif. Par conséquent, ils offrent une meilleure intégration, une consommation réduite, une taille de carte plus petite et une communication à bande passante plus élevée entre le processeur et le FPGA. Ils comprennent également un riche ensemble de périphériques, une

mémoire sur puce, une matrice logique de type FPGA et des émetteurs-récepteurs à grande vitesse. [26]

### 3.1 Les FPGA Xilinx Zynq-7000

Nous avons choisi le dispositif Xilinx Zynq-7000 pour notre preuve de concept pratique car il s'agit d'un choix populaire pour les conceptions de systèmes embarqués contemporains.

Le Xilinx Zynq-7000 est un SoC FPGA composé d'un

- Processeur ARM Cortex A9 double cœur et d'un tissu FPGA Xilinx série 7 sur le même die.
- Le système de traitement comprend une MMU et un cache à deux niveaux.
- Une petite RAM sur puce de 256 Ko est disponible pour le stockage d'informations sensibles ou de code.
- Une plus grande mémoire DDR externe est accessible via les contrôleurs de mémoire. Le bus mémoire principal est un système de bus ARM AMBA AXI.
- La communication externe est prise en charge via les interfaces CAN, I2C, Ethernet et USB. L'appareil comprend une implémentation AES et HMAC de base qui sont utilisées lors du décryptage et de la vérification de l'authenticité des images et des fichiers de configuration à exécuter sur le Zynq-7000.
- L'environnement d'exécution sécurisé ARM connu sous le nom de TrustZone est disponible pour la sécurité d'exécution du système logiciel.
- Un paramètre matériel à un bit divise tous les processus et périphériques en un monde sécurisé ou normal. [27]

Processus de démarrage sécurisé sur le Xilinx Zynq-7000

La figure II-6 illustre le processus de démarrage sur le SoC FPGA Zynq-7000. Il se compose de cinq étapes après la mise sous tension :

1. BootROM (code câblé interne non accessible)
2. Chargeur de démarrage de première étape (FSBL)
3. Configuration FPGA (bitstream)
4. Chargeur de démarrage de deuxième étape (c'est-à-dire U-boot)
5. Système d'exploitation (SE)

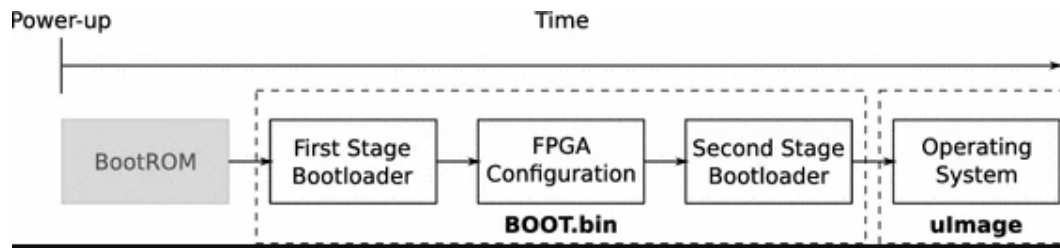


Fig.7 Processus de démarrage Xilinx Zynq-7000

### 3.1.1 Le PYNQ

PYNQ est un Framework logiciel open source basé sur Linux et Ubuntu ainsi que sur le langage Python, fournissant principalement une plate-forme de développement basée sur Python pour le SoC Zynq-7000 de Xilinx, les appareils Zynq UltraScale+ MPSoC et la série Alveo de cartes d'accélération de données de Xilinx. PYNQ signifie "Python Productivity for Zynq", comme son nom l'indique, PYNQ simplifie le processus de création d'applications embarquées utilisant des appareils Zynq à l'aide du langage de programmation Python le plus populaire, c'est un outil formidable pour les développeurs embarqués.

Avec la plate-forme PYNQ, les concepteurs peuvent utiliser Python, un langage de programmation productif, pour développer des applications de processeur et améliorer l'efficacité du développement ; Dans le même temps, sur la base de la superposition matérielle et de l'API dans la bibliothèque d'accélération matérielle PYNQ et OpenCV fournie par Xilinx, l'accélération matérielle peut être obtenue en utilisant pleinement les ressources (logique programmable) dans Zynq-7000 et Zynq UltraScale+ MPSoC. [28]

PYNQ peut être considéré comme une pile multicouche, s'étendant de la couche matérielle inférieure à la couche intermédiaire du système d'exploitation (y compris les pilotes logiciels et l'API), puis à la couche supérieure des applications logicielles. Il fournit un ensemble de composants logiciels et matériels intégrés afin que les développeurs non seulement utilisent directement les composants existants, mais aussi ajustent et étendent les fonctionnalités selon les besoins. En résumé, PYNQ accélère la co-conception matérielle/logicielle des systèmes embarqués basés sur Zynq et améliore la productivité du développement.

### 3.1.2 L'Architecture PYNQ

En tant que Framework open source, PYNQ est conçu pour découpler les détails d'implémentation du FPGA matériel sous-jacent de l'utilisation de la couche d'application supérieure par le biais d'un packaging de haut niveau. Pour les développeurs de logiciels, le Framework PYNQ a fourni une bibliothèque complète pour accéder aux ressources FPGA, il

permet aux développeurs d'applications de niveau supérieur d'appeler des modules FPGA via la programmation Python, et ils n'ont pas besoin de comprendre la programmation matérielle Verilog/VHDL pour profiter des nombreux avantages du calcul parallèle FPGA, d'une extension facile de l'interface et d'une configuration flexible.

L'architecture PYNQ est illustrée à la Figure II-6. Sur la figure, vous pouvez voir que l'architecture du système PYNQ est divisée en trois couches, à savoir la couche matérielle basée sur la conception FPGA, la couche logicielle basée sur le noyau Linux plus Python et Jupyter Notebook est la principale couche d'application

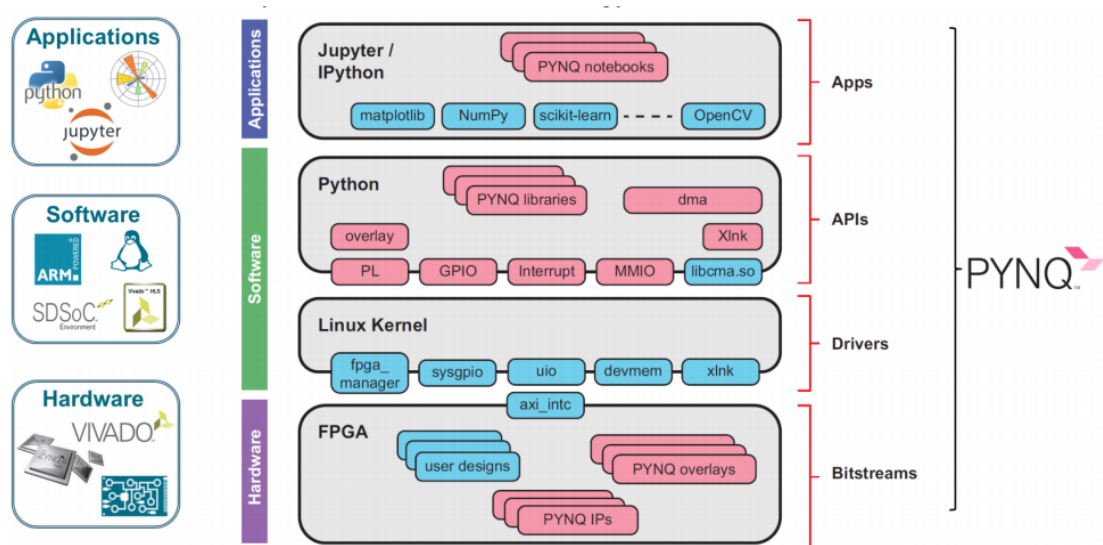


Fig.8 L'architecture PYNQ

La conception de la couche matérielle est la même que la méthode de conception intégrée pour réaliser l'interaction collaborative entre PS et PL. La conception de l'ensemble de la partie FPGA est appelée superposition, qui peut générer différents fichiers de flux binaire pour plusieurs utilisateurs et plusieurs applications, et peut être appelée via des API logicielles pour basculer dynamiquement les fonctions logiques sur le FPGA. [29]

## 4 Conclusion

Dans ce chapitre, nous avons décrit les plateformes de programmation que nous allons utiliser dans le traitement d'image médicale. Nous avons choisi pour cette fin une carte FPGA de type PYNQ. Alors nous avons décrit dans les sections précédentes les grandes lignes des circuits FPGA classique ainsi que leur flot de conception et ensuite nous avons présenté des généralités sur les cartes PYNQ avec des circuits FPGA/SOC de Xilinx.

# Chapitre 3

## Application au traitement d'image médicale avec Python

## 1 Introduction

Au cours du 20<sup>ème</sup> siècle, la médecine a vu l'apparition d'un nouvel outil d'aide au diagnostic qui l'a révolutionnée, c'est l'imagerie médicale telle que l'IRM (Imagerie par Résonance Magnétique) ou la TDM (Tomodensitométrie).

L'imagerie médicale offre une série de plans 2D du patient dont l'intensité des pixels représente une propriété physique différente selon les caractéristiques des tissus concernés. Cependant, si les techniques d'acquisition évoluent chaque jour, la lecture et la compréhension des images restent souvent un art difficile à maîtriser. Les progrès réalisés en technologie informatique ont permis de résoudre en partie les difficultés de lecture en traduisant l'information contenue.

Dans ce chapitre, nous allons nous intéresser au traitement des images médicales par le langage de programmation python, nous allons ainsi présenter le langage précité ainsi que les outils qu'y s'apparente. Ensuite, nous allons appliquer les différentes techniques du langage pour la détection des poumons en utilisant des image TDM.

## 2 Présentation du langage de programmation Python

Python est un langage de programmation dit de "très haut niveau". Cela signifie qu'il possède un haut niveau d'abstraction par rapport au langage machine. Pour le dire très simplement : plus un langage de programmation est de "haut niveau", plus sa syntaxe se rapproche de notre langage (l'anglais) plutôt que du langage machine. Un langage de haut niveau est donc plus facile à comprendre et à utiliser qu'un langage de plus bas niveau.[30]. En effet, un langage comme Python fonctionnera avec n'importe quel système d'opération (OS) du moment que l'OS en question possède l'interpréteur Python.

### 2.1 Techniques de production des programmes

Il existe deux techniques de production des programmes

La compilation est la traduction du code source en langage objet. Elle comprend au moins quatre phases (trois phases d'analyse — lexicale, syntaxique et sémantique — et une phase de production de code objet). Pour générer le langage machine il faut encore une phase particulière : l'édition de liens. La compilation est contraignante, mais offre finalement une grande vitesse d'exécution. [33]



Fig. 1 Chaîne de compilation

Dans la technique de l'interprétation, chaque ligne du code source analysé est traduite au fur et à mesure en instructions directement exécutées. Aucun programme objet n'est généré. Cette technique est très souple, mais les codes générés sont peu performants : l'interpréteur doit être utilisé à chaque nouvelle exécution...[33]

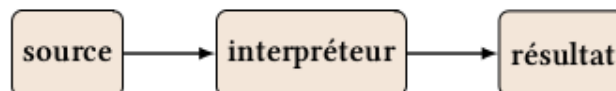


Fig.2. Chaîne d'interprétation

- Technique mixte : **l'interprétation du bytecode compilé**. Bon compromis entre la facilité de développement et la rapidité d'exécution.
- Le *bytecode* (forme intermédiaire) est portable sur tout ordinateur muni de la **machine virtuelle Python**.

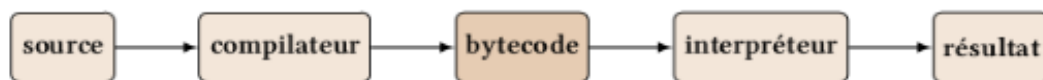


Fig.3 Interprétation du bytecode compilé

Pour **exécuter un programme**, Python charge le fichier source *.py* en mémoire vive, en fait l'analyse (lexicale, syntaxique et sémantique), produit le bytecode et enfin l'exécute.

Afin de ne pas refaire inutilement toute la phase d'analyse et de production, Python sauvegarde le bytecode produit (dans un fichier *.pyo* ou *.pyc*) et recharge simplement le fichier bytecode s'il est plus récent que le fichier source dont il est issu.

En pratique, il n'est pas nécessaire de compiler explicitement un module, Python gère ce mécanisme de façon transparente.

## 3 Programmer avec python

### 3.1 Installation de Python

Nous allons tout d'abord installer Python 3.10.4 Pour cela, il faut télécharger le programme d'installation disponible sur le site internet de Python à l'adresse suivante: <http://www.python.org/downloads/> , puis cliquer sur le "Download Python 3.x.y".

1. Enregistrez le programme d'installation sur votre disque dur.
2. Lancez le programme d'installation qui vient d'être téléchargé.
3. Choisissez "Exécuter" lors de l'avertissement de sécurité.
4. Cliquez sur "Next >"
5. Cliquez sur "Next >".



Fig.4 Site web <http://www.python.org/downloads/>

### 3.2 Lancement de l'interpréteur Python 3.10.4

Afin de vérifier que Python 3 est bien installé et fonctionne correctement, nous allons lancer l'interpréteur (ou *shell*) de Python. Pour cela, cherchez "Python" dans le menu Démarrer et lancer "IDLE (Python GUI)".

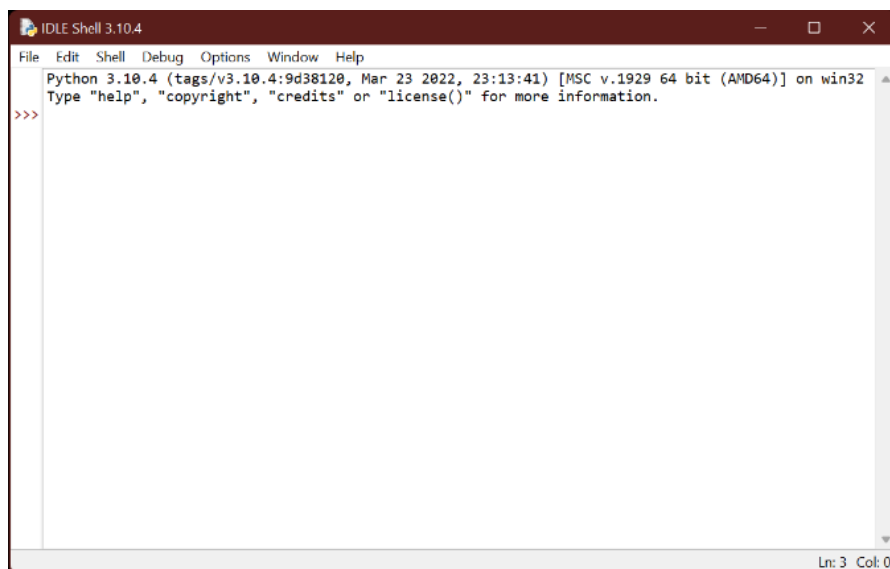


Fig.5 Interface de IDLE (Python GUI).

Cet écran n'est pas pratique, il est conseillé d'utiliser un programme supplémentaire pour pouvoir écrire et compiler facilement des programmes Python, il faut utiliser des logiciels dédiés tel Pycharm.

### 3.3 Environnement de développement intégré (IDE) : PyCharm

PyCharm est un outil développé par la société JetBrains, spécialiste du développement d'IDEs pour différents langages de programmation. Pour ce qui est du PyCharm, il faut aller sur la page de téléchargement de PyCharm <http://www.jetbrains.com/pycharm>.

- La version communautaire : elle est gratuite et déjà très riche. Tous les outils nécessaires à l'exécution de ce tutoriel sont présents dans cette version. Elle est non limitée dans le temps, mais, fonctionnellement parlant, il s'agit d'un sous-ensemble des possibilités offertes par la version professionnelle.[35]
- La version professionnelle : cette version payante.

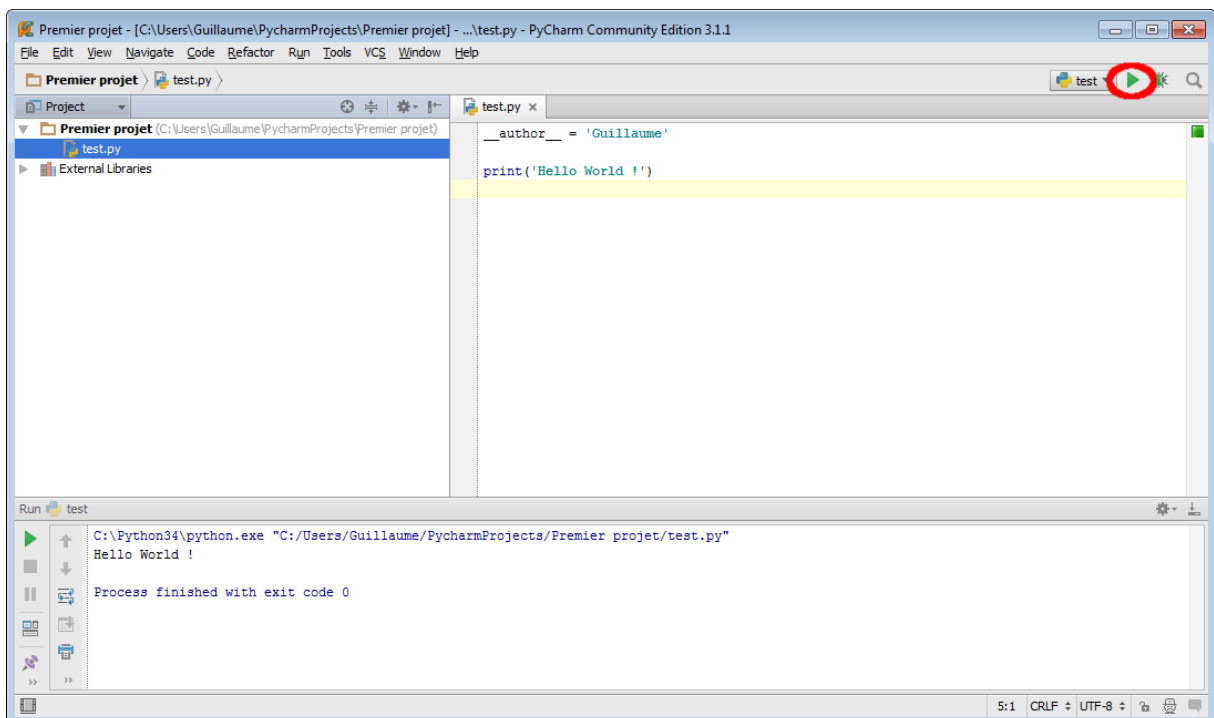


Fig.6 Interface de PyCharm.

PyCharm est utilisé pour la programmation en Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec les systèmes de contrôle de version (VCS). Il fonctionne sur plusieurs plates-formes sous Windows, Mac OS X et Linux.

### 3.4 L'outil de programmation Anaconda

Anaconda est une distribution des langages de programmation Python et R, tandis que Python est un langage de programmation généraliste de haut niveau. L'organisation Anaconda, Inc. (Continuum Analytics) a développé Anaconda. Anaconda fournit conda en tant que gestionnaire de paquets, tandis que le langage Python fournit pip en tant que gestionnaire de paquets. Python pip permet d'installer des dépendances python. En revanche, Anaconda conda permet d'installer des dépendances de bibliothèques python et non python. [37]

De plus, Anaconda est principalement utilisé pour la science des données et l'apprentissage automatique. Python est utilisé dans diverses applications telles que les systèmes embarqués, la vision par ordinateur, le développement Web, la programmation en réseau, y compris l'apprentissage par machine et la science des données. Généralement, Python a une grande communauté qu'Anaconda. [36]

#### 3.4.1 L'interface graphique de Anaconda Navigator

Anaconda Navigator est une interface incluse dans Anaconda. Grâce à cette interface, on peut facilement lancer des applications et gérer les différents packages et environnement. Il faut taper dans la barre de recherche “**Anaconda Navigator**” pour lancer cette interface.

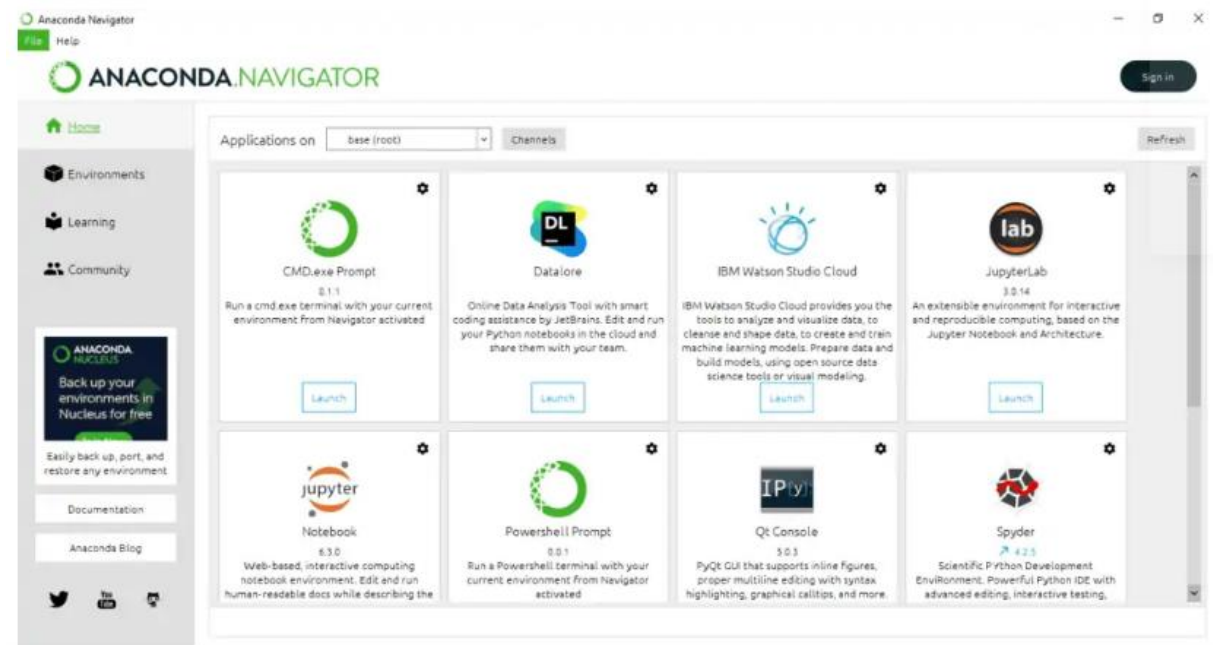


Fig.7 Interface graphique de Anaconda

Il existe plusieurs applications dans Anaconda Navigator tel que :

- JupyterLab
- JupyterNotebook
- Spyder
- Pycharm
- VSCode
- Orane 3 APP
- RStudioAnaconda powerShell

### 3.4.2 Jupyter

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala2. C'est un projet communautaire dont l'objectif est de développer des logiciels libres, des formats ouverts et des services pour l'informatique interactive. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code. Ces notebooks sont utilisés en science des données pour explorer et analyser des données de manière interactive. [42]

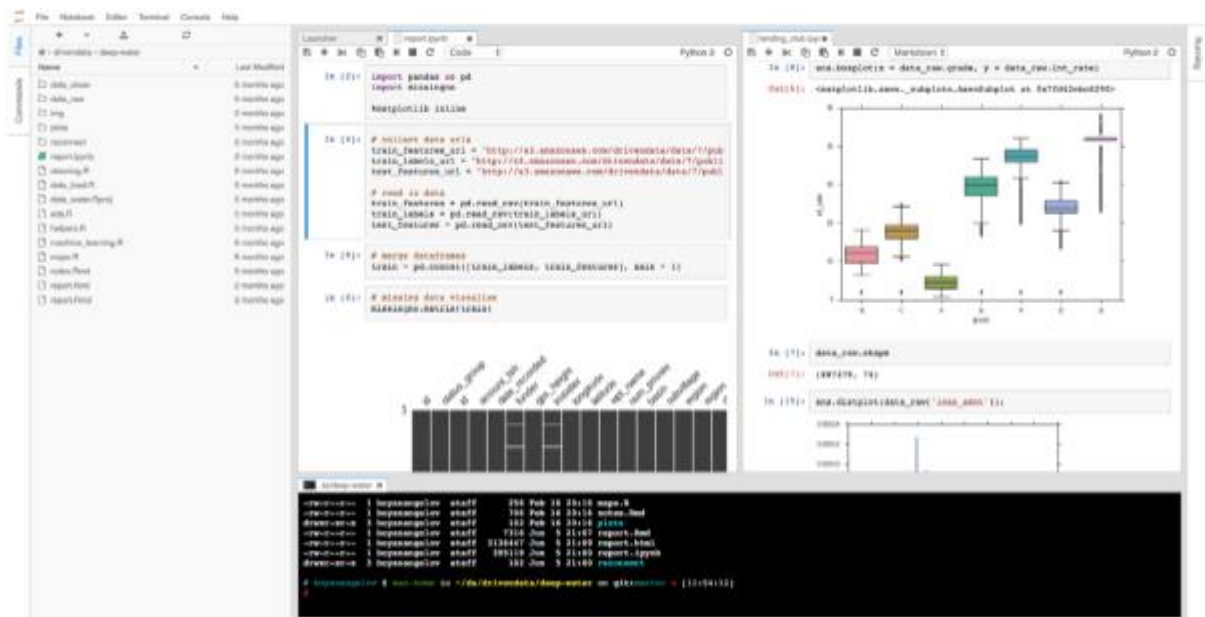


Fig.8 Interface Jupyter

### 3.4.3 Les avantages de Jupyter

Jupyter présente plusieurs avantages tels que :

- **La visualisation des données** : Les Données sont présentées sous la forme d'un graphique. Jupyter permet de créer ces visuels, de les partager et d'autoriser les modifications interactives sur le code partagé et les jeux de données ;
- **Le partage de code** : Certains services cloud tels que GitHub et Pastebin permettent le partage de code, mais ils ne sont pas interactifs. Ce qui est le cas de Jupyter ;
- **La documentation de code** : Il est possible de mettre des longues explications de code dans un cahier Jupyter ;
- **Les interactions en direct avec le code** : Le code est dynamique et peut être modifié à tout moment de façon incrémentale en direct avec le feedback.

Dans la figure précédente on voit que, dans la même interface, on a un notebook Jupiter associant du texte qui donne une explication du code ainsi que son graphique.

## 4 Application au traitement d'image TDM thoracique

Dans le reste de ce chapitre, nous allons présenter une application au traitement d'image en utilisant le langage de programmation Python. Dans cette application, nous allons essayer d'identifier sur des images de scanner TDM (tomodensitométrie) les poumons et ensuite nous allons mesurer la surface des poumons dans la coupe d'image utilisé. Cette technique est très utilisée pour le diagnostic de certaines maladies des poumons et du thorax.

L'identification des poumons dans les images TDM, nécessite l'utilisation des techniques de segmentation des images numérique.

### 4.1 Les type de segmentation d'image médicale

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation.

Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des

algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images.

1. La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes :
2. La segmentation fondée sur les régions (en anglais : region-based segmentation). On y trouve par exemple : la croissance de région (en anglais : region-growing), décomposition/fusion (en anglais : split and merge).
3. La segmentation fondée sur les contours (en anglais : edge-based segmentation).
4. La segmentation fondée sur la classification ou le seuillage des pixels en fonction de leur intensité (en anglais : classification ou thresholding).
5. La segmentation fondée sur la coopération entre les trois premières segmentations.

Dans le cas d'application que nous allons présenter, nous utilisons la technique numéros 4 qui est la segmentation basée sur le seuillage.

## 4.2 Introduction aux image TDM

La tomodensitométrie (TDM) dite aussi scanographie, tomographie axiale calculée par ordinateur (TACO), CT-scan (CT : computed tomography), CAT-scan (CAT : computer-assisted tomography), ou simplement scanner ou scanneur pour l'appareil, est une technique qui utilise des faisceaux de rayons X pour obtenir des intensités de pixels 3D du corps humain. Une cathode chauffée libère des faisceaux de haute énergie (électrons), qui à leur tour libèrent leur énergie sous forme de rayonnement X. Les rayons X traversent les tissus du corps humain et frappent un détecteur de l'autre côté. Un tissu dense (c'est-à-dire les os) absorbera plus de rayonnement que les tissus mous (c'est-à-dire la graisse). Lorsque les rayons X ne sont pas absorbés par le corps (c'est-à-dire dans la région de l'air à l'intérieur des poumons) et atteignent le détecteur, nous les voyons en noir, semblable à un film noir. À l'opposé, les tissus denses sont représentés en blanc.

De cette manière, l'imagerie CT est capable de distinguer les différences de densité et de créer une image 3D du corps. [43]

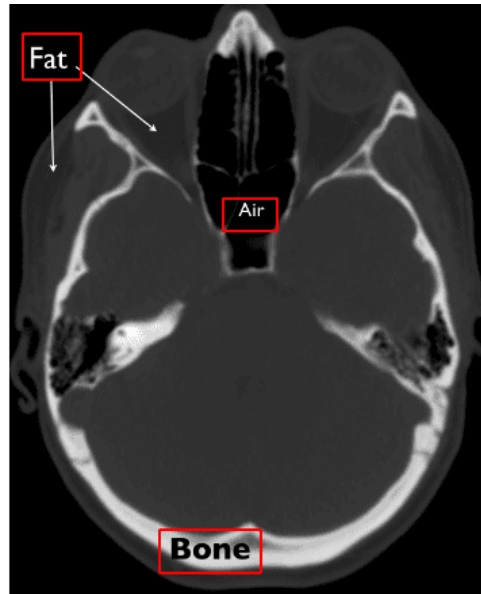


Fig. 8 Exemple d'image d'une TDM d'un crâne humain

#### 4.2.1 Intensités CT et unités Hounsfield

Les valeurs d'atténuation CT sont exprimées, selon une échelle de densité linéaire, en « unités Hounsfield (HU) », d'après Sir *Godfrey Newbold Hounsfield*, l'inventeur de la tomodynamométrie. Dans l'échelle de Hounsfield, l'eau se voit attribuer arbitrairement une valeur de 0 HU. Toutes les autres valeurs CT sont calculées selon [45]

$$HU = 100 * \frac{\mu_{tissue} - \mu_{H_2O}}{\mu_{H_2O}} \quad (1)$$

Dans laquelle  $\mu$  est le coefficient d'atténuation linéaire CT.

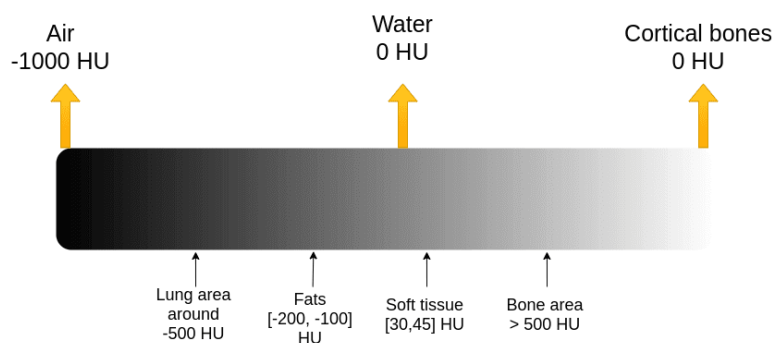
Les valeurs HU pour chaque pixel (qui reflètent la densité électronique du tissu imagé à un emplacement donné) sont converties en une image numérique en attribuant une intensité d'échelle de gris à chaque valeur. Plus le nombre est élevé, plus l'intensité du pixel est lumineuse. Par exemple, étant donné que la graisse est moins dense que l'eau, avec une valeur HU comprise entre -30 et -70, la graisse apparaît toujours plus sombre que l'eau sur les images CT. Certaines valeurs HU approximatives pour les tissus couramment trouvés sur les TDM de la tête sont présentées dans le tableau 1. [44]

**Tableau 1. Valeurs HU approximatives des tissus trouvés sur les TDM de la tête**

Unités Hounsfield	Tissu
> 1000	Os, calcium, métal
100 to 600	Contraste CT iodé
30 to 500	Calcifications ponctuelles
60 to 100	Hémorragie intracrânienne
35	Matière grise
25	Matière blanche
20 to 40	Muscle, tissus mous
0	Eau
-30 to -70	Graisse
< -1000	Air

L'absorption des rayons X est mesurée sur l'échelle de Hounsfield. Dans cette échelle, nous fixons l'intensité de l'air à -1000 et l'intensité de l'eau à 0. Il est essentiel de comprendre que Hounsfield est une échelle absolue, contrairement à l'IRM où nous avons une échelle relative de 0 à 255.

La figure 9 illustre certains des tissus de base et leurs valeurs d'intensité correspondantes. Les chiffres peuvent légèrement varier dans les images réelles.

*Fig.9 Echelle de Hounsfield*

Les os ont une intensité élevée. Nous coupons généralement l'image pour avoir une plage maximale supérieure. Par exemple, la valeur maximale peut être 1000, pour des raisons pratiques. Le problème est que les bibliothèques de visualisation fonctionnent à l'échelle [0,255]. Il ne serait pas très judicieux de visualiser toute l'échelle de Hounsfield (de -1000 à 1000+) à 256 échelles pour le diagnostic médical. Au lieu de cela, nous limitons notre attention aux différentes parties de cette gamme et nous nous concentrons sur les tissus sous-jacents.

## 4.2.2 Visualisation des données TDM

La convention d'image médicale pour couper la gamme Hounsfield consiste à choisir une intensité centrale, appelée niveau et une fenêtre, comme illustré sur la figure suivante :

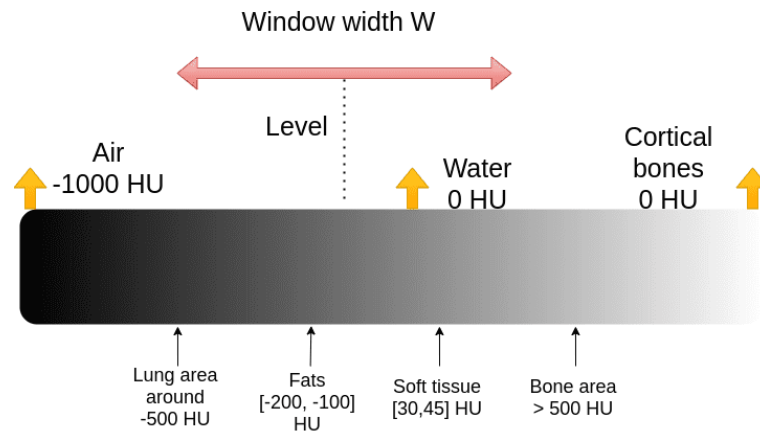


Fig.10 fenêtrage et mise à niveau de l'échelle Hounsfield

Deux paramètres modulables définissent la fenêtre utile de densités, variable en fonction des structures étudiés :

- Le niveau (level) : valeur centrale des densités visualisées
- La largeur de la fenêtre (window) détermine le nombre de niveaux de densité.

Les structures dont la densité se situe en dehors de la fenêtre seront représentées uniquement en noir (densité inférieur) ou en blanc (densité supérieur)

C'est en fait une convention assez laide pour les informaticiens. Nous aimerions juste le min et le max de la plage :

$$max = niveau + fen\hat{e}tre/2$$

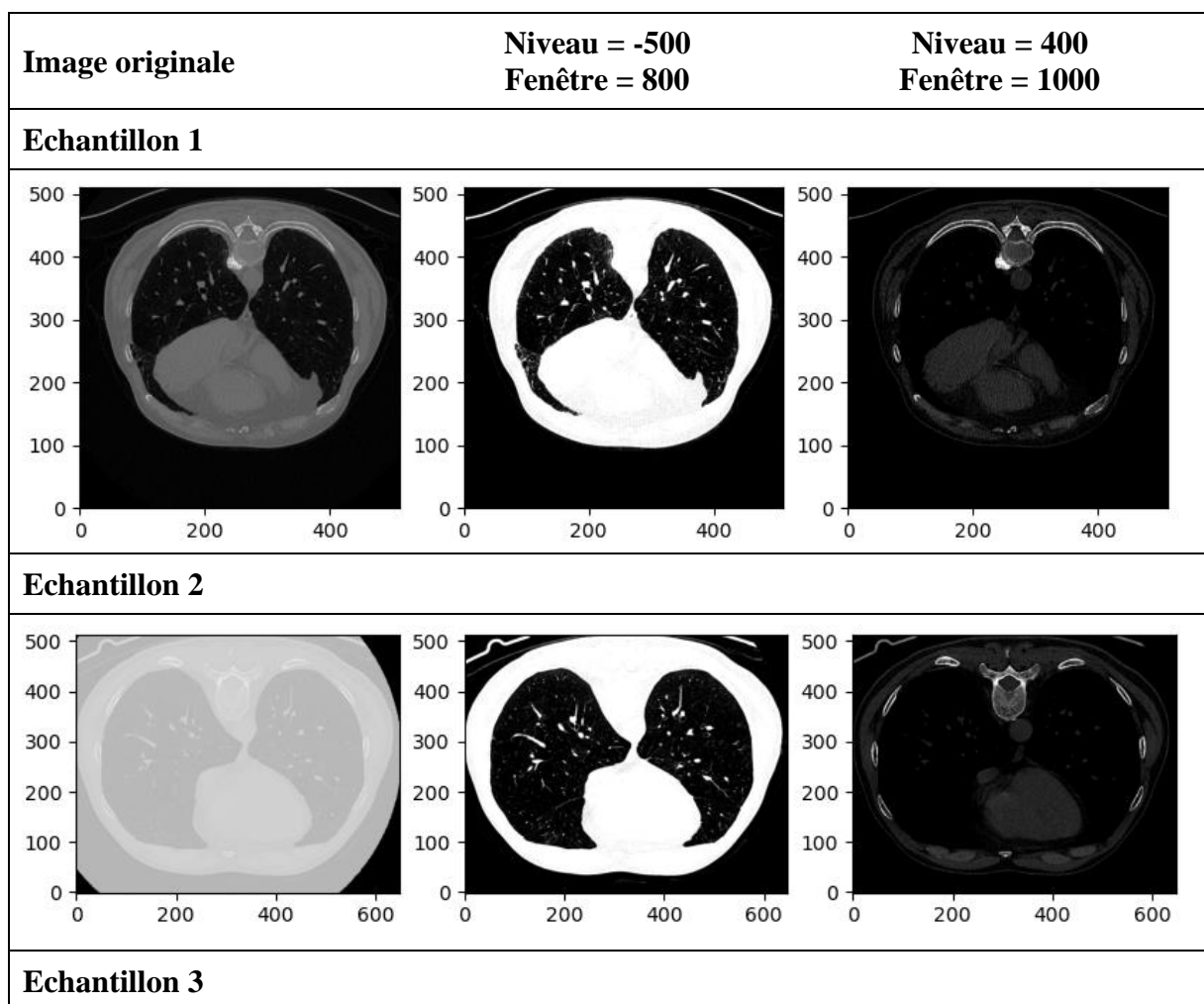
$$min = niveau - fen\hat{e}tre/2$$

## 4.2.3 Base de données utilisée

La base de données utilisé est un ensemble de fichier à accès libre, se sont des images de TDM thoracique. Cette base de données utilise un format spécifique aux images de scanner, c'est le format *NIFTI*, pour la lecture de ce format nous avons besoin d'une bibliothèque spécifique de Python qui est *NiBabel*.

### 4.3 Visualisation des données de scanner TDM : niveau et fenêtrage

Nous utilisons l'échelle de Hounsfield pour la visualisation des différentes constituants des images, dans les exemples suivants nous avons créé une fonction qui modifie l'intensité du niveau du gris selon l'échelle précitée. Nous montrons dans 3 échantillons d'image de scanner thoracique le résultat de ce traitement qui permet d'accentuer les formes de certains organes selon le niveau de Hounsfield, la figure suivante montre ces échantillons ainsi que le résultat du traitement qui permet de montrer dans l'image du milieu les poumons, et l'image de droite la cage thoracique.



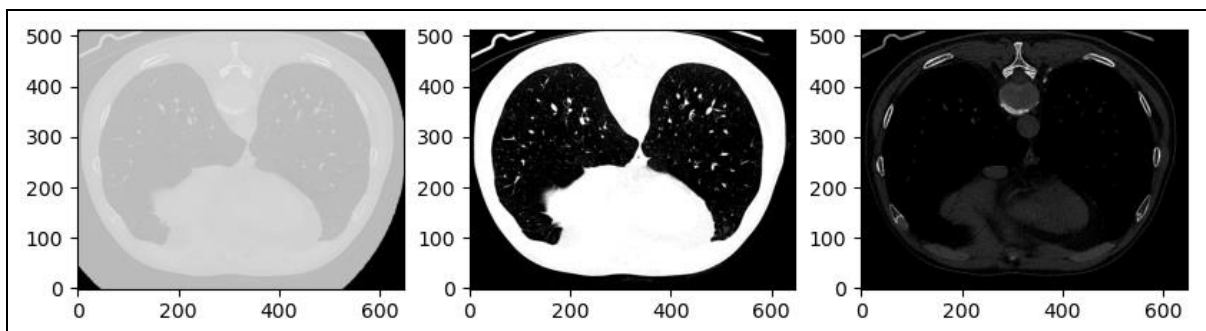


Fig.12 Application du fenêtrage et mise à niveau avec l'échelle de Hounsfield sur 3 échantillons d'image de scanner thoracique

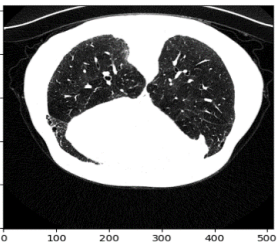
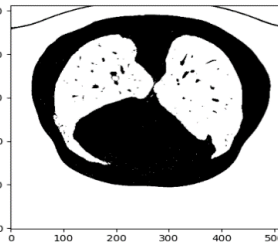
### 4.4 Application pour le calcul de la surface des poumons sur des images TDM

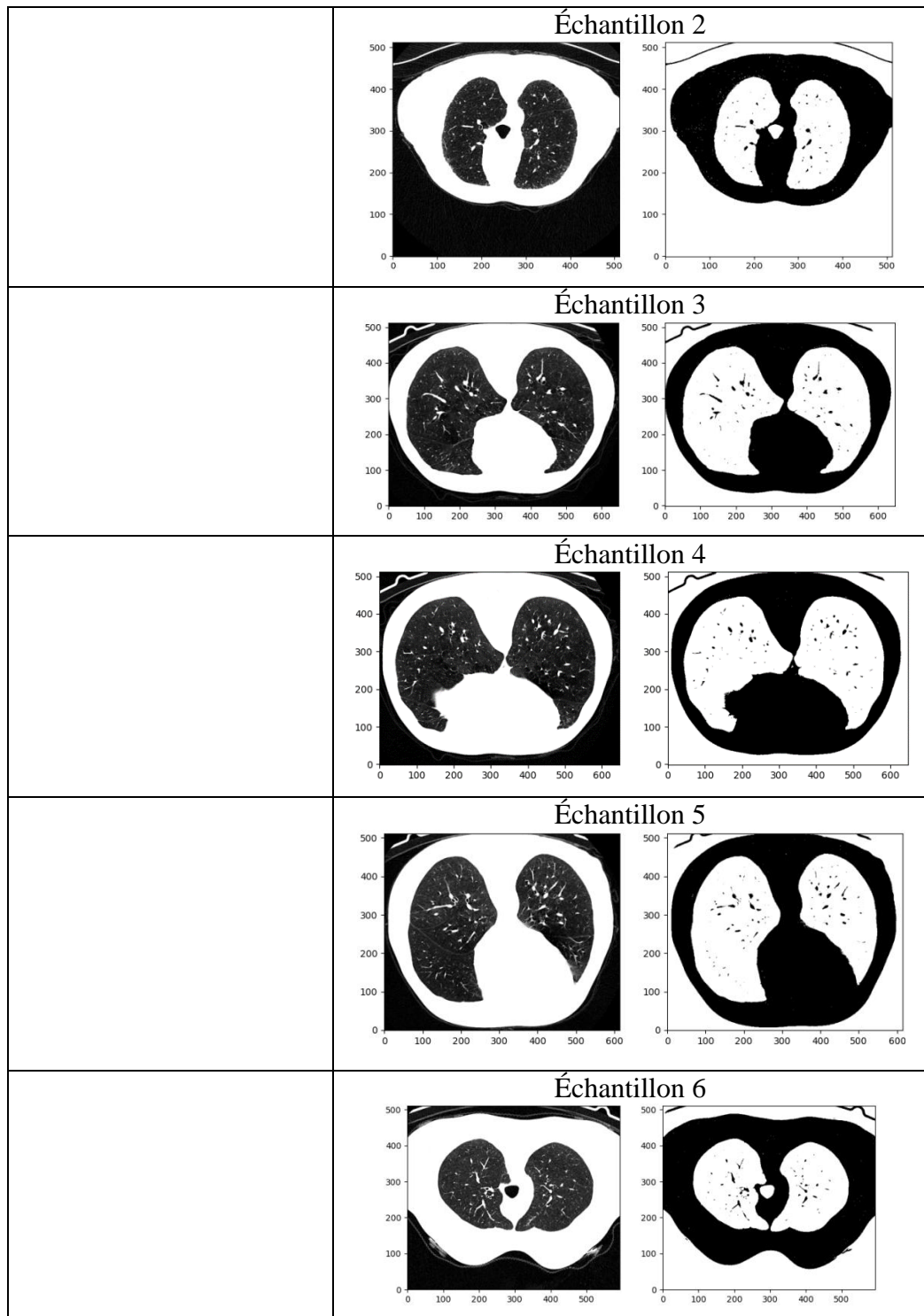
Dans la suite de ce chapitre, nous allons utiliser le langage de programmation Python ainsi qu'un ensemble de bibliothèque de traitement graphique pour le calcul de la surface des poumons, le résultat de ces calculs est un élément essentiel pour le diagnostic des maladies et pathologies pulmonaires. Pour arriver à ce résultat, nous suivrons à un certain nombre d'étapes successives pour appliquer

#### 4.4.1 Etape 1 : binarisation des images et seuillage des intensités.

Nous nous attendons à ce que les poumons soient dans la gamme d'unités Hounsfield de  $[-1000, -300]$ . À cette fin, nous devons faire un seuillage sur cette plage d'intensité et binariser ensuite les valeurs résultantes à 0 et 1, nous obtiendrons ainsi le résultat suivant de la figure 13.

Le seuillage est la méthode la plus simple de segmentation d'image. À partir d'une image en niveau de gris, le seuillage d'image peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc (monochrome). On remplace un à un les pixels d'une image par rapport à une valeur seuil fixée. Ainsi, si un pixel a une valeur supérieure au seuil, il prendra la valeur 255 (blanc), et si sa valeur est inférieure, il prendra la valeur 0 (noir).

Images originales	Après seuillage	Après binarisation
	Échantillon1	
		



*Fig.13 Application de binarisation d'image*

Dans la figure 13, nous avons choisi 6 échantillons, ou nous pouvons voir qu'après seuillage et binarisation nous pouvons identifier l'aspect des poumons dans chaque image.

#### 4.4.2 Etape 2 : Recherche de contours

Pour la vision par ordinateur, un contour est un ensemble de points décrivant une ligne ou une zone. Ainsi, pour chaque contour détecté, nous n'obtiendrons pas un masque binaire complet, mais plutôt un ensemble de valeurs x et y.

Dans cette application nous avons utilisé l'algorithme de *Martching squares* pour trouver l'ensemble des contours dans nos images de coupe thoraciques. Nous utilisons la fonction *skimage.measure.find\_contours*, où les valeurs d'une matrice sont interpolées linéairement pour fournir une meilleure précision des contours de sortie. Les contours qui coupent le bord de l'image sont ouverts ; tous les autres sont fermés.

Les résultats de l'application de cette fonction sur les échantillons présentés est donnée sur la figure 14.

#### 4.4.3 Etape 3 : Trouver la zone pulmonaire à partir d'un ensemble des contours possibles

Pour identifier la zone pulmonaire sur les échantillons, nous devons imposer un certain nombre de critères qui permettent de sélectionner une paire de contours pulmonaire parmi les contours trouvés dans l'étapes précédentes

- ✓ En premier, nous allons extraire un polygone convexe de l'ensemble des contours en utilisons la fonction *scipy.spatial.ConvexHull*. Ceci exclu les ensembles-contours non fermés.
- ✓ Ensuite, nous supposons une taille minimale du contour qui doit être de 2000 pixels pour représenter les poumons.
- ✓ Cela peut inclure ou non la circonférence du corps, ce qui donne plus de 3 bandes. Lorsque cela se produit, le corps est facilement éliminé en obtenant le plus grand volume du contour qui répond aux hypothèses décrites précédemment.
- ✓ Les zones restantes correspondent aux poumons

#### 4.4.4 Etape 4 : Extraction du masque binaire à partir du contour des poumons.

Ensuite, nous l'enregistrons sous forme de fichier astucieux, nous devons donc convertir l'ensemble de points en un masque binaire pulmonaire. Pour cela, j'ai utilisé la librairie *pillow* python qui dessine un polygone et crée un masque d'image binaire. Ensuite, je fusionne tous les masques des contours pulmonaires déjà trouvés.

Module ImageDraw :Le module ImageDraw fournit des graphiques 2D simples pour les objets Image. Vous pouvez utiliser ce module pour créer de nouvelles images, annoter ou retoucher des images existantes et générer des graphiques à la volée pour une utilisation

Desired lung area in mm<sup>2</sup> mm It is simply the number of non-zero elements multiplied by the two-pixel dimensions of the corresponding image. Lung regions are saved in a csv file with the name of the image.

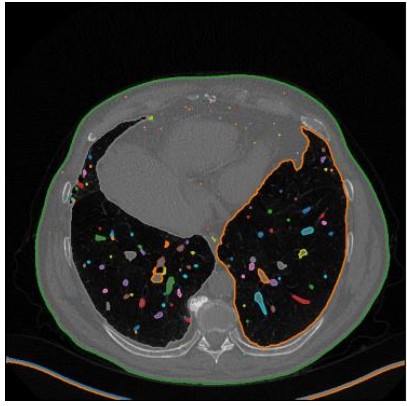
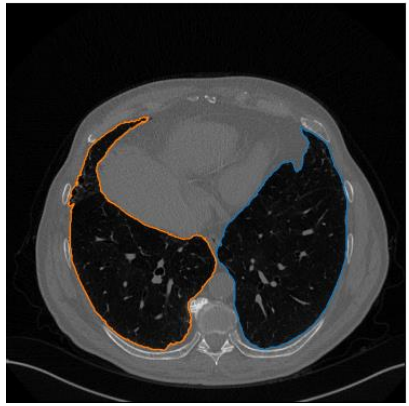
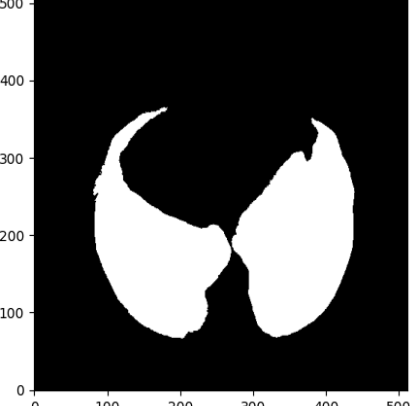
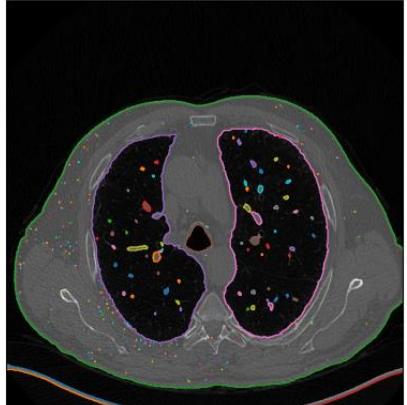

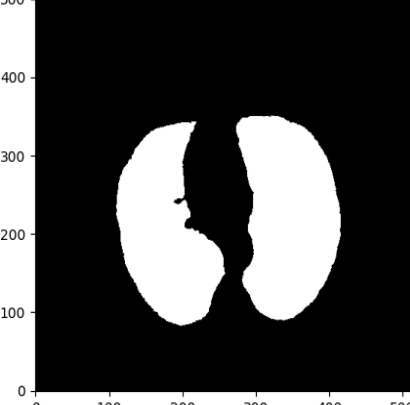
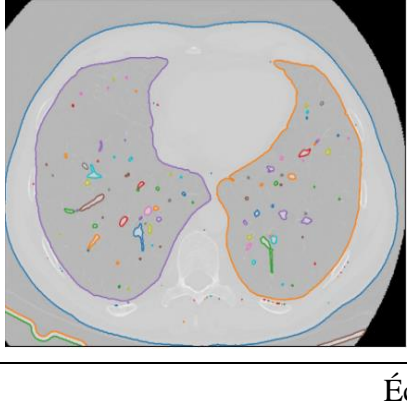
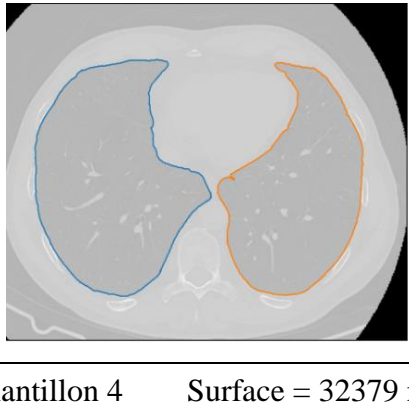
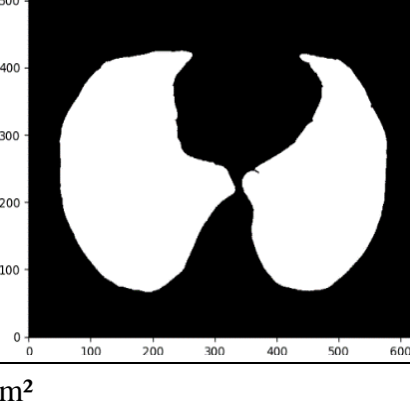
#### 4.4.5 Etape 5 : trouver la dimension de pixel pour le calcul de la surface en mm<sup>2</sup>

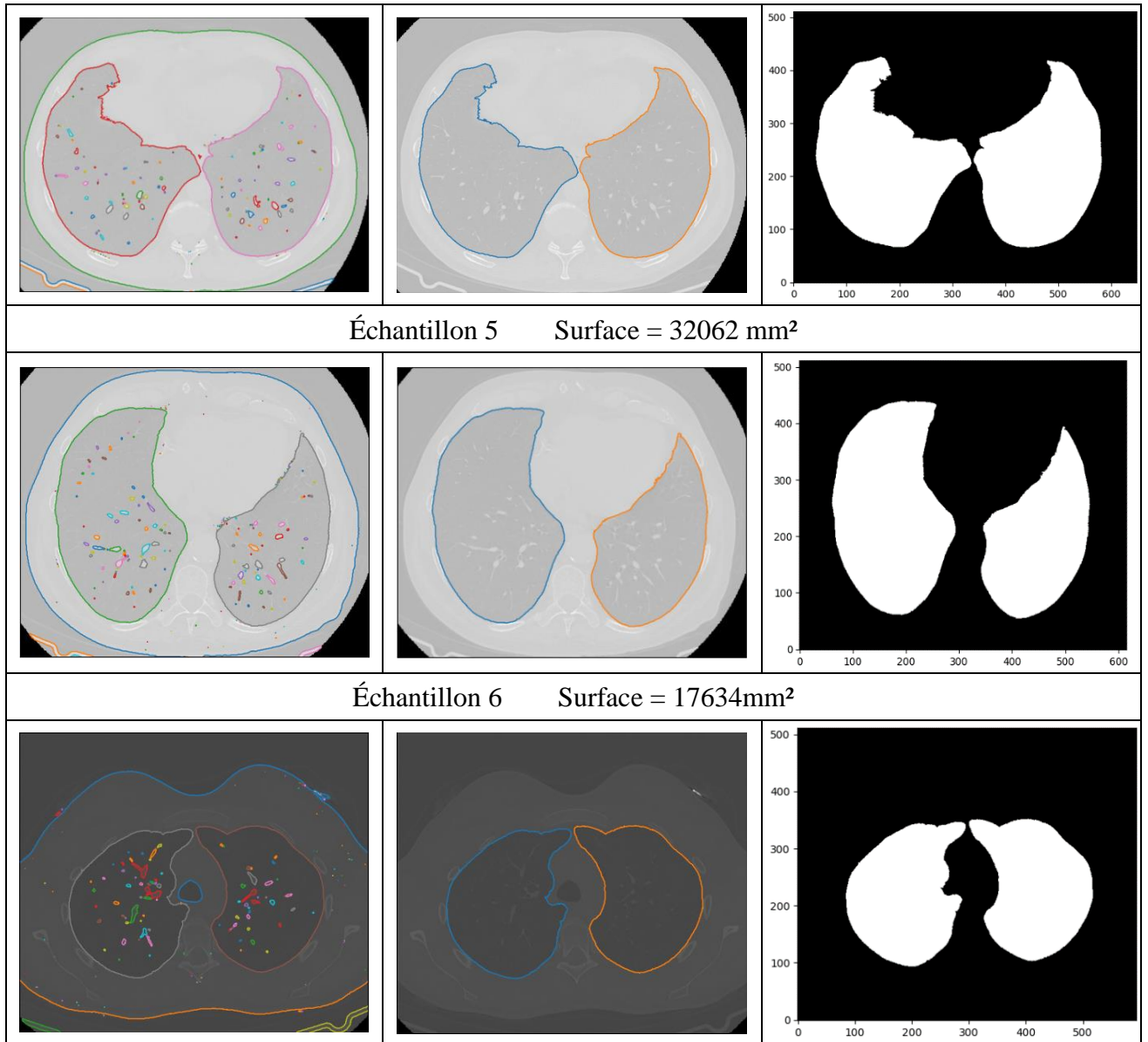
Obtenez la dimension de chaque pixel sur une image TDM, Non seulement nous diviserons les poumons, mais nous trouverons également la surface réelle en mm <sup>2</sup> mm.

Pour ce faire, nous devons trouver la taille réelle des dimensions en pixels. Chaque image peut contenir une image différente (pixdim dans le fichier d'en-tête élégant). Voyons d'abord le fichier d'en-tête :

Ici, je ne montrerai que quelques champs importants de l'en-tête :

Pour mémoire, srow\_x, srow\_y, srow\_z est la matrice affine de l'image. Bitpix est le nombre de bits que nous utilisons pour représenter l'intensité de chaque pixel. Définissons donc une fonction qui lit ces informations à partir du fichier d'en-tête. Basé sur le format astucieux, chaque dimension du fichier astucieux a une dimension en pixels. Ce dont nous avons besoin, c'est de connaître les indices à 2 dimensions de l'image et leurs dimensions respectives en pixels

Étape 3 : Recherche de tous les contours	Étape 4 : Recherche des contours des poumons	Étape 5 : Affichage du masque des poumons
Échantillon 1      Surface = 34549 mm <sup>2</sup>		
		
Échantillon 2      Surface = 31502 mm <sup>2</sup>		
		
Échantillon 3      Surface = 33369 mm <sup>2</sup>		
		
Échantillon 4      Surface = 32379 mm <sup>2</sup>		



*Fig.14 recherche des contours et les contours de poumons et l'affichage des masques et des calculs la surface*

## 5 Conclusion

Dans ce chapitre nous avons décrit la technologie ainsi que les différentes procédures que nous avons utilisées pour l'implémentation de l'approche que nous avons utilisé pour le calcul de la surface des poumons.

Nous avons utilisé la recherche des contours par segmentation des poumons on la combinant avec la détection des dimensions de pixel.

L'efficacité de l'approche utilisé est démontrée à travers un ensemble d'échantillon qui on permis de vérifier toutes les étapes de programmation ainsi que le choix des paramètres d'identification des sections de l'image.

# Chapitre 4

## Implémentation et résultats expérimentaux

## 1 Introduction

Dans ce chapitre, nous allons présenter l'implémentation sur FPGA des différents programmes que nous avons réalisé au chapitre précédent. Dans ce travail, nous allons implémenter ces programmes sur une carte Pynq-Z2 qui est basé sur le circuit FPGA Zynq XC7Z020, conçue de manière à pour prendre en charge Pynq.

PYNQ est un projet open source de Xilinx. Il fournit un framework basé sur Jupyter avec des API Python pour l'utilisation des plates-formes Xilinx. PYNQ programmeurs de concevoir des systèmes embarqués et d'utiliser des dispositifs Zynq. Ce type de circuit FPGA bénéficie d'une logique programmable et d'un processeur ARM avancé dans Zynq, Ainsi, les concepteurs peuvent construire des systèmes embarqués plus puissants.

## 2 Utilisation de la carte Pynq-Z2

Dans les paragraphes suivants nous allons présenter les outils et les méthodes que nous devons suivre pour l'utilisation et la programmation de la carte Pynq-Z2.

### 2.1 Prérequis d'utilisation

Pour l'utilisation de la carte Pynq-Z2, nous devons disposer des outils suivants :

- Une carte Pynq-Z2
- Un ordinateur avec navigateur internet compatible
- Un câble Ethernet
- Un câble micro USB
- Une carte micro-SD avec une image préchargée du système d'exploitation requis, ou une carte vierge d'un minimum de 8 Go

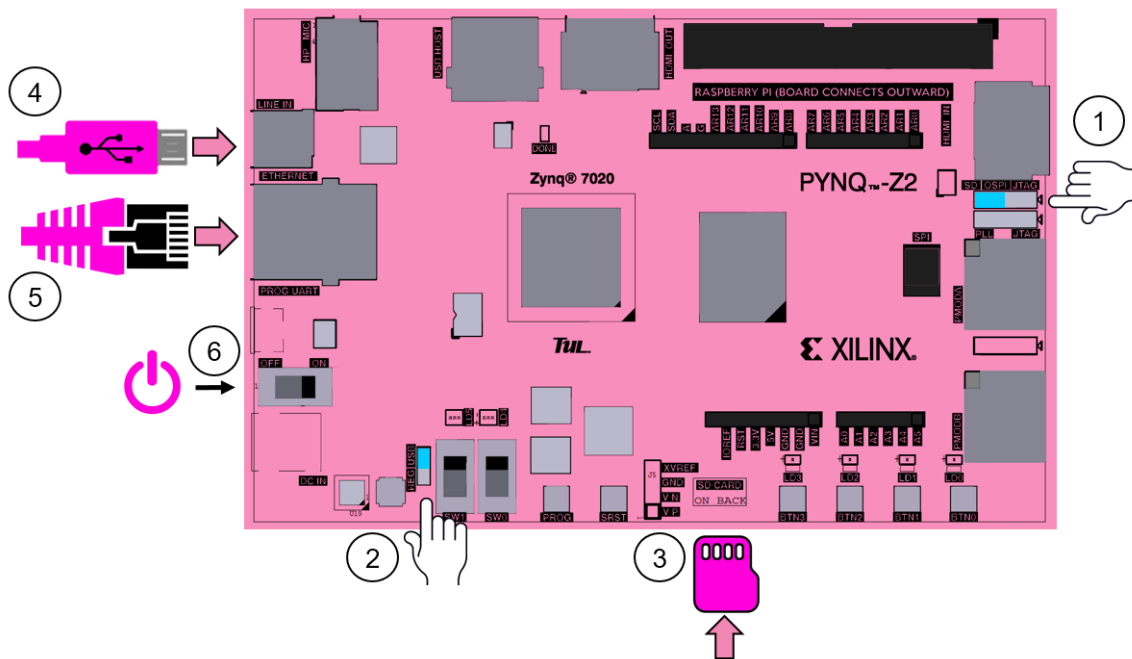


Fig.1. Schéma représentative de la carte Pynq-Z2

## 2.2 Configuration de carte

Pour une première utilisation de la carte Pynq-Z2, nous devons suivre les étapes de configuration citées ci-dessous :

1. Positionner le cavalier **\*\* Boot \*\*** sur la position SD. Cela impose à la carte pour démarrer à partir de la carte micro-SD
2. Pour alimenter la carte à partir du câble micro USB, positionner le cavalier d'alimentation sur la position USB. (Vous pouvez également alimenter la carte à partir d'une source d'alimentation continue de 12V en définissant le cavalier sur Reg)
3. Insérez la carte Micro SD chargée avec l'image Pynq-Z2 dans la fente de carte micro SD sous la carte
4. Connectez le câble USB à votre PC / ordinateur portable, et au port PROG - UART Micro USB sur la carte
5. Connectez le port Ethernet en suivant les instructions ci-dessous
6. Allumez le Pynq-Z2 et vérifiez la séquence de démarrage en suivant les instructions ci-dessous

## 2.3 Mise sous tension de la carte Pynq-Z2

Comme indiqué à l'étape 6 de la configuration de la carte, il faut positionner l'interrupteur d'alimentation en position ON pour allumer la carte. La LED rouge s'allumera immédiatement

pour confirmer que la carte est sous tension. Après quelques secondes, les LED jaune/verte/DONE s'allume pour montrer que le circuit Zynq est opérationnel.

Après une minute, deux LED bleues et quatre LED jaune/vert commence à clignoter simultanément. Les LED bleues s'allumeront et s'éteindront alors que les LED jaunes/restent allumées. Le système est maintenant démarré et prêt à l'emploi.

## 2.4 Connexion réseau

Une fois la carte configurée, il faut la connecter à un ordinateur pour commencer à utiliser la plateforme de programmation Jupyter Notebook.

### 2.4.1 Ethernet

Il est préférable de connecter la carte à un réseau ou un routeur avec un accès Internet. Cela permettra de mettre à jour carte et d'installer facilement de nouveaux packages. Sinon, il faut seulement la connectée directement à un ordinateur.

### 2.4.2 Connecter la carte à un ordinateur

Il faut avoir un port Ethernet disponible sur l'ordinateur, et il faut avoir les autorisations pour configurer l'interface réseau. Avec une connexion directe, on peut utiliser Pynq, mais la carte n'aura pas accès à Internet. Néanmoins, on peut partager la connexion Internet sur l'ordinateur avec la carte Pynq.

Connexion direct à un ordinateur (IP statique) :

- Attribuer une adresse IP statique
- Connectez la carte au port Ethernet de l'ordinateur
- Regarder à <http://192.168.2.99>

### 2.4.3 Connection à un routeur réseau

Si on veut connecter la carte à un routeur ou à un réseau avec un serveur DHCP, la carte obtiendra automatiquement une adresse IP. Il faut s'assurer que le réseau à toutes les permissions pour connecter une nouvelle machine, sinon la carte peut ne pas se connecter correctement.

Connection à un routeur / réseau (DHCP):

- Connecter le port Ethernet de votre carte à un routeur
- Connecter l'ordinateur au réseau Ethernet ou le WiFi sur le routeur
- Accéder à la page Web avec l'adresse *http://<Adresse IP du carte>*

### 2.4.4 Connexion à Jupyter Notebook

Une fois la carte Pynq Z2 configurée, pour se connecter à Jupyter Notebooks il faut ouvrir un navigateur Web et accéder à l'adresse :

<http://192.168.2.99>, dans cette configuration la carte est connectée à un ordinateur via une adresse IP statique. Si la carte est configurée correctement, un écran de connexion s'affichera. Le nom d'utilisateur est *xilinx* et le mot de passe est également *xilinx*.

Après connexion, nous aurons l'affichage suivant :



Fig.2. Fenêtre du navigateur avec Jupyter après connexion à la carte

Le nom d'hôte par défaut est pynq et l'adresse IP statique par défaut est 192.168.2.99. Si l'adresse IP statique de la carte est modifiée, il faut modifier l'adresse dans le navigateur.

## 2.5 Exemples des notebooks

PYNQ utilise l'environnement Jupyter Notebook pour fournir des exemples et de la documentation. À l'aide du navigateur internet, on peut afficher et exécuter la documentation des notebooks Jupyter de manière interactive.

Le dossier *Getting\_Started* dans la zone d'accueil de Jupyter comprend des blocs-notes d'introduction à Jupyter.

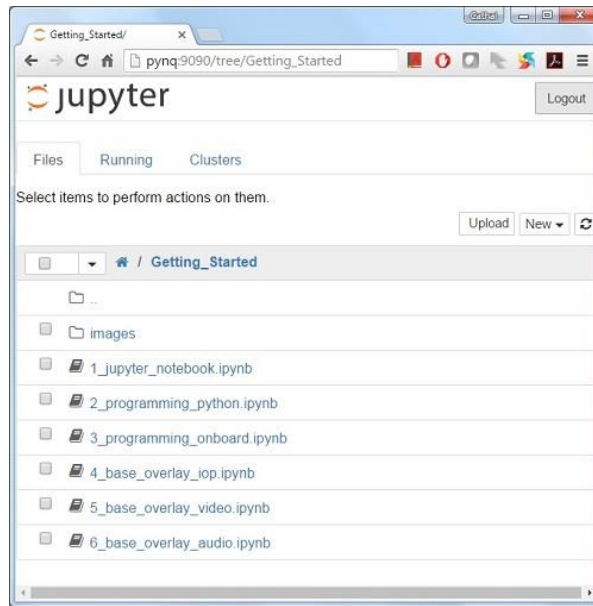


Fig.3. Dossier des exemples de notebook

Les exemples de notebook ont été divisés en catégories

Commun : exemples qui ne sont pas spécifiques à la partie matérielle du Zynq (Overlays).

En fonction de la carte et de l'image PYNQ qu'on utilise, d'autres dossiers peuvent être disponibles avec des exemples liés aux superpositions. Par exemple. Le répertoire de base contiendra des exemples liés aux overlays de base. Si on installe un overlay supplémentaire, un dossier contenant des exemples de notebook sera généralement copié ici.

### 3 Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes pour l'utilisation et la programmation des cartes Xilins Zynq 7000. Nous avons appliqué l'exemple d'application que nous avons vu dans le chapitre précédent pour la validation de notre travail.

Les résultats du traitement d'image sont les mêmes mais le temps de traitement sur la carte FPGA est plus important.

# Conclusion générale

De nos jours, l'imagerie médicale est devenue un outil indispensable en médecine. Elle permet aux médecins à mieux comprendre le fonctionnement du corps humain et de se servir pour appuyer leur diagnostic afin d'établir les traitements nécessaires. Également, en chirurgie, les praticiens utilisent l'imagerie médicale pour s'initier et guider le geste chirurgical.

d'imagerie médicale et nous avons basé plus particulièrement sur l'imagerie à rayon x scanner. Cette dernière offre une qualité d'image bien contrastée, avec une haute résolution, et une bonne finesse des coupes, ce qui en fait actuellement un élément de diagnostic incontournable au sein du milieu médical.

Dans ce travail, nous avons commencé par la présentation des différentes techniques d'imagerie médicale . Cette dernière offre une qualité d'image bien contrastée, avec une haute résolution, et une bonne finesse des coupes, ce qui en fait actuellement un élément de diagnostic incontournable au sein du milieu médical.

Nous avons d'abord abordé les python supervisées et non supervisées. Ensuite, nous avons testé quelques images par la python , qui nous semble très utilisée vu sa fiabilité. Les tests effectués nous ont permis de constater que le choix du nombre de classe joue un rôle très important dans la segmentation.

Nous concluons que les images médicales utilisées rendent la segmentation plus compliquée, vu la complicité de leurs texture très variée, à comparer aux images de simple . Nous espérons que ce travail apportera un plus aux travaux déjà fait

## Références bibliographiques

- 
- [1] P.Destuynder « Analyse et traitement des images numériques, Hermes science ISBN 2-7462-1288-9, Lavoisier, 2006
- [2] Arnaud CAPRI ‘‘Caractérisation des objets dans une image en vue d’une aide à l’interprétation et d’une compression adaptée au contenu : application aux images échographique’’. Université d’Orléans. Faculté des Sciences et Technologies Industrielles. Thèse de doctorat 2007.
- [3] <https://www.f-legrand.fr/scidoc/docmml/image/niveaux/images/images.html>
- [4] AISSAT Abdelkader "Etude sur l’imagerie médicale : prétraitement, segmentation et amélioration des exploitations" Université SAAD DAHLAB de BLIDA,2018
- [5] C.Houassine, segmentation d’images par une approche biomimétique hybride. ccccc université universite m’hamed bougara- boumerdes. 2012
- [6] N MERABET, M MAHLIA, recherche d’images par le contenu, université abou bakrbelkaid– tlemcen.2011
- [7] D.Djafar, la découverte des concepts sémantiques cachés avec plusieurs c niveaux d’abstraction pour la recherche d’images,université kasdi merbah ouargla.2013
- [8] Dominique, C. Emmanuel, G, Béatrice De Keukeleire et al. Imagerie Médicale du Futur. Pipame (Le Pôle Interministériel de Prospective et d’Anticipation de Mutations Economiques). Edition de Nicole Merle-Lamoot, Gilles Pannetier, Octobre 2013.
- [9] M.Sandeli, traitement d’images par des approches bio-inspirées application à la de segmentation d’images, université constantine 2. 2014.
- [10] <http://www.medecine-nucleaire.fr/tomographie-emission-positons-tep.html> . 11/05/2022.
- [11] <http://tpebaudechahdeldicque.e-monsite.com/pages/ii-les-differents-principes-d-imagerie-medicale-2/ii-les-differents-principes-d-imagerie-medicale.html> . 11/05/2022.
- [12] <http://techno.clicforum.fr/t201-IRM.htm> . 11/05/2022.
- [13] <http://irmquebec.com/lirm/quest-ce-quune-irm> .11/05/2022.
- [14] Abdelkrim MEZIANE ‘‘ Systèmes d’information en Imagerie médicale : Moyen de diagnostic, de thérapie et d’économie’’ RIST VOL. 11 N° 02 Année 2001.IMG
- [15] H. MEDROUK et S. TOUCHERIFT, «Segmentation par classification d'une partie de l’image», Mémoire d'ingénieur en automatique, UMMTO, 2010
- [17] ABDELLI Lotfi , "Architecture pour détection de contour actif Sur FPGA ",univ USD- Blida,2012
- [18] Simon HAENE, AndreaS BURG, Davia PERELS, Peter LUETHI, Norpert FELBER et Wolfgang FICHTNER , " FPGA implementation of Viterbi decoders for MIMO-BICM. In 39th Asilomar Conference on Signals, Systems & Computers", USA, novembre 2005.

- [19] Panan POTIPANTONG, Theerayod WIANGTONG," Phaophak SIRISUK et Apisak WORAPISHET : A scaleable FFT/IFFT kernel for communication systems using codesign approach", Singapore, décembre 2009
- [20] Mihail PETROV et Manfred GLESNER , " Optimal FFT architecture selection for OFDM receivers on FPGA. ", Singapore, décembre 2005
- [21] D. A. Buell, J.M. Arnold, W.J. Kleinfelder, " Splash 2, FPGAs in a custom computing machine ", IEEE Computer Society Press, Los Alamitos, 1996.
- [22] Jacques Weber, Sébastien Moutault, Maurice Meaudre, "Le langage VHDL : du langage au circuit, du circuit au langage", DUNOD, 2007.
- [23] le langage de description VHDL, T. BLOTIN Lycée Paul-Eluard 93206 SAINT-DENIS
- [24] Remacle Matthieu, Schmitz Thomas, Pierlot Vincent, « Le langage VHDL », Microélectronique, 24 février 2016.
- [25] C. J. Chou, S. Mohanakrishnan, J. B. Evans. "FPGA implementation of digital filters ", Application and Technology 1993.
- [26] <https://www.silabs.com/documents/public/white-papers/Which-ARM-Cortex-Core-Is-Right-for-Your-Application.pdf>
- [27] <https://pynq.readthedocs.io/en/latest/>
- [28] <https://www.avnet.com/wps/portal/apac/products/c/xilinx-pynq/reference-materials/pynq-framework-and-image/>
- [29] [www.codetd.com/fr/article/12698509?fbclid=IwAR0QbU4FrqDUqMCWW8racFtKZN9dIueCk0Ja9BzWZKnqrjFivUhIO-WzMPc](http://www.codetd.com/fr/article/12698509?fbclid=IwAR0QbU4FrqDUqMCWW8racFtKZN9dIueCk0Ja9BzWZKnqrjFivUhIO-WzMPc)